

УТВЕРЖДЕН

13.00078-01 33 01-ЛУ

МНОГОУРОВНЕВАЯ СИСТЕМА ПРОГРАММНОГО ДИАГНОСТИРОВАНИЯ  
ВЫЧИСЛИТЕЛЬНОГО КОМПЛЕКСА СМ 1700 (МСПД)

РУКОВОДСТВО ПРОГРАММИСТА

13.00078-01 33 01

ЛИСТОВ 214

ПЕРВ. ПРИМЕН.  
13.00078-01

1987

ЛИТЕРА 0

АННОТАЦИЯ

В НАСТОЯЩЕМ ДОКУМЕНТЕ ОПИСАНЫ УСЛОВИЯ ПРИМЕНЕНИЯ, ХАРАКТЕРИСТИКИ, НАЗНАЧЕНИЕ И СТРУКТУРА ДИАГНОСТИЧЕСКИХ ПРОГРАММ, А ТАКЖЕ СПОСОБЫ ОБРАЩЕНИЯ К НИМ. РАССМОТРЕНЫ ФУНКЦИИ ДИАГНОСТИЧЕСКОГО СУПЕРВИЗОРА И ЕГО ВЗАИМОДЕЙСТВИЕ С ДИАГНОСТИЧЕСКОЙ ПРОГРАММОЙ. ДАНО ОПИСАНИЕ МАКРОКОМАНД ДИАГНОСТИЧЕСКОЙ СИСТЕМЫ. РУКОВОДСТВО ЯВЛЯЕТСЯ ПОСОБИЕМ ПО РАЗРАБОТКЕ И ЭКСПЛУАТАЦИИ ДИАГНОСТИЧЕСКИХ ПРОГРАММ МСПД ВК СМ 1700.

СОДЕРЖАНИЕ

1.	НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ ПРОГРАММЫ	13
1.1.	НАЗНАЧЕНИЕ ДИАГНОСТИЧЕСКИХ ПРОГРАММ	13
1.2.	ПРИМЕНЕНИЕ ДИАГНОСТИЧЕСКИХ ПРОГРАММ	13
2.	ХАРАКТЕРИСТИКИ ПРОГРАММЫ	14
3.	ДИАГНОСТИЧЕСКИЙ СУПЕРВИЗОР. ОПИСАНИЕ ВХОДНОГО ЯЗЫКА	17
3.1.	КОМАНДЫ УПРАВЛЕНИЯ ПОСЛЕДОВАТЕЛЬНОСТЬЮ ВЫПОЛНЕНИЯ ПРОГРАММ И ТЕСТОВ	18
3.2.	КОМАНДА ОБРАЩЕНИЯ К КОМАНДНЫМ ФАЙЛАМ	19
3.3.	КОМАНДА УПРАВЛЕНИЯ ВЫПОЛНЕНИЕМ	19
3.4.	ФЛАГИ КОМАНД	19
3.5.	ФЛАГИ СОБЫТИЙ	21
3.6.	КОМАНДЫ ОТЛАДКИ ПРОГРАММЫ	22
3.7.	ФУНКЦИОНАЛЬНОЕ ОПИСАНИЕ СУПЕРВИЗОРА	22
4.	СТРУКТУРА ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ	24
4.1.	ГОЛОВНОЙ МОДУЛЬ	25
4.1.1.	ПРЕДИСЛОВИЕ МОДУЛЯ	25
4.1.2.	ОПИСАНИЯ МОДУЛЯ	26
4.1.3.	ОПРЕДЕЛЕНИЯ ЛОКАЛЬНЫХ МАКРОКОМАНД	26
4.1.4.	ОПЕРАТОРЫ ПРЯМОГО ПРИСВАИВАНИЯ	26
4.1.5.	ПРОГРАММНАЯ СЕКЦИЯ ЗАГОЛОВКА ПРОГРАММЫ	

	(PHEADER)	27
4.1.6.	ПРОГРАММНАЯ СЕКЦИЯ ДИСПЕТЧЕРА ПРОГРАММЫ (DISPATCH)	28
4.1.7.	ПРОГРАММНАЯ СЕКЦИЯ ГЛОБАЛЬНЫХ ДАННЫХ ПРОГРАММЫ	28
4.1.8.	ТЕКСТОВАЯ ЧАСТЬ ПРОГРАММЫ	29
4.1.9.	ПОДПРОГРАММА ИНИЦИАЛИЗАЦИИ	29
4.1.10.	ПОДПРОГРАММА ОЧИСТКИ	31
4.1.11.	ПОДПРОГРАММА ЗАКЛЮЧИТЕЛЬНОГО (ИТОГОВОГО) ОТЧЕТА	31
4.1.12.	ДОКУМЕНТИРОВАНИЕ ПОДПРОГРАММ ИНИЦИАЛИЗАЦИИ, ОЧИСТКИ И ЗАКЛЮЧИТЕЛЬНОЙ ПОДПРОГРАММЫ	32
4.1.13.	ПРИМЕР ОФОРМЛЕНИЯ ГОЛОВНОГО МОДУЛЯ	33
4.2.	МОДУЛЬ ГЛОБАЛЬНЫХ ПОДПРОГРАММ	34
4.3.	ТЕСТОВЫЕ МОДУЛИ	36
4.3.1.	ПРЕДИСЛОВИЕ МОДУЛЯ	37
4.3.2.	ОПЕРАТОРЫ ПРЯМОГО ПРИСВАИВАНИЯ	37
4.3.3.	ТЕСТОВЫЕ ПОДПРОГРАММЫ	37
4.3.4.	ПРИМЕР ОФОРМЛЕНИЯ ТЕСТОВОГО МОДУЛЯ	40
5.	ОПИСАНИЕ ПОДКЛЮЧЕНИЯ ПРОГРАММ ПОЛЬЗОВАТЕЛЯ	41
5.1.	ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ УРОВНЯ 2	42
5.2.	ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ УРОВНЯ 2R	42
5.3.	ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ УРОВНЯ 3	43
5.4.	ТЕСТИРОВАНИЕ РЕГИСТРОВ	43
5.5.	РАСПЕЧАТКА СООБЩЕНИИ ОБ ОШИБКАХ	44
6.	МАКРОКОМАНДЫ ДИАГНОСТИЧЕСКИХ ПРОГРАММ	45
7.	МАКРОКОМАНДЫ ДИАГНОСТИЧЕСКОЙ СИСТЕМЫ МСПД	46

7.1.	КОДИРОВКА МАКРОКОМАНД ДИАГНОСТИЧЕСКОЙ СИСТЕМЫ	46
7.2.	ИСПОЛЬЗОВАНИЕ МАКРОКОМАНД #DS_ИМЯ_S И #DS_ИМЯ_L	48
7.3.	ИСПОЛЬЗОВАНИЕ МАКРОКОМАНДЫ ФОРМАТА #DS_ИМЯ_S	50
7.4.	ВЫДАЧА КОДОВ СОСТОЯНИЯ	50
7.5.	МАКРОКОМАНДЫ ФОРМАТА ПРОГРАММЫ	51
7.5.1.	МАКРОКОМАНДЫ ПРОГРАММНОГО МОДУЛЯ	52
7.5.1.1.	#DS_BGNMOD - НАЧАЛО ПРОГРАММНОГО МОДУЛЯ	52
7.5.1.2.	#DS_ENDMOD - КОНЕЦ ПРОГРАММНОГО МОДУЛЯ	53
7.5.2.	#DS_HEADER - ФОРМИРОВАНИЕ ЗАГОЛОВКА ПРОГРАММНОЙ СЕКЦИИ	53
7.5.3.	#DS_DISPATCH - ИНИЦИАЛИЗАЦИЯ ТАБЛИЦЫ ДИСПЕТЧЕРА	55
7.5.4.	МАКРОКОМАНДЫ СТАТИСТИЧЕСКОЙ ТАБЛИЦЫ	56
7.5.4.1.	#DS_BGNSTAT - НАЧАЛО СТАТИСТИЧЕСКОЙ ТАБЛИЦЫ	56
7.5.4.2.	#DS_ENDSTAT - КОНЕЦ СТАТИСТИЧЕСКОЙ ТАБЛИЦЫ	56
7.5.5.	МАКРОКОМАНДЫ ОПРЕДЕЛЕНИЯ ТЕСТОВЫХ СЕКЦИЙ	57
7.5.5.1.	#DS_SECTION - ОПРЕДЕЛЕНИЕ ТЕСТОВЫХ СЕКЦИЙ В ГОЛОВНОМ МОДУЛЕ	57
7.5.5.2.	#DS_SECDEF - ОПРЕДЕЛЕНИЕ ТЕСТОВЫХ СЕКЦИЙ В ТЕСТОВЫХ МОДУЛЯХ	57
7.5.6.	МАКРОКОМАНДЫ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ	58
7.5.6.1.	#DS_BGNINIT - НАЧАЛО ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ	58
7.5.6.2.	#DS_ENDINIT - КОНЕЦ ПОДПРОГРАММЫ	

	ИНИЦИАЛИЗАЦИИ	59
7.5.7.	МАКРОКОМАНДЫ ПОДПРОГРАММЫ ОЧИСТКИ	59
7.5.7.1.	#DS_BGNCLEAN - НАЧАЛО ПОДПРОГРАММЫ ОЧИСТКИ	59
7.5.7.2.	#DS_ENDCLEAN - КОНЕЦ ПОДПРОГРАММЫ ОЧИСТКИ	60
7.5.8.	МАКРОКОМАНДЫ ПОДПРОГРАММЫ ЗАКЛЮЧИТЕЛЬНОГО ОТЧЕТА	60
7.5.8.1.	#DS_BGNSUMMARY - НАЧАЛО ПОДПРОГРАММЫ ПЕЧАТИ ИТОГОВОГО ОТЧЕТА	61
7.5.8.2.	#DS_ENDSUMMARY - КОНЕЦ ПОДПРОГРАММЫ ПЕЧАТИ ИТОГОВОГО ОТЧЕТА	61
7.5.9.	МАКРОКОМАНДЫ ПОДПРОГРАММЫ СООБЩЕНИИ ОБ ОШИБКАХ	62
7.5.9.1.	#DS_BGNMESSAGE - НАЧАЛО ПОДПРОГРАММЫ ПЕЧАТИ СООБЩЕНИИ ОБ ОШИБКАХ	62
7.5.9.2.	#DS_ENDMESSAGE - КОНЕЦ ПОДПРОГРАММЫ ПЕЧАТИ СООБЩЕНИИ ОБ ОШИБКАХ	62
7.5.10.	МАКРОКОМАНДЫ ПОДЗАГОЛОВКА	63
7.5.10.1.	#DS_SBTTL - ФОРМИРОВАНИЕ ПОДЗАГОЛОВКА ТЕСТА ИЛИ ПОДТЕСТА	63
7.5.10.2.	#DS_PAGE - ИСКЛЮЧЕНИЕ РАСПЕЧАТКИ ВЫЗОВА МАКРОКОМАНДЫ #DS_SBTTL	64
7.5.11.	МАКРОКОМАНДЫ ТЕСТОВОЙ ПОДПРОГРАММЫ	64
7.5.11.1.	#DS_BGNTEST - НАЧАЛО ТЕСТОВОЙ ПОДПРОГРАММЫ	64
7.5.11.2.	#DS_ENDTEST - КОНЕЦ ТЕСТОВОЙ ПОДПРОГРАММЫ	65
7.5.12.	МАКРОКОМАНДЫ ДАННЫХ ДЛЯ ТЕСТА	66
7.5.12.1.	#DS_BGNDATA - НАЧАЛО МАССИВА ДАННЫХ ДЛЯ	

ТЕСТА	66
7.5.12.2. #DS_ENDDATA - КОНЕЦ МАССИВА ДАННЫХ ДЛЯ ТЕСТА	67
7.6. МАКРОКОМАНДЫ УПРАВЛЕНИЯ ПРОГРАММОЙ	68
7.6.1. МАКРОКОМАНДЫ ПЕРЕДАЧИ УПРАВЛЕНИЯ В ПОДПРОГРАММЕ ИНИЦИАЛИЗАЦИИ	68
7.6.1.1. #DS_BPASSO - ПЕРЕДАЧА УПРАВЛЕНИЯ ПРИ ПЕРВОМ ВЫПОЛНЕНИИ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ	70
7.6.1.2. #DS_BNPASSO - ПЕРЕДАЧА УПРАВЛЕНИЯ ПРИ НЕ ПЕРВОМ ВЫПОЛНЕНИИ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ	71
7.6.1.3. #DS_ENDPASS_X - СООБЩЕНИЕ СУПЕРВИЗОРУ ОБ ОКОНЧАНИИ ПРОХОДА ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ	71
7.6.2. МАКРОКОМАНДЫ ФЛАГА БЫСТРОЙ ПРОВЕРКИ	72
7.6.2.1. #DS_BQUICK - ПЕРЕХОД В РЕЖИМЕ БЫСТРОЙ ПРОВЕРКИ	72
7.6.2.2. #DS_BNQUICK - ПЕРЕХОД ПРИ ОТСУТСТВИИ РЕЖИМА БЫСТРОЙ ПРОВЕРКИ	73
7.6.3. МАКРОКОМАНДЫ ФЛАГА ОПЕРАТОРА	73
7.6.3.1. #DS_BOPER - ПЕРЕХОД ПРИ НАЛИЧИИ ОПЕРАТОРА	74
7.6.3.2. #DS_BNOPER - ПЕРЕХОД ПРИ ОТСУТСТВИИ ОПЕРАТОРА	74
7.6.4. МАКРОКОМАНДЫ УПРАВЛЕНИЯ ПРОГРАММНЫМ ПОДТЕСТОМ	75
7.6.4.1. #DS_BGNSUB - НАЧАЛО ПОДТЕСТА	75
7.6.4.2. #DS_ENDSUB - КОНЕЦ ПОДТЕСТА	76
7.6.5. МАКРОКОМАНДЫ ЗАЦИКЛИВАНИЯ	76

7.6.5.1.	ADS_SCKLOOP - УПРАВЛЕНИЕ ЗАЦИКЛИВАНИЕМ	77
7.6.5.2.	ADS_INLOOP_X - ПРОВЕРКА ЗАЦИКЛИВАНИЯ	77
7.6.5.3.	ПРИМЕР ИСПОЛЬЗОВАНИЯ МАКРОКОМАНДЫ ЗАЦИКЛИВАНИЯ	78
7.6.6.	МАКРОКОМАНДЫ ВЫХОДА ИЗ ПРОГРАММЫ И ПОДПРОГРАММЫ	78
7.6.6.1.	ADS_ESCAPE - УСЛОВНЫЙ ВЫХОД ИЗ ТЕСТА ИЛИ ПОДТЕСТА	79
7.6.6.2.	ADS_EXIT - БЕЗУСЛОВНЫЙ ВЫХОД ИЗ ПРОГРАММНОЙ ФАЗЫ	80
7.6.6.3.	ADS_ABOPT - ОТМЕНА ПРОГРАММЫ ИЛИ ТЕСТА	81
7.6.6.4.	ADS_BREAK - ВКЛИНИВАНИЕ В ДИАГНОСТИЧЕСКУЮ ПРОГРАММУ	82
7.6.7.	МАКРОКОМАНДЫ ПЕРЕХОДА ПО ОКОНЧАНИИ ВЫПОЛНЕНИЯ ПОДПРОГРАММЫ	83
7.6.7.1.	ADS_VSCOMPLETE - ПЕРЕХОД ПРИ УСПЕШНОМ ВЫПОЛНЕНИИ ПОДПРОГРАММЫ	84
7.6.7.2.	ADS_VNSCOMPLETE - ПЕРЕХОД ПРИ НЕУСПЕШНОМ ВЫПОЛНЕНИИ ПОДПРОГРАММЫ	84
7.6.8.	МАКРОКОМАНДЫ ПЕРЕХОДА ПРИ ОШИБКЕ	85
7.6.8.1.	ADS_ERROR - ПЕРЕХОД ПРИ ОШИБКЕ	85
7.6.8.2.	ADS_VNERROR - ПЕРЕХОД ПРИ ОТСУТСТВИИ ОШИБКИ	86
7.7.	МАКРОКОМАНДЫ Р-ТАБЛИЦЫ	86
7.7.1.	ОБЩИЕ СВЕДЕНИЯ О Р-ТАБЛИЦЕ	86
7.7.2.	ФОРМАТ Р-ТАБЛИЦЫ	88
7.7.3.	ОБЩИЕ СВЕДЕНИЯ О МАКРОКОМАНДАХ Р-ТАБЛИЦЫ	90
7.7.4.	МАКРОКОМАНДЫ ОПИСАТЕЛИ Р-ТАБЛИЦЫ	92
7.7.4.1.	ADS_INITIALIZE - НАЧАЛО ОБРАБОТКИ	



ТАБЛИЦЫ P	93
7.7.4.2. #DS_#DECIMAL - ЗАДАНИЕ ИНФОРМАЦИИ ДЛЯ ЗАПРОСА У ОПЕРАТОРА ДЕСЯТИЧНОЙ ВЕЛИЧИНЫ	94
7.7.4.3. #DS_#OCTAL - ЗАДАНИЕ ИНФОРМАЦИИ ДЛЯ ЗАПРОСА У ОПЕРАТОРА ВОСЬМЕРИЧНОЙ ВЕЛИЧИНЫ	94
7.7.4.4. #DS_#HEXADECIMAL - ЗАДАНИЕ ИНФОРМАЦИИ ДЛЯ ЗАПРОСА У ОПЕРАТОРА ШЕСТНАДЦАТЕРИЧНОЙ ВЕЛИЧИНЫ	95
7.7.4.5. #DS_#STRING - ЗАДАНИЕ ИНФОРМАЦИИ ДЛЯ ЗАПРОСА У ОПЕРАТОРА СТРОКИ СИМВОЛОВ	96
7.7.4.6. #DS_#LITERAL - ОПРЕДЕЛЕНИЕ ЗНАЧЕНИЯ КОНСТАНТЫ	96
7.7.4.7. #DS_#FETCH - ИЗВЛЕЧЕНИЕ ПОЛЯ ИЗ ТАБЛИЦЫ P	97
7.7.4.8. #DS_#STORE - ВНЕСЕНИЕ ВЕЛИЧИНЫ В ТАБЛИЦУ P	97
7.7.4.9. #DS_#END - ОКОНЧАНИЕ ОБРАБОТКИ ТАБЛИЦЫ P	98
7.7.5. #DS_#DEVTYPE - УКАЗАНИЕ ТИПОВ УСТРОЙСТВ И АДРЕСОВ ОПИСАТЕЛЕЙ P-ТАБЛИЦЫ	98
7.7.6. МАКРОКОМАНДЫ ОПРЕДЕЛЕНИЯ СТРУКТУРЫ	99
7.7.6.1. #DEFINI - НАЧАЛО СТРУКТУРЫ ДАННЫХ	99
7.7.6.2. #DEF - ОПРЕДЕЛЕНИЕ ПОЛЕЙ В СТРУКТУРЕ	99
7.7.6.3. #DEFEND - ОКОНЧАНИЕ ОПРЕДЕЛЕНИЙ	100
7.7.7. ПРИМЕРЫ МАКРОКОМАНД ОПРЕДЕЛЕНИЯ СТРУКТУРЫ ДАННЫХ И ОПИСАТЕЛЕЙ P-ТАБЛИЦЫ	100
7.7.8. #DS_#PHARD_X - ПОЛУЧЕНИЕ БАЗОВОГО АДРЕСА ТАБЛИЦЫ P	102
7.8. МАКРОКОМАНДЫ ДИАЛОГА МЕЖДУ ПРОГРАММОЙ И	

	ОПЕРАТОРОМ	103
7.8.1.	ADS_ASKSTR_X - ЗАПРОС У ОПЕРАТОРА СТРОКИ	104
7.8.2.	ADS_ASKDATA_X - ЗАПРОС У ОПЕРАТОРА ЧИСЛОВОЙ ВЕЛИЧИНЫ	105
7.8.3.	ADS_ASKVLD_X - ЗАПРОС У ОПЕРАТОРА ЧИСЛОВОГО ЗНАЧЕНИЯ В КОДЕ КОИ-8	107
7.8.4.	ADS_ASKLGCL_X - ЗАПРОС У ОПЕРАТОРА ЛОГИЧЕСКОГО ОТВЕТА	109
7.8.5.	ADS_ASKADR_X - ЗАПРОС АДРЕСА У ОПЕРАТОРА	110
7.8.6.	ADS_PARSE_X - АНАЛИЗ КОМАНДЫ	112
7.8.7.	ADS_CLI - ДЕРЕВО ИНТЕРПРЕТАТОРА КОМАНДНОЙ СТРОКИ	113
7.8.8.	ПРИМЕР ИСПОЛЬЗОВАНИЯ МАКРОКОМАНД ДИАЛОГА МЕЖДУ ПРОГРАММОЙ И ОПЕРАТОРОМ	115
7.9.	МАКРОКОМАНДЫ СООБЩЕНИЯ ОБ ОШИБКАХ	119
7.9.1.	МАКРОКОМАНДЫ ПЕЧАТИ ЗАГОЛОВКА СООБЩЕНИЯ	120
7.9.1.1.	ADS_ERRSYS_X - РАСПЕЧАТКА ЗАГЛАВНОЙ ИНФОРМАЦИИ ПРИ НЕУСТРАНИМОЙ СИСТЕМНОЙ ОШИБКЕ	121
7.9.1.2.	ADS_ERRDEV_X - РАСПЕЧАТКА ЗАГЛАВНОЙ ИНФОРМАЦИИ ПРИ НЕУСТРАНИМОЙ ОШИБКЕ УСТРОЙСТВА	122
7.9.1.3.	ADS_ERRHARD_X - РАСПЕЧАТКА ЗАГЛАВНОЙ ИНФОРМАЦИИ ОБ АППАРАТНОЙ ОШИБКЕ	123
7.9.1.4.	ADS_ERRSOFT_X - РАСПЕЧАТКА ЗАГЛАВНОЙ ИНФОРМАЦИИ ОБ УСТРАНИМОЙ ОШИБКЕ ПРОГРАММНЫХ СРЕДСТВ	124
7.9.2.	МАКРОКОМАНДЫ ПЕЧАТИ ИНФОРМАЦИИ	125

7.9.2.1.	#DS_PRINTB_X - ПЕЧАТЬ ОСНОВНОЙ ИНФОРМАЦИИ ОБ ОШИБКЕ	127
7.9.2.2.	#DS_PRINTX_X - ПЕЧАТЬ РАСШИРЕННОЙ ИНФОРМАЦИИ ОБ ОШИБКЕ	128
7.9.2.3.	#DS_PRINTF_X - ПРИНУДИТЕЛЬНАЯ ПЕЧАТЬ СООБЩЕНИЯ	128
7.9.2.4.	#DS_PRINTS_X - ПЕЧАТЬ ИТОГОВОГО ОТЧЕТА	129
7.9.3.	ДИРЕКТИВЫ ПРЕОБРАЗОВАНИЯ FAO	130
7.10.	МАКРОКОМАНДЫ ОПРЕДЕЛЕНИЯ СИМВОЛИЧЕСКИХ ИМЕН	133
7.10.1.	#DS_BITDEF - ОПРЕДЕЛЕНИЕ МНЕМОНИЧЕСКИХ ОБОЗНАЧЕНИЙ БИТОВОЙ МАСКИ	134
7.10.2.	#DS_DSADef - ОПРЕДЕЛЕНИЕ СИМВОЛИЧЕСКИХ ИМЕН ФЛАГОВ	135
7.10.3.	#DS_DSDEF - ОПРЕДЕЛЕНИЕ КОДОВ СОСТОЯНИЯ	136
7.10.4.	#DS_DSSDEF - ОПРЕДЕЛЕНИЕ ТОЧЕК ВХОДА СЛУЖБНЫХ СРЕДСТВ СУПЕРВИЗОРА	138
7.10.5.	#DS_ENVDEF - ОПРЕДЕЛЕНИЕ КОДОВ СРЕДЫ	138
7.10.6.	#DS_ERRDEF - ОПРЕДЕЛЕНИЕ СМЕЩЕНИЙ СПИСКА АРГУМЕНТОВ ВЫЗОВА ОШИБКИ	139
7.10.7.	#DS_HRDEF - ОПРЕДЕЛЕНИЕ СИМВОЛИЧЕСКИХ ИМЕН СМЕЩЕНИЙ ЭЛЕМЕНТОВ ТАБЛИЦЫ P	139
7.10.8.	#DS_PARDEF - ОПРЕДЕЛЕНИЕ ПАРАМЕТРОВ	141
7.11.	МАКРОКОМАНДЫ УПРАВЛЕНИЯ ПАМЯТЬЮ	142
7.11.1.	#DS_MM0N_X - ВКЛЮЧЕНИЕ УПРАВЛЕНИЯ ПАМЯТЬЮ	142
7.11.2.	#DS_MM0FF_X - ВЫКЛЮЧЕНИЕ УПРАВЛЕНИЯ ПАМЯТЬЮ	143
7.11.3.	#DS_GETBUF_X - ПОЛУЧЕНИЕ ОБЛАСТИ ВИРТУАЛЬНОЙ ПАМЯТИ	143

7.11.4.	EDS_RELBUF_X - ОСВОБОЖДЕНИЕ БУФЕРА	144
8.	ОБРАЩЕНИЕ К ПРОГРАММЕ	145
8.1.	ЗАПУСК ДИАГНОСТИЧЕСКОГО СУПЕРВИЗОРА В АВТОНОМНОМ РЕЖИМЕ	145
8.2.	ЗАПУСК ДИАГНОСТИЧЕСКОГО СУПЕРВИЗОРА ПОД УПРАВЛЕНИЕМ МОС ВП	146
8.3.	ЗАПУСК ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ	147
9.	ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ. СООБЩЕНИЯ	148
	ПРИЛОЖЕНИЕ 1. АЛФАВИТНЫЙ УКАЗАТЕЛЬ МАКРОКОМАНД	150
	ПРИЛОЖЕНИЕ 2. ПРИМЕР ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ	154
	ПРИЛОЖЕНИЕ 3. РИСУНКИ	202
	ПЕРЕЧЕНЬ ССЫЛОЧНЫХ ДОКУМЕНТОВ	213

13.00078-01 33 01

## 1. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ ПРОГРАММЫ

### 1.1. НАЗНАЧЕНИЕ ДИАГНОСТИЧЕСКИХ ПРОГРАММ

ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ ИСПОЛЬЗУЮТСЯ ПРИ РАЗРАБОТКЕ ВЫЧИСЛИТЕЛЬНЫХ МАШИН, ИХ ПРОИЗВОДСТВЕ И ЭКСПЛУАТАЦИИ. ОТ ДИАГНОСТИЧЕСКИХ ПРОГРАММ ТРЕБУЕТСЯ ОБНАРУЖЕНИЕ МАКСИМАЛЬНО ВОЗМОЖНОГО ЧИСЛА НЕИСПРАВНОСТЕЙ, КОТОРЫЕ МОГУТ ВОЗНИКНУТЬ В ПРОВЕРЯЕМЫХ СХЕМАХ.

ПРИ РАЗРАБОТКЕ ВЫЧИСЛИТЕЛЬНЫХ МАШИН ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ НЕОБХОДИМЫ ДЛЯ ОБНАРУЖЕНИЯ ФУНКЦИОНАЛЬНЫХ ОШИБОК И ОШИБОК, ВОЗНИКАЮЩИХ НА РАННЕЙ СТАДИИ РЕАЛИЗАЦИИ РАЗРАБОТКИ ОБОРУДОВАНИЯ. НЕПОЛНОЕ ОБНАРУЖЕНИЕ ОШИБОК НА СТАДИИ РАЗРАБОТКИ МОЖЕТ ВПОСЛЕДСТВИИ ПРИВЕСТИ К ДОРОГОСТОЯЩИМ ПЕРЕДЕЛКАМ В ОБОРУДОВАНИИ.

В ПРОЦЕССЕ ПРОИЗВОДСТВА ТЕСТИРОВАНИЕ ИСПОЛЬЗУЕТСЯ ДЛЯ ОТБРАКОВКИ ДЕФЕКТНЫХ МОДУЛЕЙ, ДЛЯ ОБНАРУЖЕНИЯ И ЛОКАЛИЗАЦИИ НЕИСПРАВНОСТЕЙ УСТРОЙСТВ И СИСТЕМЫ В ЦЕЛОМ.

НА ЭТАПЕ ЭКСПЛУАТАЦИИ ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ ИСПОЛЬЗУЮТСЯ ДЛЯ УСТАНОВКИ, ОБСЛУЖИВАНИЯ И РЕМОНТА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ САМЫХ РАЗНООБРАЗНЫХ КОНФИГУРАЦИЙ, ВЫПОЛНЯЮЩИХ БОЛЬШОЕ ЧИСЛО РАЗЛИЧНЫХ ПРИКЛАДНЫХ ЗАДАЧ.

### 1.2. ПРИМЕНЕНИЕ ДИАГНОСТИЧЕСКИХ ПРОГРАММ

НАИБОЛЕЕ ВАЖНЫМ ПРИМЕНЕНИЕМ ДИАГНОСТИЧЕСКИХ ПРОГРАММ ЯВЛЯЕТСЯ ИХ ИСПОЛЬЗОВАНИЕ ОПЕРАТОРОМ ВЫЧИСЛИТЕЛЬНОЙ МАШИНЫ, КОТОРЫЙ УПРАВЛЯЕТ ТЕСТИРОВАНИЕМ, ЛОКАЛИЗАЦИЕЙ НЕИСПРАВНОСТЕЙ И ПРОВЕРКОЙ КАЧЕСТВА РЕМОНТА. ВАЖНЫМ МОМЕНТОМ ПРИ РАЗРАБОТКЕ ДИАГНОСТИЧЕСКИХ ТЕСТОВ И ДОКУМЕНТАЦИИ ДОЛЖНО БЫТЬ

13.00078-01 33 01

ОБЕСПЕЧЕНИЕ ЭФФЕКТИВНОЙ РАБОТЫ ОПЕРАТОРА. ДОСТИЖЕНИЮ ЭТОЙ ЦЕЛИ СЛУЖАТ ТАКЖЕ И ЗНАЧИТЕЛЬНОЕ ЧИСЛО ФУНКЦИИ ДИАГНОСТИЧЕСКОГО СУПЕРВИЗОРА СМ 1700. ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ ДОЛЖНЫ БЫТЬ РАЗРАБОТАНЫ И РЕАЛИЗОВАНЫ ТАКИМ ОБРАЗОМ, ЧТОБЫ НАРЯДУ С ВЫСОКОЙ ЭФФЕКТИВНОСТЬЮ СРЕДСТВ ТЕСТИРОВАНИЯ И РЕМОНТА БЫЛА ОБЕСПЕЧЕНА ПРОСТОТА ИХ ИСПОЛЬЗОВАНИЯ ОПЕРАТОРОМ.

ПОЛЕЗНО ТАКЖЕ ИМЕТЬ ВОЗМОЖНОСТЬ АВТОМАТИЗИРОВАННОГО ПРИМЕНЕНИЯ ДИАГНОСТИЧЕСКИХ ПРОГРАММ С ИСПОЛЬЗОВАНИЕМ РАЗМЕЩЕННЫХ НА НОСИТЕЛЕ КОМАНДНЫХ ФАЙЛОВ ИЛИ СПЕЦИАЛЬНЫХ УПРАВЛЯЮЩИХ ПРОГРАММ.

## 2. ХАРАКТЕРИСТИКИ ПРОГРАММЫ

ОСНОВНЫМИ ХАРАКТЕРИСТИКАМИ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ ЯВЛЯЮТСЯ СЛЕДУЮЩИЕ:

- 1) ПОЛНОТА ОБНАРУЖЕНИЯ НЕИСПРАВНОСТЕЙ;
- 2) ВОЗМОЖНОСТЬ ЛОКАЛИЗАЦИИ НЕИСПРАВНОСТЕЙ И ПОИСКА НЕИСПРАВНОСТЕЙ;
- 3) РАЗМЕР ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ;
- 4) ВРЕМЯ ВЫПОЛНЕНИЯ.

2.1. ПОЛНОТА ОБНАРУЖЕНИЯ НЕИСПРАВНОСТЕЙ ЯВЛЯЕТСЯ ОСНОВНЫМ ПОКАЗАТЕЛЕМ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ.

2.2. ЛОКАЛИЗАЦИЯ НЕИСПРАВНОСТЕЙ ИМЕЕТ ВАЖНОЕ ЗНАЧЕНИЕ ДЛЯ РЕМОНТНЫХ ДИАГНОСТИЧЕСКИХ ПРОГРАММ. ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ ДОЛЖНЫ СООБЩАТЬ О НЕИСПРАВНОСТЯХ С ТОЧНОСТЬЮ ДО ОДНОГО ИЛИ НЕСКОЛЬКИХ СМЕННЫХ БЛОКОВ.

13.00078-01 33 01

2.3. ДЛЯ ПОИСКА НЕИСПРАВНОСТЕЙ ИСПОЛЬЗУЮТСЯ СООБЩЕНИЯ ЛИБЕ ИТОГОВЫЕ ОТЧЕТЫ ОБ ОШИБКАХ, В КОТОРЫХ УКАЗЫВАЕТСЯ ТЕСТ И ПОДТЕСТ, ОБНАРУЖИВШИЕ ОШИБКУ, КРАТКОЕ ОПИСАНИЕ ПРОВЕРЯЕМОЙ ФУНКЦИИ И ДАННЫХ.

ОТЧЕТ ОБ ОШИБКЕ ДОЛЖЕН ИНФОРМИРОВАТЬ И НАПРАВЛЯТЬ СПЕЦИАЛИСТА ПО РЕМОНТУ, НЕ ПЕРЕГРУЖАЯ ЕГО ИЗЛИШНИМИ ДАННЫМИ. ДИАГНОСТИКА СМ. 1700 ПРЕДУСМАТРИВАЕТ ТРЕХУРОВНЕВУЮ СТРУКТУРУ ОТЧЕТОВ (СООБЩЕНИЙ) ОБ ОШИБКАХ: ЗАГОЛОВОК, ОСНОВНОЙ ОТЧЕТ И РАСШИРЕННЫЙ ОТЧЕТ. ЗАГОЛОВОК СОДЕРЖИТ СУЩЕСТВЕННУЮ ИНФОРМАЦИЮ О ТЕСТАХ И ИХ ФУНКЦИИ. ОСНОВНОЙ ОТЧЕТ ДАЕТ ИСХОДНУЮ И КОНЕЧНУЮ ИНФОРМАЦИЮ О ТЕСТЕ. РАСШИРЕННЫЙ ОТЧЕТ ДОЛЖЕН СОДЕРЖАТЬ В ПРОИЗВОЛЬНОЙ ФОРМЕ ИНФОРМАЦИЮ ДЛЯ ПОИСКА НЕИСПРАВНОСТЕЙ. ДОЛЖНА ОБЕСПЕЧИВАТЬСЯ ВОЗМОЖНОСТЬ ВЫДАЧИ ОТЧЕТОВ ПО ТРЕБОВАНИЮ ПОЛЬЗОВАТЕЛЯ.

ПОИСКУ И УСТРАНЕНИЮ НЕИСПРАВНОСТЕЙ ДОЛЖНА ПОМОГАТЬ ТАКЖЕ ХОРОШО РАЗРАБОТАННАЯ И ПОСТРОЕННАЯ ДОКУМЕНТАЦИЯ.

В ДИАГНОСТИЧЕСКОЙ ПРОГРАММЕ ДОЛЖНО БЫТЬ ЯСНО УКАЗАНО (С ПОМОЩЬЮ КОММЕНТАРИЕВ), ЧТО ДЕЛАЕТ СООТВЕТСТВУЮЩИЙ ТЕСТ И КАК ОН ЭТО ДЕЛАЕТ, КАКОВЫ ТЕСТОВЫЕ ДАННЫЕ И РЕЗУЛЬТАТЫ ТЕСТА.

ДЛЯ УПРАВЛЕНИЯ ПОИСКОМ НЕИСПРАВНОСТЕЙ ДОЛЖНЫ ИСПОЛЬЗОВАТЬСЯ ТАКИЕ СРЕДСТВА, КАК ЗАЦИКЛИВАНИЕ НА ОШИБКЕ, ОСТАНОВКА ПО ОШИБКЕ, ВЫБОР ТЕСТА И ПОДТЕСТА, ЗВУКОВОЙ СИГНАЛ ПРИ ОШИБКЕ, А ТАКЖЕ ДРУГИЕ ВОЗМОЖНОСТИ, ПРЕДОСТАВЛЯЕМЫЕ ДИАГНОСТИЧЕСКИМ СУПЕРВИЗОРОМ.

2.4. НЕОБХОДИМО СТРЕМИТЬСЯ К УМЕНЬШЕНИЮ РАЗМЕРОВ ДИАГНОСТИЧЕСКИХ ПРОГРАММ ЗА СЧЕТ РЕАЛИЗАЦИИ В ВИДЕ ПОДПРОГРАММ ЧАСТЕ ПОВТОРЯЮЩИХСЯ ФУНКЦИЙ, А ТАКЖЕ ИСПОЛЬЗОВАНИЯ ДРУГИХ СПОСОБОВ.

ОДНАКО УМЕНЬШЕНИЕ РАЗМЕРА ПРОГРАММЫ НЕ ДОЛЖНО ПРОИЗВОДИТЬСЯ В УЩЕРБ ПОЛНОТЕ ОБНАРУЖЕНИЯ НЕИСПРАВНОСТЕЙ. ЕСТЕСТВЕННО ТАКЖЕ, ЧТО РАЗМЕР ПРОГРАММЫ НЕ ДОЛЖЕН ПРЕВЫШАТЬ МАКСИМАЛЬНЫЙ ОБЕСПЕЧИВАЕМЫЙ СИСТЕМОЙ ОБЪЕМ ПАМЯТИ.

2.5. ВРЕМЯ ВЫПОЛНЕНИЯ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ ПРЕДСТАВЛЯЕТ СОБОЙ ПЕРИОД ОТ НАЧАЛА ДО ОКОНЧАНИЯ ОДНОГО ПРОХОДА ТЕСТА. ПРОХОДОМ ТЕСТА МОЖЕТ БЫТЬ ВЫПОЛНЕНИЕ ВСЕХ ТЕСТОВ ДЛЯ КАЖДОГО ВЫБРАННОГО ПРОВЕРЯЕМОГО УСТРОЙСТВА (ПОСЛЕДОВАТЕЛЬНЫЙ ТЕСТ) ИЛИ ВЫПОЛНЕНИЕ ВСЕХ ТЕСТОВ ДЛЯ ВСЕХ ВЫБРАННЫХ ПРОВЕРЯЕМЫХ УСТРОЙСТВ (ПАРАЛЛЕЛЬНЫЙ ТЕСТ).

ВРЕМЯ ВЫПОЛНЕНИЯ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ ОПРЕДЕЛЯЕТСЯ, В ПЕРВУЮ ОЧЕРЕДЬ, ХАРАКТЕРИСТИКАМИ И ТРЕБОВАНИЯМИ, ПРЕДЪЯВЛЯЕМЫМИ К ТЕСТИРОВАНИЮ УСТРОЙСТВА. ТЕСТЫ ЛОГИЧЕСКИХ СХЕМ ВЫПОЛНЯЮТСЯ ОБЫЧНО С ВЫСТРОИТЕЛЬНЫМ МАШИНЫ, ПОЭТОМУ В ТЕЧЕНИЕ НЕСКОЛЬКИХ СЕКУНД МОЖНО ВЫПОЛНИТЬ БОЛЬШОЕ ЧИСЛО ПРОХОДОВ.

ТЕСТЫ ЭЛЕКТРОМЕХАНИЧЕСКОГО ОБОРУДОВАНИЯ ИЛИ ТЕСТЫ С ЗАЦИКЛИВАНИЕМ ДАННЫХ (НАПРИМЕР, ТЕСТЫ ПОЗИЦИОНИРОВАНИЯ ГОЛОВКИ НАКОПИТЕЛЯ НА МАГНИТНЫХ ДИСКАХ) ВЫПОЛНЯЮТСЯ ЗА НЕСКОЛЬКО МИНУТ.

ТЕСТИРОВАНИЕ УСТРОЙСТВ ПАМЯТИ (ДИСКА ИЛИ ЛЕНТЫ) МОЖЕТ ПОТРЕБОВАТЬ НА КАЖДЫЙ ПРОХОД НЕСКОЛЬКО МИНУТ ИЛИ ДАЖЕ ДЕСЯТКОВ МИНУТ.

ДЛЯ БОЛЬШИХ НАКОПИТЕЛЕЙ НА МАГНИТНЫХ ДИСКАХ ИЛИ НА МАГНИТНОЙ ЛЕНТЕ ОБЫЧНЫМ ЯВЛЯЕТСЯ 30 МИНУТ ОДНОГО ПРОХОДА ТЕСТА.



13.00078-01 33 01

### 3. ДИАГНОСТИЧЕСКИЙ СУПЕРВИЗОР. ОПИСАНИЕ ВХОДНОГО ЯЗЫКА

ДИАГНОСТИЧЕСКИЙ СУПЕРВИЗОР ЯВЛЯЕТСЯ ОСНОВНОЙ УПРАВЛЯЮЩЕЙ ПРОГРАММОЙ МСПД. СУПЕРВИЗОР В ПРОЦЕССЕ ДИАГНОСТИРОВАНИЯ НАХОДИТСЯ В ПАМЯТИ ВМЕСТЕ С ДИАГНОСТИЧЕСКОЙ ПРОГРАММОЙ.

ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ, РАБОТАЮЩИЕ СОВМЕСТНО С СУПЕРВИЗОРОМ, ОФОРМЛЯЮТСЯ В ВИДЕ СОВОКУПНОСТИ ПОДПРОГРАММ, ВЫЗОВ КОТОРЫХ ОСУЩЕСТВЛЯЕТСЯ ДИАГНОСТИЧЕСКИМ СУПЕРВИЗОРОМ. В ЧАСТНОСТИ, СУПЕРВИЗОР УПРАВЛЯЕТ ПОДПРОГРАММАМИ ИНИЦИАЛИЗАЦИИ, ОЧИСТКИ, СТАТИСТИЧЕСКОГО ОТЧЕТА, РАСШИРЕННЫХ СООБЩЕНИЙ ОБ ОШИБКАХ, А ТАКЖЕ ТЕСТОВЫМИ ПОДПРОГРАММАМИ, ВХОДЯЩИМИ В ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ.

В ТО ЖЕ ВРЕМЯ В СОСТАВЕ СУПЕРВИЗОРА ИМЕЕТСЯ ШИРОКИЙ НАБОР ПОДПРОГРАММ, КОТОРЫЕ ВЫЗЫВАЮТСЯ ДИАГНОСТИЧЕСКОЙ ПРОГРАММОЙ, НАПРИМЕР, ПОДПРОГРАММЫ ПЕЧАТИ ЗАГОЛОВКА СООБЩЕНИЯ ОБ ОШИБКЕ, УПРАВЛЕНИЯ ПАМЯТЬЮ, ВЫДАЧИ ЗАПРОСОВ ОПЕРАТОРУ, ИНТЕРПРЕТАТОРА КОМАНДНОЙ СТРОКИ И Т.Д.

ОБРАЩЕНИЕ ДИАГНОСТИЧЕСКИХ ПРОГРАММ К ПОДПРОГРАММАМ СУПЕРВИЗОРА ПРОИЗВОДИТСЯ С ПОМОЩЬЮ МАКРОКОМАНД, КОТОРЫЕ ХРАНЯТСЯ В ДИАГНОСТИЧЕСКОЙ МАКРО БИБЛИОТЕКЕ. В ЭТОЙ БИБЛИОТЕКЕ ИМЕЕТСЯ ТАКЖЕ РЯД ДРУГИХ МАКРОКОМАНД, ПОЗВОЛЯЮЩИХ УПРАВЛЯТЬ ВЫПОЛНЕНИЕМ ДИАГНОСТИЧЕСКИХ ПРОГРАММ В СООТВЕТСТВИИ С ФЛАГАМИ КОМАНД СУПЕРВИЗОРА И ТАКЖЕ ВЫПОЛНЯТЬ НЕКОТОРЫЕ ДРУГИЕ ФУНКЦИИ.

СУПЕРВИЗОР УПРАВЛЯЕТ ВЫПОЛНЕНИЕМ ДИАГНОСТИЧЕСКИХ ПРОГРАММ УРОВНЕЙ Z, 2R И 3, И РЕАЛИЗУЕТ СПЕЦИАЛЬНЫЙ ПРОГРАММНЫЙ ИНТЕРФЕЙС ДЛЯ ВЗАИМОДЕЙСТВИЯ С НИМИ.

ПРОГРАММЫ УРОВНЯ 3 ВЫПОЛНЯЮТСЯ В АВТОНОМНОМ РЕЖИМЕ. ПРОГРАММЫ УРОВНЯ 2R ВЫПОЛНЯЮТСЯ ТОЛЬКО ПОД УПРАВЛЕНИЕМ ОПЕРАЦИОННОЙ СИСТЕМЫ ИОС ЭП.

19.00078-01 33 01

ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ УРОВНЯ 2 МОГУТ ВЫПОЛНЯТЬСЯ КАК В АВТОНОМНОМ РЕЖИМЕ, ТАК И ТОД УПРАВЛЕНИЕМ ОПЕРАЦИОННОЙ СИСТЕМЫ.

В РАСПОРЯЖЕНИИ ОПЕРАТОРА СМ 1700 ИМЕЕТСЯ НАБОР КОМАНД ДИАГНОСТИЧЕСКОГО СУПЕРВИЗОРА, ПОЗВОЛЯЮЩИИ ВЫПОЛНЯТЬ СЛЕДУЮЩИЕ ФУНКЦИИ:

1) УПРАВЛЕНИЕ ПОСЛЕДОВАТЕЛЬНОСТЬЮ ВЫПОЛНЕНИЯ ПРОГРАММ И ТЕСТОВ;

2) ОБРАЩЕНИЕ К КОМАНДНЫМ ФАЙЛАМ;

3) УПРАВЛЕНИЕ ВЫПОЛНЕНИЕМ;

4) ОТЛАДКА ПРОГРАММ.

В ПОСЛЕДУЮЩИХ ПОДРАЗДЕЛАХ ПРИВОДЯТСЯ ОБЩИЕ СВЕДЕНИЯ О КОМАНДАХ ДИАГНОСТИЧЕСКОГО СУПЕРВИЗОРА. ПОДРОБНОЕ ОПИСАНИЕ ВХОДНОГО ЯЗЫКА ДАНО В [1].

### 3.1. КОМАНДЫ УПРАВЛЕНИЯ ПОСЛЕДОВАТЕЛЬНОСТЬЮ ВЫПОЛНЕНИЯ ПРОГРАММ И ТЕСТОВ

В ГРУППУ ВХОДЯТ КОМАНДЫ, ВЫПОЛНЯЮЩИЕ, В ЧАСТНОСТИ, СЛЕДУЮЩИЕ ОПЕРАЦИИ:

1) УСТАНОВКА УСТРОЙСТВА, С КОТОРОГО СУПЕРВИЗОР ДОЛЖЕН ЗАГРУЖАТЬ ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ;

2) ЗАГРУЗКА ФАЙЛА В ОПЕРАТИВНУЮ ПАМЯТЬ;

3) ПРИСОЕДИНЕНИЕ И ВЫБОР ПРОВЕРЯЕМЫХ УСТРОЙСТВ;

4) ЗАПУСК ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ;

5) ВЫВОД НА ТЕРМИНАЛ СТАТИСТИЧЕСКОГО ОТЧЕТА О ХОДЕ ВЫПОЛНЕНИЯ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ;

6) ПЕРЕВОД ДИАГНОСТИЧЕСКОГО СУПЕРВИЗОРА В РЕЖИМ ОЖИДАНИЯ КОМАНД;

13.00078-01 33 31

## 7) ПРЕКРАЩЕНИЕ ВЫПОЛНЕНИЯ ПРОГРАММЫ.

### 3.2. КОМАНДА ОБРАЩЕНИЯ К КОМАНДНЫМ ФАЙЛАМ

КОМАНДА ОБЕСПЕЧИВАЕТ ВЫПОЛНЕНИЕ СУПЕРВИЗОРОМ ТЕХ КОМАНД, КОТОРЫЕ БЫЛИ ПРЕДВАРИТЕЛЬНО ЗАПИСАНЫ В КОМАНДНЫЙ ФАЙЛ, ЧТО ПОЗВОЛЯЕТ ОПЕРАТОРУ ВЫЧИСЛИТЕЛЬНОЙ МАШИНЫ АВТОМАТИЧЕСКИ ВЫПОЛНЯТЬ ЗАРАНЕЕ ОПРЕДЕЛЕННУЮ ПОСЛЕДОВАТЕЛЬНОСТЬ КОМАНД. КОМАНДЫ СУПЕРВИЗОРА, КОТОРЫЕ ОБЫЧНО ЗВОДЯТСЯ С ТЕРМИНАЛА ОПЕРАТОРА, ПОСТУПАЮТ ПРИ ЭТОМ ИЗ СПЕЦИАЛЬНОГО КОМАНДНОГО ФАЙЛА.

### 3.3. КОМАНДА УПРАВЛЕНИЯ ВЫПОЛНЕНИЕМ

В ГРУППУ ВХОДЯТ КОМАНДЫ УСТАНОВКИ И СБРОСА ФЛАГОВ КОМАНД И ФЛАГОВ СОБЫТИЙ, НА ОСНОВЕ КОТОРЫХ ИЗМЕНЯЕТСЯ ХОД ВЫПОЛНЕНИЯ ДИАГНОСТИЧЕСКИХ ПРОГРАММ, ЧТО СПОСОБСТВУЕТ УПРОЩЕНИЮ И УСКОРЕНИЮ ПОИСКА НЕИСПРАВНОСТЕЙ ОБОРУДОВАНИЯ.

### 3.4. ФЛАГИ КОМАНД

ДИАГНОСТИЧЕСКИЙ СУПЕРВИЗОР ИСПОЛЬЗУЕТ СЛЕДУЮЩИЕ ФЛАГИ КОМАНД:

HALT - ОСТАНОВКА ПРИ ОБНАРУЖЕНИИ ОШИБКИ. ЕСЛИ УСТАНОВЛЕН ЭТОТ ФЛАГ И ПРОГРАММА ЗАФИКСИРУЕТ ОШИБКУ, СУПЕРВИЗОР ПОСЛЕ ВЫДАЧИ СООБЩЕНИЯ ОБ ОШИБКЕ ПЕРЕХОДИТ В СОСТОЯНИЕ ОЖИДАНИЯ КОМАНДЫ ОПЕРАТОРА. ПОСЛЕ ЭТОГО ОПЕРАТОР МОЖЕТ ПРОДОЛЖИТЬ, ПОВТОРНО ЗАПУСТИТЬ ИЛИ ПРЕКРАТИТЬ ВЫПОЛНЕНИЕ ПРОГРАММЫ. ДАННЫЙ ФЛАГ ИМЕЕТ ПРИОРИТЕТ ПЕРЕД ФЛАГОМ ЗАЦИКЛИВАНИЯ (LOOP).

LOOP - ЗАЦИКЛИВАНИЕ НА ОШИБКЕ. УСТАНОВКА ЭТОГО ФЛАГА

13.00078-01 33 01

ПРИВОДИТ К ЗАЦИКЛИВАНИЮ ОПРЕДЕЛЕННОЙ ЧАСТИ ПРОГРАММЫ. ЗАЦИКЛИВАНИЕ ПРОИСХОДИТ НА ТЕСТЕ ИЛИ НА ПОДТЕСТЕ, ЗАФИКСИРОВАННОМ ОШИБКУ.

ПРИ НЕОБХОДИМОСТИ ЗАПРЕТА ВЫДАЧИ СООБЩЕНИИ ОБ ОШИБКАХ СЛЕДУЕТ УСТАНОВИТЬ ФЛАГ IET. ЗАЦИКЛИВАНИЕ ПРОДОЛЖАЕТСЯ ДО ТЕХ ПОР ПОКА ОПЕРАТОР НЕ ВЫДАСТ КОМАНДУ CONTROL C, ПО КОТОРОЙ УПРАВЛЕНИЕ БУДЕТ ВОЗВРАЩЕНО СУПЕРВИЗОРУ. ЗАТЕМ ОПЕРАТОР МОЖЕТ ОЧИСТИТЬ ФЛАГ LOOP И ПРОДОЛЖИТЬ РАБОТУ ИЛИ ПРЕКРАТИТЬ ВЫПОЛНЕНИЕ СООТВЕТСТВУЮЩЕЙ ПРОГРАММЫ.

BELL - ЗВУКОВОЙ СИГНАЛ ПРИ ОШИБКЕ. ЕСЛИ ЭТОТ ФЛАГ УСТАНОВЛЕН, СУПЕРВИЗОР ПОДАЕТ ЗВУКОВОЙ СИГНАЛ ПРИ ОБНАРУЖЕНИИ ОШИБКИ.

IET - ЗАПРЕТ СООБЩЕНИИ ОБ ОШИБКАХ НА УРОВНЕ 1. УСТАНОВКА ЭТОГО ФЛАГА ПРИВОДИТ К ЗАПРЕТУ ВЫВОДА ВСЕХ СООБЩЕНИИ ОБ ОШИБКАХ, ЗА ИСКЛЮЧЕНИЕМ ТЕХ, КОТОРЫЕ ОБЯЗАТЕЛЬНО ВЫДАЮТСЯ ПРОГРАММОЙ ИЛИ СУПЕРВИЗОРОМ.

IET - ЗАПРЕТ СООБЩЕНИИ ОБ ОШИБКАХ НА УРОВНЕ 2. ПРИ УСТАНОВКЕ ЭТОГО ФЛАГА ЗАПРЕЩАЕТСЯ ВЫВОД ОСНОВНОЙ И РАСШИРЕННОЙ ИНФОРМАЦИИ ОБ ОШИБКЕ. ДЛЯ КАЖДОЙ ЗАФИКСИРОВАННОЙ ОШИБКИ ВЫВОДИТСЯ ТОЛЬКО ЗАГОЛОВОК СООБЩЕНИЯ ОБ ОШИБКЕ (ПЕРВЫЕ 3 СТРОКИ).

IET - ЗАПРЕТ СООБЩЕНИИ ОБ ОШИБКАХ НА УРОВНЕ 3. ЕСЛИ ЭТОТ ФЛАГ УСТАНОВЛЕН, ЗАПРЕЩАЕТСЯ ВЫВОД РАСШИРЕННОЙ ИНФОРМАЦИИ ОБ ОШИБКЕ. ЗАГОЛОВОК И ОСНОВНАЯ ИНФОРМАЦИЯ ОБ ОШИБКЕ ВЫВОДЯТСЯ ПРИ ЭТОМ ДЛЯ КАЖДОЙ ОШИБКИ.

IET - ЗАПРЕТ ВЫВОДА ИТОГОВОГО ОТЧЕТА. ПРИ УСТАНОВКЕ ДАННОГО ФЛАГА ЗАПРЕЩАЕТСЯ ВЫВОД СТАТИСТИЧЕСКОГО ОТЧЕТА.

QUICK - БЫСТРАЯ ПРОВЕРКА. ПРИ УСТАНОВКЕ ДАННОГО ФЛАГА В ДИАГНОСТИЧЕСКОЙ ПРОГРАММЕ ИСКЛЮЧАЕТСЯ ВЫПОЛНЕНИЕ НЕКОТОРЫХ (ЗАДАННЫХ В ПРОГРАММЕ) ЧАСТЕЙ И ТЕСТОВ С ЦЕЛЬЮ УСКОРЕНИЯ

ИХ ВЫПОЛНЕНИЯ.

TRACE - СООБЩЕНИЕ О ВЫПОЛНЕНИИ КАЖДОГО ТЕСТА. УСТАНОВКА ДАННОГО ФЛАГА ПРИВОДИТ К ТОМУ, ЧТО СУПЕРВИЗОР СООБЩАЕТ О ВЫПОЛНЕНИИ КАЖДОГО ТЕСТА.

OPERATOR - ПРИСУТСТВИЕ ОПЕРАТОРА. ЭТОТ ФЛАГ ГОВОРИТ СУПЕРВИЗОРУ О ТОМ, ЧТО ВОЗМОЖЕН ДИАЛОГ С ОПЕРАТОРОМ. ЕСЛИ ЭТОТ ФЛАГ НЕ УСТАНОВЛЕН, ТО ИЗ ПРОГРАММЫ (ЗАРАНЕЕ ЗАДАННЫМ В НЕЙ ОБРАЗОМ) ИСКЛЮЧАЮТСЯ ЧАСТИ, В КОТОРЫХ ВЕДЕТСЯ ДИАЛОГ С ОПЕРАТОРОМ.

PROMPT - ВОСПРОИЗВЕДЕНИЕ РАСШИРЕННОГО ДИАЛОГА. ПРИ УСТАНОВКЕ ЭТОГО ФЛАГА ВЫВОДЯТСЯ ВСЕ ОГРАНИЧЕНИЯ И УМОЛЧАНИЯ, КОТОРЫЕ ИСПОЛЗУЮТСЯ В ДИАЛОГЕ С ОПЕРАТОРОМ.

ALL - ВСЕ ФЛАГИ ДАННОГО СПИСКА.

ПО УМОЛЧАНИЮ УСТАНОВЛЕННЫ ФЛАГИ OPERATOR И PROMPT.

### 3.5. ФЛАГИ СОБЫТИЙ

ОПЕРАЦИОННАЯ СИСТЕМА МОС ВЛ И ДИАГНОСТИЧЕСКИЙ СУПЕРВИЗОР ДОПУСКАЮТ ДОПОЛНИТЕЛЬНО ИСПОЛЬЗОВАНИЕ 23-Х ФЛАГОВ СОБЫТИЙ (НОМЕРА ФЛАГОВ ОТ 1 ДО 23).

ОПРЕДЕЛЕННЫ ЗНАЧЕНИЯ ТОЛЬКО ДВУХ ФЛАГОВ (1 И 2). ЕСЛИ ФЛАГ СОБЫТИЯ 1 УСТАНОВЛЕН (ОЧИЩЕН), ТО РАЗРЕШАЕТСЯ (ЗАПРЕЩАЕТСЯ) РЕГИСТРАЦИЯ ОШИБОК ПРИ РАБОТЕ ПОД УПРАВЛЕНИЕМ МОС ВД.

ЕСЛИ ФЛАГ СОБЫТИЯ 2 УСТАНОВЛЕН (ОЧИЩЕН), ТО РАЗРЕШАЕТСЯ (ЗАПРЕЩАЕТСЯ) ПОСТОРНЫЙ ЗАПУСК ПРОГРАММЫ МОС ВП.

У ОСТАЛЬНЫХ ФЛАГОВ СОБЫТИЯ НЕТ ПОСТОЯННОГО НАЗНАЧЕНИЯ. ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ МОГУТ ИСПОЛЬЗОВАТЬ ИХ ДЛЯ ВЫПОЛНЕНИЯ РАЗЛИЧНЫХ СИГНАЛЬНЫХ ФУНКЦИЙ, ВКЛЮЧАЯ СВЯЗЬ С ОПЕРАТОРОМ.

19.00078-01 33 01

### 3.6. КОМАНДЫ ОТЛАДКИ ПРОГРАММЫ

В ГРУППУ ВХОДЯТ КОМАНДЫ, ПОЗВОЛЯЮЩИЕ ПРОИЗВОДИТЬ ОТЛАДКУ ДИАГНОСТИЧЕСКИХ ПРОГРАММ.

КОМАНДЫ ОТЛАДКИ ПОЗВОЛЯЮТ УСТАНОВЛИВАТЬ ДО 15 ТОЧЕК ПРЕРЫВАНИЯ ДЛЯ ПРОГРАММЫ, СЧИТЫВАТЬ И ИЗМЕНЯТЬ СОДЕРЖИМОЕ ЯЧЕЕК ПАМЯТИ.

### 3.7. ФУНКЦИОНАЛЬНОЕ ОПИСАНИЕ СУПЕРВИЗОРА

ДИАГНОСТИЧЕСКИЙ СУПЕРВИЗОР (ДС) ОБЕСПЕЧИВАЕТ ВОЗМОЖНОСТЬ ВЫПОЛНЕНИЯ ДИАГНОСТИЧЕСКИХ ПРОГРАММ УРОВНЕЙ 2, 2R И 3.

В ДИАГНОСТИЧЕСКОМ СУПЕРВИЗОРЕ МОЖНО ВЫДЕЛИТЬ ДВЕ ОСНОВНЫЕ ЧАСТИ: ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС, РЕАЛИЗУЕМЫЙ ИНТЕРПРЕТАТОРОМ КОМАНДНОЙ СТРОКИ И ИНТЕРФЕЙС ПРОГРАММЫ, РЕАЛИЗУЕМЫЙ НАБОРОМ СЛУЖЕБНЫХ ПРОГРАММ.

ИНТЕРПРЕТАТОР КОМАНДНОЙ СТРОКИ ОСУЩЕСТВЛЯЕТ СВЯЗЬ МЕЖДУ ТЕРМИНАЛОМ ОПЕРАТОРА, СУПЕРВИЗОРОМ И ВЫПОЛНЯЕМОЙ ДИАГНОСТИЧЕСКОЙ ПРОГРАММОЙ, А ИНТЕРФЕЙС ПРОГРАММЫ ОБЕСПЕЧИВАЕТ СВЯЗЬ МЕЖДУ ДИАГНОСТИЧЕСКОЙ ПРОГРАММОЙ И СУПЕРВИЗОРОМ (СМ. РИС.1 ПРИЛОЖЕНИЕ 3)

ИНТЕРПРЕТАТОР КОМАНДНОЙ СТРОКИ РЕАЛИЗУЕТ КОМАНДЫ СУПЕРВИЗОРА И КОМАНДЫ ОТЛАДКИ СУПЕРВИЗОРА. ЕСЛИ ИНТЕРПРЕТАТОР КОМАНДНОЙ СТРОКИ ОЖИДАЕТ КОМАНДУ ОПЕРАТОРА, ОН ВЫВОДИТ НА ТЕРМИНАЛ СИМВОЛЫ ЗАПРОСА DS>. КОГДА ОПЕРАТОР ВВОДИТ С КЛАВИАТУРЫ КОМАНДУ, ИНТЕРПРЕТАТОР КОМАНДНОЙ СТРОКИ АНАЛИЗИРУЕТ ЕЕ И ПЕРЕДАЕТ УПРАВЛЕНИЕ СООТВЕТСТВУЮЩЕЙ ПОДПРОГРАММЕ ДЕЙСТВИЯ. ПОСЛЕ ВЫПОЛНЕНИЯ ПОДПРОГРАММЫ УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ ОБРАТНО ИНТЕРПРЕТАТОРУ КОМАНДНОЙ СТРОКИ ИЛИ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЕ. НАПРИМЕР, В ОТВЕТ НА КОМАНДУ ОТ ОПЕРА-

13.00078-01 33 01

ГОРА SET FLAG IE2 ИНТЕРПРЕТАТОР ВЫЗОВЕТ ПОДПРОГРАММУ, КОТОРАЯ УСТАНОВИТ СООТВЕТСТВУЮЩИЙ ФЛАГ. ЗАТЕМ УПРАВЛЕНИЕ ВОЗВРАЩАЕТСЯ ИНТЕРПРЕТАТОРУ, КОТОРЫЙ ВЫВОДИТ НА ТЕРМИНАЛ ОПЕРАТОРА СИМВОЛЫ ЗАПРОСА DS>. ЕСЛИ ОПЕРАТОР ВЫДАСТ КОМАНДУ START, ТО ИНТЕРПРЕТАТОР КОМАНДНОЙ СТРОКИ ВЫЗОВЕТ ДИСПЕТЧЕР СУПЕРВИЗОРА, КОТОРЫЙ "ЗАПУСТИТ" ДИАГНОСТИЧЕСКУЮ ПРОГРАММУ.

ИНТЕРФЕЙС ПРОГРАММЫ РЕАЛИЗУЕТ СЛУЖЕБНЫЕ СРЕДСТВА УПРАВЛЕНИЯ ДИАГНОСТИЧЕСКОЙ ПРОГРАММОЙ, ОБРАБОТКИ СООБЩЕНИИ, КАНАЛА И ВВОДА-ВЫВОДА. ПОДПРОГРАММА СУПЕРВИЗОРА "ДИСПЕТЧЕР" ВЫЗЫВАЕТ ПОДПРОГРАММЫ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ В НУЖНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ (ИНИЦИАЛИЗАЦИЯ, ТЕСТ1, ТЕСТ2, ....., ТЕСТN И ОЧИСТКА). В СВОЮ ОЧЕРЕДЬ УКАЗАННЫЕ ПОДПРОГРАММЫ ВЫЗЫВАЮТ ПО МЕРЕ НЕОБХОДИМОСТИ СЛУЖЕБНЫЕ СРЕДСТВА ПРОГРАММНОГО ИНТЕРФЕЙСА.

НА РИС.2 (СМ. ПРИЛОЖЕНИЕ 3) ПРИВЕДЕНА УПРОЩЕННАЯ БЛОК-СХЕМА, КОТОРАЯ ПОКАЗЫВАЕТ КАК ВЗАИМОДЕЙСТВУЮТ ДИАГНОСТИЧЕСКИЙ СУПЕРВИЗОР И ДИАГНОСТИЧЕСКАЯ ПРОГРАММА. ДАННАЯ СХЕМА ПОКАЗЫВАЕТ, В ЧАСТНОСТИ, ДВА СВОЙСТВА ДИАГНОСТИЧЕСКОЙ СИСТЕМЫ. ВО-ПЕРВЫХ, УПРАВЛЕНИЕ НАЧИНАЕТСЯ И ЗАКАНЧИВАЕТСЯ ЗАПРОСОМ DS> ИНТЕРПРЕТАТОРА КОМАНДНОЙ СТРОКИ СУПЕРВИЗОРА. ВО-ВТОРЫХ, ДИАГНОСТИЧЕСКАЯ ПРОГРАММА НЕ ЯВЛЯЕТСЯ НЕЗАВИСИМОЙ. ОНА СОСТОИТ ИЗ РЯДА ПОДПРОГРАММ (ИНИЦИАЛИЗАЦИЯ, ТЕСТ1.....), КОТОРЫЕ ПООЧЕРЕДНО ВЫЗЫВАЮТСЯ ДИСПЕТЧЕРОМ ДИАГНОСТИЧЕСКОГО СУПЕРВИЗОРА.

НА РИС.3 (СМ. ПРИЛОЖЕНИЕ 3) ПОКАЗАНО РАСПРЕДЕЛЕНИЕ ПАМЯТИ ДИАГНОСТИЧЕСКОЙ СИСТЕМЫ. ОБЛАСТЬ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ ВСЕГДА НАЧИНАЕТСЯ С ВИРТУАЛЬНОГО АДРЕСА 200, А ДИАГНОСТИЧЕСКОГО СУПЕРВИЗОРА - С АДРЕСА 10000 (АДРЕСА УКАЗАНЫ В ШЕСТНАДЦАТЕРИЧНОЙ СИСТЕМЕ СЧИСЛЕНИЯ).

#### 4. СТРУКТУРА ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ

ДИАГНОСТИЧЕСКАЯ ПРОГРАММА СОСТОИТ ИЗ МОДУЛЕЙ, КАЖДИЙ ИЗ КОТОРЫХ ПРЕДСТАВЛЯЕТ СОБОЙ ОТДЕЛЬНО ТРАНСЛИРУЕМУЮ ЧАСТЬ ПРОГРАММЫ.

ПЕРВЫЙ МОДУЛЬ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ НАЗЫВАЕТСЯ ГОЛОВНЫМ. ЗА НИМ СЛЕДУЕТ МОДУЛЬ ГЛОБАЛЬНЫХ ПОДПРОГРАММ. ДАЛЕЕ В ДИАГНОСТИЧЕСКОЙ ПРОГРАММЕ РАСПОЛАГАЮТСЯ МОДУЛИ ТЕСТОВ, ИЛИ ТЕСТОВЫЕ МОДУЛИ.

В ТЕХ СЛУЧАЯХ, КОГДА ОБЩИЙ РАЗМЕР ГЛОБАЛЬНЫХ ПОДПРОГРАММ НЕВЕЛИК, ОНИ МОГУТ ВКЛЮЧАТЬСЯ В КОНЕЦ ГОЛОВНОГО МОДУЛЯ.

В ПЕРВОЙ СТРОКЕ КАЖДОГО МОДУЛЯ ДОЛЖНА ЗАПИСЫВАТЬСЯ ДИРЕКТИВА МАКРОАССЕМБЛЕРА .TITLE, В КОТОРОЙ УКАЗЫВАЕТСЯ ИМЯ МОДУЛЯ И НАЗВАНИЕ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ, В КОТОРУЮ ВХОДИТ МОДУЛЬ.

ВТОРАЯ СТРОКА МОДУЛЯ С ПОМОЩЬЮ ДИРЕКТИВЫ .IDENT УКАЗЫВАЕТ ТЕКУЩИЙ НОМЕР ВЕРСИИ ПРОГРАММЫ.

ДАЛЕЕ В СТРОКАХ МОДУЛЯ ДОЛЖНА ЗАПИСЫВАТЬСЯ СЛЕДУЮЩАЯ ИНФОРМАЦИЯ:

1) ДИРЕКТИВЫ РАСПЕЧАТКИ ЛИСТИНГА .NLIST И .LIST С СООТВЕТСТВУЮЩИМИ АРГУМЕНТАМИ;

2) ДИРЕКТИВЫ ВКЛЮЧЕНИЯ ФАЙЛОВ МАКРОБИБЛИОТЕК, ИСПОЛЬЗУЕМЫХ ПРОГРАММАМИ МОДУЛЯ, .LIBRARY;

3) ДИРЕКТИВА ОПРЕДЕЛЕНИЯ ПЕРВОЙ ПРОГРАММНОЙ СЕКЦИИ МОДУЛЯ .PSECT;

4) МАКРОКОМАНДА НАЧАЛА МОДУЛЯ PDS\_BGNMOD, ОПРЕДЕЛЯЮЩАЯ СРЕДУ РАБОТЫ ПРОГРАММЫ И НОМЕР ПЕРВОГО ТЕСТА В МОДУЛЕ (ДЛЯ ТЕСТОВЫХ МОДУЛЕЙ).



13.00078-01 33 01

СОДЕРЖИМОЕ ПОСЛЕДУЮЩИХ СТРОК ЗАВИСИТ ОТ ТИПА МОДУЛЯ И ЕГО ФУНКЦИИ.

ОТДЕЛЬНЫЕ ЧАСТИ МОДУЛЯ СЛЕДУЕТ ПРЕДВАРЯТЬ С ПОМОЩЬЮ ДИРЕКТИВЫ `.SVTTL`, В КОТОРОЙ УКАЗЫВАЕТСЯ НАИМЕНОВАНИЕ СООТВЕТСТВУЮЩЕЙ ЧАСТИ.

В ПРЕДПОСЛЕДНЕЙ СТРОКЕ МОДУЛЯ ДОЛЖНА ЗАПИСЫВАТЬСЯ МАКРОКОМАНДА КОНЦА МОДУЛЯ `ENDMOD`.

ПОСЛЕДНЯЯ СТРОКА МОДУЛЯ СОДЕРЖИТ ДИРЕКТИВУ ЗАВЕРШЕНИЯ ТРАНСЛЯЦИИ `.END`.

#### 4.1. ГОЛОВНОЙ МОДУЛЬ

ГОЛОВНОЙ МОДУЛЬ СОСТОИТ ИЗ СЛЕДУЮЩИХ ОТДЕЛОВ:

- 1) ПРЕДИСЛОВИЕ;
- 2) ОПИСАНИЯ;
- 3) ПОДПРОГРАММА ИНИЦИАЛИЗАЦИИ;
- 4) ПОДПРОГРАММА ОЧИСТКИ;
- 5) ПОДПРОГРАММА ЗАКЛЮЧИТЕЛЬНОГО ОТЧЕТА.

##### 4.1.1. ПРЕДИСЛОВИЕ МОДУЛЯ

ПРЕДИСЛОВИЕ ЯВЛЯЕТСЯ ПЕРВЫМ ОТДЕЛОМ ГОЛОВНОГО МОДУЛЯ. ПОСЛЕ СТРОКИ С МАКРОКОМАНДОЙ `ENDSVGNMOD` В НЕМ ДАЕТСЯ КРАТКОЕ ФУНКЦИОНАЛЬНОЕ ОПИСАНИЕ ПРОГРАММЫ. БОЛЕЕ ПОДРОБНОЕ ОПИСАНИЕ ПРОГРАММЫ ДОЛЖНО ПРИВОДИТЬСЯ В ОТДЕЛЬНОМ ОПИСАНИИ ПРОГРАММЫ, КОТОРОЕ ПРИЛАГАЕТСЯ К ЛИСТИНГУ В КАЧЕСТВЕ ПРЕДИСЛОВИЯ.

ДАЛЕЕ УКАЗЫВАЕТСЯ АВТОР (ПРОГРАММИСТ), ДАТА НАПИСАНИЯ ПРОГРАММЫ И НОМЕР ЕЕ ВЕРСИИ.

ЗАТЕМ ОПИСЫВАЕТСЯ ПОДРОБНАЯ ИСТОРИЯ ТЕКУЩЕГО ВЫПУСКА. ЭТО ОПИСАНИЕ ОПРЕДЕЛЯЕТ ВЕРСИИ, ИЗМЕНЕНИЯ, ПОСЛЕДНЮЮ ДАТУ

ИЗМЕНЕНИИ КАЖДОЙ ВЕРСИИ. В НЕМ ТАКЖЕ УКАЗЫВАЮТСЯ КОНКРЕТНЫЕ ИЗМЕНЕНИЯ, ВЫПОЛНЕННЫЕ В ТЕЧЕНИЕ РАЗРАБОТКИ ПРОГРАММЫ И ФАМИЛИИ ЛИЦ, ОТВЕТСТВЕННЫХ ЗА КАЖДУЮ ВЕРСИЮ.

ВСЯ УКАЗАННАЯ ЗДЕСЬ ИНФОРМАЦИЯ ДОЛЖНА БЫТЬ ЗАПИСАНА В СТРОКАХ, НАЧИНАЮЩИХСЯ С СИМВОЛА 2.

#### 4.1.2. ОПИСАНИЯ МОДУЛЯ

В ЭТОМ ОТДЕЛЕ СОДЕРЖАТСЯ ДАННЫЕ ДЛЯ ВСЕЙ ПРОГРАММЫ. ОН ВКЛЮЧАЕТ СЛЕДУЮЩИЕ ЭЛЕМЕНТЫ:

- 1) ОПРЕДЕЛЕНИЯ ЛОКАЛЬНЫХ МАКРОКОМАНД;
- 2) ОПЕРАТОРЫ ПРЯМОГО ПРИСВАИВАНИЯ;
- 3) ПРОГРАММНАЯ СЕКЦИЯ ЗАГОЛОВКА ПРОГРАММЫ #HEADER;
- 4) ПРОГРАММНАЯ СЕКЦИЯ ДИСПЕТЧЕРА ПРОГРАММЫ DISPATCH;
- 5) ПРОГРАММНАЯ СЕКЦИЯ ГЛОБАЛЬНЫХ ДАННЫХ;
- 6) ТЕКСТОВАЯ ЧАСТЬ ПРОГРАММЫ.

#### 4.1.3. ОПРЕДЕЛЕНИЯ ЛОКАЛЬНЫХ МАКРОКОМАНД

В ЧАСТИ ЗАПИСЫВАЮТСЯ ОПРЕДЕЛЕНИЯ МАКРОКОМАНД, КОТОРЫЕ ИСПОЛЬЗУЮТСЯ ЛИШЬ В ДАННОЙ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЕ. ОНИ РАЗРАБАТЫВАЮТСЯ СОСТАВИТЕЛЕМ ДАННОЙ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ.

#### 4.1.4. ОПЕРАТОРЫ ПРЯМОГО ПРИСВАИВАНИЯ

ОПЕРАТОРЫ В ЭТОЙ ЧАСТИ ПРОГРАММЫ ПРИСВАИВАЮТ ЗНАЧЕНИЯ СИМВОЛИЧЕСКИМ ИМЕНАМ, КОТОРЫЕ ИСПОЛЬЗУЮТСЯ МАКРОКОМАНДАМИ И ИНСТРУКЦИЯМИ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ. УКАЗАННЫЕ ОПЕРАТОРЫ ЗАДАЮТСЯ ИЛИ НЕПОСРЕДСТВЕННО ИЛИ С ПОМОЩЬЮ МАКРОКОМАНД ДИАГНОСТИЧЕСКОЙ И СИСТЕМНЫХ МАКРОБИБЛИОТЕК.

13.00078-01 33 01

БИБЛИОТЕЧНЫЕ МАКРОКОМАНДЫ ПРЯМОГО ПРИСВАИВАНИЯ ИМЕЮТ СЛЕДУЮЩИЙ ВИД:

Ч...DEF

#### 4.1.5. ПРОГРАММНАЯ СЕКЦИЯ ЗАГОЛОВКА ПРОГРАММЫ (#HEADER)

В НЕЙ ЗАПИСЫВАЮТСЯ ПАРАМЕТРЫ, ПОЗВОЛЯЮЩИЕ ДИАГНОСТИЧЕСКОМУ СУПЕРВИЗОРУ УПРАВЛЯТЬ ПРОГРАММОЙ. УКАЗАННАЯ ПРОГРАММНАЯ СЕКЦИЯ ФОРМИРУЕТСЯ МАКРОКОМАНДОЙ #DS\_HEADER И СОДЕРЖИТ СЛЕДУЮЩИЕ СВЕДЕНИЯ:

- 1) ИМЯ ПРОГРАММЫ;
- 2) НОМЕР УРОВНЯ ВЫПУСКА ПРОГРАММЫ;
- 3) НОМЕР УРОВНЯ ИЗМЕНЕНИЙ;
- 4) МАКСИМАЛЬНОЕ ЧИСЛО УСТРОЙСТВ, КОТОРЫЕ МОГУТ ТЕСТИРОВАТЬСЯ ОДНОВРЕМЕННО;
- 5) ЧИСЛО ТИПОВ ОШИБОК, КОТОРЫЕ МОГУТ ВСТРЕТИТЬСЯ В ТЕСТИРУЕМОМ УСТРОЙСТВЕ;
- 6) УКАЗАТЕЛЬ НАЛИЧИЯ СТАТИСТИЧЕСКОЙ ТАБЛИЦЫ;
- 7) УКАЗАТЕЛЬ ТАБЛИЦЫ ДИСПЕТЧЕРА ТЕСТОВ;
- 8) УКАЗАТЕЛЬ СПИСКА ТИПОВ УСТРОЙСТВ;
- 9) УКАЗАТЕЛЬ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ;
- 10) УКАЗАТЕЛЬ ПОДПРОГРАММЫ ОЧИСТКИ;
- 11) УКАЗАТЕЛЬ СПИСКА АДРЕСОВ ИМЕН СЕКЦИЙ И НЕКОТОРЫЕ ДРУГИЕ.

ПРОГРАММНАЯ СЕКЦИЯ #HEADER РАСПОЛАГАЕТСЯ В ПАМЯТИ НАЧИНАЯ С ВИРТУАЛЬНОГО АДРЕСА 200 (ШЕСТНАДЦАТЕРИЧНЫЙ).

13.00078-01 33 01

#### 4.1.6. ПРОГРАММНАЯ СЕКЦИЯ ДИСПЕТЧЕРА ПРОГРАММЫ (DISPATCH)

СЕКЦИЯ СОЗДАЕТСЯ МАКРОКОМАНДОЙ `WDS_DISPATCH`, КОТОРАЯ ГЕНЕРИРУЕТ МЕТКУ НАЧАЛА ТАБЛИЦЫ ДИСПЕТЧЕРА (`DISPATCH`) И ПРИЗНАК КОНЦА ТАБЛИЦЫ ДИСПЕТЧЕРА (6 ДЛИННЫХ СЛОВ С НУЛЯМИ). ЗАПИСИ В ТАБЛИЦЕ ДИСПЕТЧЕРА ФОРМИРУЮТСЯ МАКРОКОМАНДАМИ `WDS_VGNTEST`, КАЖДАЯ ИЗ КОТОРЫХ ЗАПИСЫВАЕТ В ТАБЛИЦУ ДИСПЕТЧЕРА СЛЕДУЮЩУЮ ИНФОРМАЦИЮ:

- 1) АДРЕС СЛОВА, В КОТОРОМ НАХОДИТСЯ НОМЕР ТЕСТА;
- 2) АДРЕС СТРОКИ, СОДЕРЖАЩЕЙ НАЗВАНИЕ ТЕСТА;
- 3) АДРЕС МАССИВА ВХОДНЫХ ДАННЫХ ДЛЯ ТЕСТА;
- 4) АДРЕС ТОЧКИ ВХОДА В ТЕСТОВУЮ ПОДПРОГРАММУ;
- 5) МАСКА ТЕСТОВЫХ СЕКЦИЙ, В КОТОРЫЕ ВХОДИТ ДАННЫЙ ТЕСТ.

#### 4.1.7. ПРОГРАММНАЯ СЕКЦИЯ ГЛОБАЛЬНЫХ ДАННЫХ ПРОГРАММЫ

СЕКЦИЯ СОДЕРЖИТ ВСЕ МОДИФИЦИРУЕМЫЕ В ДИНАМИЧЕСКОМ РЕЖИМЕ ДАННЫЕ, АДРЕСНЫЕ УКАЗАТЕЛИ И БУФЕРЫ.

СЕКЦИЯ СОДЕРЖИТ ДВЕ ТАБЛИЦЫ:

1) ТАБЛИЦА СОДЕРЖИМОГО РЕГИСТРОВ УСТРОЙСТВ, В КОТОРОЙ ЗАПОМИНАЕТСЯ СОДЕРЖИМОЕ РЕГИСТРОВ УСТРОЙСТВ. ДЛЯ УКАЗАНИЯ НАЧАЛА ТАБЛИЦЫ ИСПОЛЬЗУЕТСЯ МАКРОКОМАНДА `WDS_VGNREG`, КОТОРАЯ ГЕНЕРИРУЕТ МЕТКУ НАЧАЛА ТАБЛИЦЫ `DEV_REG`. В КОНЦЕ ТАБЛИЦЫ ЗАПИСЫВАЕТСЯ МАКРОКОМАНДА `WDS_ENDREG`.

2) СТАТИСТИЧЕСКАЯ ТАБЛИЦА, В КОТОРОЙ ЗАПИСЫВАЮТСЯ СВЕДЕНИЯ ОБ АППАРАТНЫХ И ПРОГРАММНЫХ ОШИБКАХ. ДЛЯ КАЖДОГО ПРОВЕРЯЕМОГО УСТРОЙСТВА ДОЛЖНА СОЗДАВАТЬСЯ ОТДЕЛЬНАЯ ПОДТАБЛИЦА. ИНФОРМАЦИИ, ХРАНЯЩУЮСЯ В СТАТИСТИЧЕСКОЙ ТАБЛИЦЕ, ИСПОЛЬЗУЕТ ПОДПРОГРАММА ЗАКЛЮЧИТЕЛЬНОГО ОТЧЕТА ПРИ РАСПЕ-

13.00078-01 33 01'

ЧАТКЕ СТАТИСТИЧЕСКОГО ОТЧЕТА. ДЛЯ УКАЗАНИЯ НАЧАЛА СТАТИСТИЧЕСКОЙ ТАБЛИЦЫ ИСПОЛЬЗУЕТСЯ МАКРОКОМАНДА #DS\_BGNSTAT, ФОРМИРУЮЩАЯ МЕТКУ НАЧАЛА ТАБЛИЦЫ STATISTIC. В КОНЦЕ ТАБЛИЦЫ ЗАПИСЫВАЕТСЯ МАКРОКОМАНДА #DS\_ENDSTAT.

#### 4.1.8. ТЕКСТОВАЯ ЧАСТЬ ПРОГРАММЫ

В НЕЙ ЗАПИСЫВАЮТСЯ СТРОКИ СИМВОЛОВ В КОДЕ КОИ-8. В ЧАСТНОСТИ, ЗДЕСЬ УКАЗЫВАЕТСЯ СЛЕДУЮЩАЯ ИНФОРМАЦИЯ:

1) ТАБЛИЦА ИМЕН ТЕСТОВЫХ СЕКЦИЙ В КОДЕ КОИ-8, КОТОРЫЕ ИСПОЛЬЗУЮТСЯ МАКРОКОМАНДАМИ #DS\_BGNTEST ДЛЯ УКАЗАНИЯ СУПЕРВИЗОРУ, КАКИЕ ТЕСТОВЫЕ СЕКЦИИ ОТНОСЯТСЯ К КАЖДОМУ ИЗ ТЕСТОВ. УКАЗАННАЯ ТАБЛИЦА ФОРМИРУЕТСЯ МАКРОКОМАНДОЙ #DS\_SECTION;

2) ТАБЛИЦА ТИПОВ УСТРОЙСТВ, В КОТОРОЙ ПЕРЕЧИСЛЕННЫ ИМЕНА ВСЕХ УСТРОЙСТВ, ИЗВЕСТНЫХ ПРОГРАММЕ. ДЛЯ ФОРМИРОВАНИЯ ТАБЛИЦЫ НЕОБХОДИМО ИСПОЛЬЗОВАТЬ МАКРОКОМАНДУ #DS\_DEVTYPE;

3) СИМВОЛЬНЫЕ СТРОКИ В КОДЕ КОИ-8, СОДЕРЖАЩИЕ СООБЩЕНИЯ ДЛЯ ОПЕРАТОРА;

4) СИМВОЛЬНЫЕ СТРОКИ В КОДЕ КОИ-8, В КОТОРЫХ ПРИВЕДЕНЫ ТЕКСТЫ ВСЕХ СООБЩЕНИЙ ОБ ОШИБКАХ.

#### 4.1.9. ПОДПРОГРАММА ИНИЦИАЛИЗАЦИИ

ПОДПРОГРАММА ИНИЦИАЛИЗАЦИИ СЛУЖИТ ДЛЯ НАЧАЛЬНОЙ УСТАНОВКИ ПРОВЕРЯЕМЫХ УСТРОЙСТВ. ОНА ПРОИЗВОДИТ РЯД ПРЕДВАРИТЕЛЬНЫХ ДЕЙСТВИЙ, КОТОРЫЕ НЕОБХОДИМЫ ДЛЯ ВЫПОЛНЕНИЯ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ, НАПРИМЕР, НАЗНАЧЕНИЕ КАНАЛА ДЛЯ КАЖДОГО Тестируемого УСТРОЙСТВА, ОЧИСТКА СУФЕРОВ ПАМЯТИ И СЧЕТЧИКОВ.

13.00078-01 33 01

ПОДПРОГРАММА ИНИЦИАЛИЗАЦИИ ЯВЛЯЕТСЯ ПЕРВОЙ ЧАСТЬЮ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ, ВЫЗЫВАЕМОЙ ДИСПЕТЧЕРОМ СУПЕРВИЗОРА. ОНА РАБОТАЕТ КАЖДЫЙ РАЗ ПЕРЕД НАЧАЛОМ ВЫПОЛНЕНИЯ ПОСЛЕДОВАТЕЛЬНОСТИ ТЕСТОВ.

ПОДПРОГРАММА ИНИЦИАЛИЗАЦИИ ПРОВЕРЯЕТ ЗАВЕРШЕНИЕ ПОСЛЕДНЕГО ПРОХОДА С ПОМОЩЬЮ МАКРОКОМАНДЫ СУПЕРВИЗОРА `ADS_ENDPASS_X`. ПОСЛЕ ОКОНЧАНИЯ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ УПРАВЛЕНИЕ ВОЗВРАЩАЕТСЯ ДИСПЕТЧЕРУ СУПЕРВИЗОРА, КОТОРЫЙ ВЫЗЫВАЕТ СЛЕДУЮЩУЮ ПОДПРОГРАММУ (НАПРИМЕР, `ТЕСТ1`).

ИЗ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ НЕ ДОЛЖНЫ ВЫЗЫВАТЬСЯ ПОДПРОГРАММЫ ПЕЧАТИ (ВЫВОДА), ЗА ИСКЛЮЧЕНИЕМ СЛУЧАЕВ ОБНАРУЖЕНИЯ НЕУСТРАНИМОЙ ОШИБКИ УСТРОЙСТВА ИЛИ СИСТЕМЫ. ПРИ ОБНАРУЖЕНИИ НЕУСТРАНИМОЙ ОШИБКИ СИСТЕМЫ ДЛЯ ПЕЧАТИ НУЖНО ИСПОЛЬЗОВАТЬ МАКРОКОМАНДУ `ADS_ERRSYS_X`, А ПРИ ОБНАРУЖЕНИИ НЕУСТРАНИМОЙ ОШИБКИ УСТРОЙСТВА `ADS_ERRDEV_X`.

ПРОГРАММА, ПРЕДНАЗНАЧЕННАЯ ДЛЯ ПРОВЕРКИ БОЛЕЕ ЧЕМ ОДНОГО УСТРОЙСТВА, МОЖЕТ БЫТЬ ОРГАНИЗОВАНА ТАКИМ ОБРАЗОМ, ЧТОБЫ УСТРОЙСТВА ПРОВЕРЯЛИСЬ ЛИБО ПОСЛЕДОВАТЕЛЬНО ОДНО ЗА ДРУГИМ (ПОСЛЕДОВАТЕЛЬНОЕ ТЕСТИРОВАНИЕ) ИЛИ НЕСКОЛЬКО УСТРОЙСТВ ОДНОВРЕМЕННО (ПАРАЛЛЕЛЬНОЕ ТЕСТИРОВАНИЕ). ПОСТРОЕНИЕ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ ЗАВИСИТ ОТ ТОГО, КАКОЙ СПОСОБ ТЕСТИРОВАНИЯ ИСПОЛЬЗУЕТ ДИАГНОСТИЧЕСКАЯ ПРОГРАММА. В НАЧАЛЕ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ ЗАПИСЫВАЕТСЯ МАКРОКОМАНДА `ADS_BGNINIT`, ГЕНЕРИРУЮЩАЯ ПРОГРАММНУЮ СЕКЦИЮ, В КОТОРОЙ РАЗМЕЩАЕТСЯ ПОДПРОГРАММА, И МЕТКУ НАЧАЛА ПОДПРОГРАММЫ `INITIALIZE`. В КОНЦЕ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ ЗАПИСЫВАЕТСЯ МАКРОКОМАНДА `ADS_ENDINIT`, ФОРМИРУЮЩАЯ МЕТКУ КОНЦА ПОДПРОГРАММЫ `INITIALIZE_X` И ИНСТРУКЦИЮ ВЫХОДА ИЗ ПОДПРОГРАММЫ.

НА РИС.4 (СМ. ПРИЛОЖЕНИЕ 3) ИЗОБРАЖЕНА БЛОК-СХЕМА ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ ДЛЯ ПАРАЛЛЕЛЬНОГО ТЕСТИРОВАНИЯ

УСТРОЙСТВ, А НА РИС.5 (СМ. ПРИЛОЖЕНИЕ 3) - ДЛЯ ПОСЛЕДОВАТЕЛЬНОГО ТЕСТИРОВАНИЯ.

#### 4.1.10. ПОДПРОГРАММА ОЧИСТКИ

ПОДПРОГРАММА ОЧИСТКИ ВЫЗЫВАЕТСЯ ДИСПЕТЧЕРОМ СУПЕРВИЗОРА ПОСЛЕ ВЫПОЛНЕНИЯ ПОСЛЕДНЕГО ТЕСТА В ПОСЛЕДНЕМ ПРОХОДЕ ПРОГРАММЫ, А ТАКЖЕ В ТЕХ СЛУЧАЯХ, КОГДА ПРОГРАММА ЗАКАНЧИВАЕТСЯ НЕНОРМАЛЬНО.

ПОДПРОГРАММА ОЧИСТКИ ВЫЗЫВАЕТ ИНИЦИАЛИЗАЦИЮ ПРОВЕРЯЕМЫХ УСТРОЙСТВ БЕЗ ПРОВЕРКИ ОШИБОК. ОНА ЗАПРЕЩАЕТ ПЕРЕДАЧИ ДАННЫХ, ОСВОБОЖДАЕТ БУФЕРЫ ПАМЯТИ И ОТМЕНЯЕТ ТАЙМЕРЫ. В ДИАГНОСТИЧЕСКИХ ПРОГРАММАХ УРОВНЯ 2 ПРОГРАММА ОЧИСТКИ ОТМЕНЯЕТ НАЗНАЧЕНИЯ КАНАЛОВ УСТРОЙСТВАМ.

В НАЧАЛЕ ПОДПРОГРАММЫ ОЧИСТКИ ЗАПИСЫВАЕТСЯ МАКРОКОМАНДА `ADS_BEGINCLEAN`, ФОРМИРУЮЩАЯ ПРОГРАММНУЮ СЕКЦИЮ, В КОТОРОЙ РАСПОЛАГАЕТСЯ ПОДПРОГРАММА ОЧИСТКИ, И МЕТКУ НАЧАЛА ПОДПРОГРАММЫ `CLEAN_UP`.

В КОНЦЕ ПОДПРОГРАММЫ ЗАПИСЫВАЕТСЯ МАКРОКОМАНДА `ADS_ENDCLEAN`, КОТОРАЯ ГЕНЕРИРУЕТ МЕТКУ КОНЦА `CLEAN_UP_X` И ИНСТРУКЦИЮ ВЫХОДА ИЗ ПОДПРОГРАММЫ.

НА РИС.6 (СМ. ПРИЛОЖЕНИЕ 3) ИЗОБРАЖЕНА БЛОК-СХЕМА ТИПИЧНОЙ ПОДПРОГРАММЫ ОЧИСТКИ ПРОГРАММЫ УРОВНЯ 2.

#### 4.1.11. ПОДПРОГРАММА ЗАКЛЮЧИТЕЛЬНОГО (ИТОГОВОГО) ОТЧЕТА

ПОДПРОГРАММА ИТОГОВОГО ОТЧЕТА ВЫДАЕТ ОПЕРАТОРУ ИНФОРМАЦИЮ, ОПИСЫВАЮЩУЮ РАБОТУ ПРОГРАММЫ. ОПЕРАТОР МОЖЕТ В ЛЮБОЙ МОМЕНТ ПРЕРВАТЬ ВЫПОЛНЕНИЕ ПРОГРАММЫ, ЧТОБЫ ПОЛУЧИТЬ ЗАКЛЮЧИТЕЛЬНЫЙ ОТЧЕТ, А ЗАТЕМ ЛИБО ПРОДОЛЖИТЬ ЛИБО ПРЕКРАТИТЬ

19.00078-01 33 01

ВЫПОЛНЕНИЕ СООТВЕТСТВУЮЩЕЙ ПРОГРАММЫ.

ПОДПРОГРАММА ЗАКЛЮЧИТЕЛЬНОГО ОТЧЕТА НЕ ЯВЛЯЕТСЯ ОБЯЗАТЕЛЬНОЙ. ИТОГОВЫЙ ОТЧЕТ, НАПРИМЕР, НЕ НУЖЕН ДЛЯ ТЕХ ПРОГРАММ, КОТОРЫЕ ВЫПОЛНЯЮТСЯ ЗА НЕСКОЛЬКО СЕКУНД.

В НАЧАЛЕ ПОДПРОГРАММЫ ИТОГОВОГО ОТЧЕТА ЗАПИСЫВАЕТСЯ МАКРОКОМАНДА `ADS_BGN SUMMARY`, ГЕНЕРИРУЮЩАЯ ПРОГРАММНУЮ СЕКЦИЮ, В КОТОРОЙ РАЗМЕЩАЕТСЯ ПОДПРОГРАММА, И МЕТКУ НАЧАЛА ПОДПРОГРАММЫ `SUMMARY`.

В КОНЦЕ ПОДПРОГРАММЫ ЗАПИСЫВАЕТСЯ МАКРОКОМАНДА `ADS_END SUMMARY`, КОТОРАЯ ГЕНЕРИРУЕТ МЕТКУ КОНЦА `SUMMARY_X` И ИНСТРУКЦИЮ ВЫХОДА ИЗ ПОДПРОГРАММЫ.

НА РИС.7 (СМ. ПРИЛОЖЕНИЕ 3) ИЗОБРАЖЕНА БЛОК-СХЕМА ТИПИЧНОЙ ЗАКЛЮЧИТЕЛЬНОЙ ПОДПРОГРАММЫ. ЭТА ПОДПРОГРАММА ОБЕСПЕЧИВАЕТ РАСПЕЧАТКУ СТАТИСТИЧЕСКОЙ ИНФОРМАЦИИ ДЛЯ КАЖДОГО ПРОВЕРЯЕМОГО УСТРОЙСТВА ПРИ УСЛОВИИ ДОСТОВЕРНОСТИ ЭТОЙ СТАТИСТИКИ.

#### 4.1.12. ДОКУМЕНТИРОВАНИЕ ПОДПРОГРАММ ИНИЦИАЛИЗАЦИИ, ОЧИСТКИ И ЗАКЛЮЧИТЕЛЬНОЙ ПОДПРОГРАММЫ

ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ, ОЧИСТКИ И ЗАКЛЮЧИТЕЛЬНОГО ОТЧЕТА ДОЛЖНЫ СОДЕРЖАТЬ ПРЕДИСЛОВИЕ ПОДПРОГРАММЫ, КОММЕНТАРИИ ДЛЯ КАЖДОГО ЛОГИЧЕСКОГО БЛОКА И СТРОКИ КОММЕНТАРИЕВ.

ПРЕДИСЛОВИЕ ПОДПРОГРАММЫ ВКЛЮЧАЕТ СЛЕДУЮЩИЕ ЭЛЕМЕНТЫ:

1) ДИРЕКТИВА `SBTTL`, В КОТОРОЙ ЗАПИСЫВАЕТСЯ ИМЯ ПОДПРОГРАММЫ;

2) ФУНКЦИОНАЛЬНОЕ ОПИСАНИЕ ПОДПРОГРАММЫ;

3) ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫЗОВА ПОДПРОГРАММ.

КАЖДАЯ ПОДПРОГРАММА ДОЛЖНА ИМЕТЬ ОДНУ ТОЧКУ ВХОДА И



ОДНУ ТОЧКУ ВЫХОДА.

ТОЧКА ВХОДА ИМЕЕТ МЕТКУ, УКАЗЫВАЮЩУЮ ПЕРВУЮ ЯЧЕЙКУ В ПОДПРОГРАММЕ.

ПЕРВАЯ ЯЧЕЙКА СОДЕРЖИТ МАСКУ СОХРАНЕНИЯ РЕГИСТРОВ, ИСПОЛЬЗУЕМЫХ ПОДПРОГРАММОЙ.

ТОЧКА ВЫХОДА ИМЕЕТ МЕТКУ, КОТОРАЯ УКАЗЫВАЕТ ИНСТРУКЦИЮ ВОЗВРАТА (RET).

4.1.13. ПРИМЕР ОФОРМЛЕНИЯ ГОЛОВНОГО МОДУЛЯ

```

.TITLE ZZ-EVAAA1 ТЕСТИРОВАНИЕ ПЕЧАТАЮЩЕГО УСТРОЙСТВА
;НАЗВАНИЕ ПРОГРАММЫ
.IDENT /5.3/ ;НОМЕР ВЕРСИИ
.LIST MEV ;ДИРЕКТИВЫ РАСПЕЧАТКИ
.NLIST CND ;ЛИСТИНГА
.LIBRARY /SYS=LIBRARY:DIAG/ ;ВКЛЮЧЕНИЕ ФАЙЛА
;ДИАГНОСТИЧЕСКОЙ
;МАКРОБИБЛИОТЕКИ
.PSECT HEADER,PAGE,NOWRT ;ОПРЕДЕЛЕНИЕ ПРОГРАММНОЙ
;СЕКЦИИ
#DS_BGNMOD SEP_FUNCTIONAL ;НАЧАЛО МОДУЛЯ
;КРАТКОЕ ОПИСАНИЕ
;ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ
.PAGE ;ПЕРЕХОД НА НОВУЮ СТРАНИЦУ
;СВТTL ИСТОРИЯ ;СВЕДЕНИЯ ОБ ИЗМЕНЕНИЯХ
;В ПРОГРАММЕ
;МАКРООПРЕДЕЛЕНИЯ: ;МАКРООПРЕДЕЛЕНИЯ ДАННОЙ
;ПРОГРАММЫ
.PAGE
;СВТTL ОПРЕДЕЛЕНИЕ СИМВОЛИЧЕСКИХ ИМЕН
#DS_DSSDEF
;СВТTL ОПРЕДЕЛЕНИЕ СИМВОЛИЧЕСКИХ ИМЕН
#DS_DSADEF
.PAGE
;СВТTL ДАННЫЕ БЛОКА ЗАГОЛОВКА ПРОГРАММЫ
#DS_HEADER <ТЕСТИРОВАНИЕ ПЕЧАТАЮЩЕГО УСТРОЙСТВА>5,3,10
;ФОРМИРОВАНИЕ ТАБЛИЦЫ
;ЗАГОЛОВКА
.PAGE
;СВТTL ТАБЛИЦА ДИСПЕТЧЕРА
;КРАТКОЕ ОПИСАНИЕ
;ТАБЛИЦЫ ДИСПЕТЧЕРА
#DS_DISPATCH ;ОПРЕДЕЛЕНИЕ ПРОГРАММНОЙ
;СЕКЦИИ ДИСПЕТЧЕРА
.PAGE
;СВТTL СЕКЦИЯ ГЛОБАЛЬНЫХ ДАННЫХ

```

19.00078-01 33 01

```
.PSECT DATA,PAGE
.....
MDS_BGNREG
.....
;ЗАПОМИНАНИЕ СОДЕРЖИМОГО
;РЕГИСТРОВ УСТРОЙСТВ

MDS_ENDREG
.....
MDS_BGNSTAT
.....
;СТАТИСТИЧЕСКАЯ ТАБЛИЦА
.....
MDS_ENDSTAT
.PAGE
.SBTTL ТЕКСТОВАЯ ЧАСТЬ
.....
MDS_SECTION MANUAL,HALT,FUNCTION ;ФОРМИРОВАНИЕ ТАБЛИЦЫ
..... ;ИМЕН ТЕСТОВЫХ СЕКЦИИ
MDS_DEVTYPE <LP05,LP06>,<T_LP27,Г_LN01>
.....
;ТАБЛИЦА ТИПОВ УСТРОЙСТВ
;ТЕКСТЫ СООБЩЕНИЙ
;ОБ ОШИБКАХ
;СООБЩЕНИЯ ДЛЯ ОПЕРАТОРА
.PAGE
.SBTTL ПОДПРОГРАММА ИНИЦИАЛИЗАЦИИ
;.....
;ОПИСАНИЕ ФУНКЦИИ
;ПОДПРОГРАММЫ

MDS_BGNINIT <R2,R3>
.....
;ПОДПРОГРАММА ИНИЦИАЛИЗАЦИИ
MDS_ENDINIT ОЧИСТКИ
.PAGE
.SBTTL ПОДПРОГРАММА
.....
;ОПИСАНИЕ ФУНКЦИИ
;ПОДПРОГРАММЫ

MDS_BGNCLEAN <R3,R4,R5>
.....
;ПОДПРОГРАММА ОЧИСТКИ
MDS_ENDCLEAN
.PAGE
.SBTTL ПОДПРОГРАММА ИТОГОВОГО ОТЧЕТА
.....
;ОПИСАНИЕ ПОДПРОГРАММЫ
MDS_BGNSUMMARY
.....
;ПОДПРОГРАММА
;ЗАКЛЮЧИТЕЛЬНОГО ОТЧЕТА
.PAGE
MDS_ENDSUMMARY
MDS_ENDMOD
;КОНЕЦ МОДУЛЯ
.END
```

#### 4.2. МОДУЛЬ ГЛОБАЛЬНЫХ ПОДПРОГРАММ

ОБЫЧНО ГЛОБАЛЬНЫЕ ПОДПРОГРАММЫ ГРУППИРУЮТСЯ В ОТДЕЛЬНОМ МОДУЛЕ. ЕСЛИ ЖЕ ОБЩИИ РАЗМЕР ГЛОБАЛЬНЫХ ПОДПРОГРАММ НЕВЕЛИК, ТО ОНИ ВКЛЮЧАЮТСЯ В ГОЛОВНОЙ МОДУЛЬ.

ГЛОБАЛЬНЫЕ ПОДПРОГРАММЫ ДОЛЖНЫ ИСПОЛЬЗОВАТЬСЯ ДЛЯ ТЕХ ФУНКЦИЙ, КОТОРЫЕ ВЫПОЛНЯЮТСЯ ТЕСТОВЫМИ ПРОГРАММАМИ БОЛЕЕ ОДНОГО РАЗА.

19.00078-01 33 01'

КАЖДАЯ ГЛОБАЛЬНАЯ ПОДПРОГРАММА ДОЛЖНА ИМЕТЬ ТОЛЬКО ОДНУ ТОЧКУ ВХОДА И ОДНУ ТОЧКУ ВЫХОДА.

ГЛОБАЛЬНЫЕ ПОДПРОГРАММЫ ОБЫЧНО ВЫПОЛНЯЮТ СЛЕДУЮЩИЕ ФУНКЦИИ:

- 1) РАСПЕЧАТКА СОСТОЯНИЯ КАНАЛА ПРИ ОБНАРУЖЕНИИ ОШИБКИ КАНАЛА;
- 2) РАСПЕЧАТКА ОЖИДАЕМЫХ И ПОЛУЧЕННЫХ ДАННЫХ;
- 3) РАСПЕЧАТКА СОДЕРЖИМОГО РЕГИСТРОВ УСТРОЙСТВА;
- 4) СРАВНЕНИЕ ДАННЫХ;
- 5) ПРОВЕРКА СОСТОЯНИЯ КАНАЛА;
- 6) ОБРАБОТКА ПРЕРЫВАНИИ УСТРОЙСТВА.

ПАРАМЕТРЫ, ПЕРЕДАВАЕМЫЕ ВЫЗЫВАЮЩЕЙ ПРОГРАММОЙ ГЛОБАЛЬНОЙ ПОДПРОГРАММЕ, ОБЫЧНО ЗАНОСЯТСЯ В СТЕК ИСТОЧНИКОМ ВЫЗОВА В ПОРЯДКЕ, КОТОРЫЙ ОПРЕДЕЛЯЕТСЯ ГЛОБАЛЬНОЙ ПОДПРОГРАММОЙ.

ГЛОБАЛЬНАЯ ПОДПРОГРАММА ВЫБИРАЕТ ЭТИ ПАРАМЕТРЫ ПУТЕМ ПРИБАВЛЕНИЯ СМЕДЕНИЙ К УКАЗАТЕЛЮ АРГУМЕНТОВ И ПЕРЕСЫЛКИ СООТВЕТСТВУЮЩИХ ДАННЫХ В ОБЩИЕ РЕГИСТРЫ.

ПРИМЕР.

```

;ВЫЗЫВАЮЩАЯ ПРОГРАММА
TEST_N:= PUSHL R9           ;ОЖИДАЕМЫЕ ДАННЫЕ
          PUSHL R8           ;ПОЛУЧЕННЫЕ ДАННЫЕ
          PUSHAL ADR_NAME    ;АДРЕС РЕГИСТРА ИМЕНИ
          CALLS #3,PRINT_EX_REC ;ВЫЗОВ ПОДПРОГРАММЫ
;ГЛОБАЛЬНАЯ ПОДПРОГРАММА
PRINT_EX_REC:=
  .WORD M<R2,R3,R4,AR5,R6,R7>;СОХРАНЕНИЕ
          ;СОДЕРЖИМОГО РЕГИСТРОВ
  MOVL 4(AP),R2             ;СОХРАНЕНИЕ АДРЕСА
          ;РЕГИСТРА ИМЕНИ
  MOVL 8(AP),R3             ;СОХРАНЕНИЕ ПОЛУЧЕННЫХ
          ;ДАННЫХ
  MOVWL 12(AP),R4          ;СОХРАНЕНИЕ ОЖИДАЕМЫХ
          ;ДАННЫХ
  .
  .
  .
  RET                       ;ВОЗВРАТ К ВЫЗЫВАЮЩЕЙ
          ;ПРОГРАММЕ
```

13.00078-01 33 01

В ПРИМЕРЕ ПОКАЗАНО КАК ПЕРЕДАЮТСЯ ПАРАМЕТРЫ С ПОМОЩЬЮ ИНСТРУКЦИИ CALLS.

ДААННЫЕ В СТЕКЕ РАЗМЕЩАЮТСЯ В СЛЕДУЮЩЕМ ПОРЯДКЕ:

- 1) УКАЗАТЕЛЬ АРГУМЕНТОВ - 4(АР);
- 2) АДРЕС ИМЕНИ РЕГИСТРА - 4(АР);
- 3) ПОЛУЧЕННЫЕ ДАННЫЕ - 8(АР);
- 4) ОЖИДАЕМЫЕ ДАННЫЕ - 12(АР).

ОБЩИЕ РЕГИСТРЫ, ИСПОЛЗУЕМЫЕ ГЛОБАЛЬНОЙ ПОДПРОГРАММОЙ (РЕГИСТРЫ R2-R7) СОХРАНЯЮТСЯ В НАЧАЛЕ ПОДПРОГРАММЫ С ПОМОЩЬЮ МАСКИ СОХРАНЕНИЯ СОДЕРЖИМОГО РЕГИСТРОВ. ИНСТРУКЦИЯ RET В КОНЦЕ ГЛОБАЛЬНОЙ ПОДПРОГРАММЫ ВОССТАНАВЛИВАЕТ СОДЕРЖИМОЕ ЭТИХ РЕГИСТРОВ.

ЕСЛИ ГЛОБАЛЬНАЯ ПОДПРОГРАММА ДОЛЖНА ПЕРЕДАВАТЬ ДАННЫЕ ВЫЗЫВАЮЩЕЙ ПРОГРАММЕ, ТО ВЫЗЫВАЮЩАЯ ПРОГРАММА УКАЗЫВАЕТ БУФЕР, В КОТОРЫИ ГЛОБАЛЬНАЯ ПОДПРОГРАММА ДОЛЖНА ЗАПИСАТЬ ЭТИ ДАННЫЕ.

КАК И ГОЛОВНОЙ МОДУЛЬ, МОДУЛЬ ГЛОБАЛЬНЫХ ПОДПРОГРАММ ДОЛЖЕН ИМЕТЬ ПРЕДИСЛОВИЕ И ОПЕРАТОРЫ ПРЯМОГО ПРИСВАИВАНИЯ.

КАЖДАЯ ГЛОБАЛЬНАЯ ПОДПРОГРАММА ДОЛЖНА ИМЕТЬ СВОЕ ПРЕДИСЛОВИЕ И ДОСТАТОЧНОЕ ЧИСЛО КОММЕНТАРИЕВ.

#### 4.3. ТЕСТОВЫЕ МОДУЛИ

ТЕСТЫ ДИАГНОСТИЧЕСКИХ ПРОГРАММ УРОВНЕЙ 2, 2R И 3 РАСПОЛАГАЮТСЯ В ОДНОМ ИЛИ НЕСКОЛЬКИХ МОДУЛЯХ, НАЗЫВАЕМЫХ ТЕСТОВЫМИ МОДУЛЯМИ.

КАЖДЫИ ТЕСТОВЫИ МОДУЛЬ ДОЛЖЕН СОДЕРЖАТЬ СЛЕДУЮЩИЕ ОТДЕЛЫ:

- 1) ПРЕДИСЛОВИЕ МОДУЛЯ;
- 2) ОПЕРАТОРЫ ПРЯМОГО ПРИСВАИВАНИЯ;

### 3) ТЕСТОВЫЕ ПОДПРОГРАММЫ.

#### 4.3.1. ПРЕДИСЛОВИЕ МОДУЛЯ

В ПРЕДИСЛОВИИ МОДУЛЯ ДОЛЖЕН УКАЗЫВАТЬСЯ АВТОР ПРОГРАММЫ, ДАТА ЕЕ НАПИСАНИЯ И НОМЕР ВЕРСИИ. ДОЛЖНА ОПИСЫВАТЬСЯ ПОДРОБНАЯ ИСТОРИЯ ТЕКУЩЕГО ВЫПУСКА. ЭТО ОПИСАНИЕ ОПРЕДЕЛЯЕТ ВЕРСИИ, ИЗМЕНЕНИЯ, ПОСЛЕДНЮЮ ДАТУ ИЗМЕНЕНИЙ КАЖДОЙ ВЕРСИИ. В НЕМ ТАКЖЕ УКАЗЫВАЮТСЯ КОНКРЕТНЫЕ ИЗМЕНЕНИЯ, ВЫПОЛНЕННЫЕ В ТЕЧЕНИЕ РАЗРАБОТКИ ПРОГРАММЫ И ФАМИЛИИ ЛИЦ, ОТВЕТСТВЕННЫХ ЗА КАЖДУЮ ВЕРСИЮ.

ВСЯ ИНФОРМАЦИЯ ПРЕДИСЛОВИЯ ДОЛЖНА БЫТЬ ЗАПИСАНА В СТРОКАХ, НАЧИНАЮЩИХСЯ С СИМВОЛА ; .

#### 4.3.2. ОПЕРАТОРЫ ПРЯМОГО ПРИСВАИВАНИЯ

ОПЕРАТОРЫ В ЭТОЙ ЧАСТИ МОДУЛЯ ПРИСВАИВАЮТ ЗНАЧЕНИЯ СИМВОЛИЧЕСКИМ ИМЕНАМ, КОТОРЫЕ ИСПОЛЬЗУЮТСЯ МАКРОКОМАНДАМИ И ИНСТРУКЦИЯМИ МОДУЛЯ. УКАЗАННЫЕ ОПЕРАТОРЫ ЗАДАЮТСЯ ИЛИ НЕПОСРЕДСТВЕННО ИЛИ С ПОМОЩЬЮ МАКРОКОМАНД ДИАГНОСТИЧЕСКОЙ ИЛИ СИСТЕМНЫХ МАКРОБИБЛИОТЕК.

В ЧАСТНОСТИ, СИМВОЛИЧЕСКИЕ ИМЕНА ТЕСТОВЫХ СЕКЦИЙ ДОЛЖНЫ БЫТЬ ОПРЕДЕЛЕННЫ С ПОМОЩЬЮ МАКРОКОМАНДЫ #DS\_SECTION.

#### 4.3.3. ТЕСТОВЫЕ ПОДПРОГРАММЫ

КАЖДАЯ ТЕСТОВАЯ ПОДПРОГРАММА СООТВЕТСТВУЕТ ОДНОМУ ТЕСТУ.

ТЕСТОВАЯ ПОДПРОГРАММА ОФОРМЛЯЕТСЯ ТАКИМ ОБРАЗОМ, ЧТОБЫ ОНА МОГЛА БЫТЬ ВЫЗВАНА ДИСПЕТЧЕРОМ СУПЕРВИЗОРА.

ТЕСТ МОЖЕТ БЫТЬ ЧАСТЬЮ БОЛЕЕ, ЧЕМ ОДНОЙ ТЕСТОВОЙ СЕК-

19.00078-01 33 01

ЦИИ. ЖЕЛАТЕЛЬНО, ЧТОБЫ КАЖДЫЙ ТЕСТ ИМЕЛ ОДНУ ТОЧКУ ВХОДА И ОДНУ ТОЧКУ ВЫХОДА, Т.К. ТАКАЯ СТРУКТУРА УПРОЩАЕТ ПРОЦЕДУРУ ОТЛАДКИ. КАЖДЫЙ ТЕСТ МОЖЕТ СОСТОЯТЬ ИЗ ОДНОГО ИЛИ НЕСКОЛЬКИХ ПОДТЕСТОВ (НАЛИЧИЕ ПОДТЕСТОВ НЕ ОБЯЗАТЕЛЬНО).

КАЖДЫЙ ТЕСТ В ДИАГНОСТИЧЕСКОЙ ПРОГРАММЕ ДОЛЖЕН БЫТЬ НЕЗАВИСИМЫМ ОТ ДРУГИХ ТЕСТОВ, ОДНАКО НА ПОДТЕСТЫ ЭТО ТРЕБОВАНИЕ НЕ РАСПРОСТРАНЯЕТСЯ. НАПРИМЕР, ТРЕТИЙ ПОДТЕСТ В ДАННОМ ТЕСТЕ МОЖЕТ ЗАВИСЕТЬ ОТ ФУНКЦИЙ, ВЫПОЛНЯЕМЫХ В ДВУХ ПРЕДЫДУЩИХ ПОДТЕСТАХ. ОДНАКО КАЖДЫЙ ПОДТЕСТ ДОЛЖЕН БЫТЬ ПОСТРОЕН ТАКИМ ОБРАЗОМ, ЧТОБЫ ИМЕЛАСЬ ВОЗМОЖНОСТЬ ЕГО МНОГОКРАТНОГО ВЫПОЛНЕНИЯ (ЗАЦИКЛИВАНИЯ) ПОСЛЕ ОДНОКРАТНОГО ВЫПОЛНЕНИЯ ПРЕДЫДУЩИХ ПОДТЕСТОВ.

КАЖДЫЙ ТЕСТ ДОЛЖЕН ИМЕТЬ ПРЕДИСЛОВИЕ, В КОТОРОМ ОПИСЫВАЕТСЯ ЭТОТ ТЕСТ И ОБЪЯСНЯЕТСЯ, КАКИЕ ФУНКЦИОНАЛЬНЫЕ ИЛИ ЛОГИЧЕСКИЕ ОБЛАСТИ СЧИТАЮТСЯ РАБОТАЮЩИМИ ПРАВИЛЬНО ПРИ ЗАПУСКЕ ТЕСТА.

ДОЛЖНО БЫТЬ ТАКЖЕ ДАНО ОПИСАНИЕ КАЖДОГО ПОДТЕСТА. В ЭТОМ ОПИСАНИИ НЕОБХОДИМО УКАЗАТЬ, КАКИЕ ОБЛАСТИ ПРОВЕРЯЕТ ПОДТЕСТ, КАК ОН РАБОТАЕТ И КАКАЯ ПРОЦЕДУРА ЛОКАЛИЗАЦИИ НЕИСПРАВНОСТИ ДОЛЖНА БЫТЬ ИСПОЛЬЗОВАНА ПРИ ОБНАРУЖЕНИИ ПОДТЕСТОМ ОШИБКИ.

ПЕРВЫМ ОПЕРАТОРОМ КАЖДОГО ТЕСТА ДОЛЖНА БЫТЬ МАКРОКОМАНДА #DS\_SBTIL, КОТОРАЯ УКАЗЫВАЕТ НАЗВАНИЕ ТЕСТА И ЕГО НОМЕР.

В СЛУЧАЕ, ЕСЛИ ОДИН И ТОТ ЖЕ ТЕСТ ДОЛЖЕН ВЫПОЛНЯТЬСЯ НЕСКОЛЬКО РАЗ С РАЗЛИЧНЫМИ ДАННЫМИ, СЛЕДУЮЩИМ ОПЕРАТОРОМ ПОСЛЕ #DS\_SBTIL ДОЛЖНА БЫТЬ МАКРОКОМАНДА #DS\_#GNDATA, ОПРЕДЕЛЯЮЩАЯ НАЧАЛО ДАННЫХ ДЛЯ ТЕСТА.

НЕПОСРЕДСТВЕННО ПОСЛЕ ВЫЗОРА #DS\_#GNDATA ДОЛЖЕН ЗАПИСЫВАТЬСЯ МАССИВ ДАННЫХ ДЛЯ ТЕСТА. МАССИВ СОСТОИТ ИЗ НЕС-

КОЛЬКИХ СПИСКОВ АРГУМЕНТОВ, ОПРЕДЕЛЯЕМЫХ СОСТАВИТЕЛЕМ ТЕСТОВОЙ ПОДПРОГРАММЫ. КАЖДОМУ ВЫПОЛНЕНИЮ ТЕСТА СООТВЕТСТВУЕТ ОДИН СПИСОК.

СПИСОК АРГУМЕНТОВ ДОЛЖЕН СОСТОЯТЬ ИЗ ДЛИННЫХ СЛОВ, ПРИЧЕМ В ПЕРВОМ ДЛИННОМ СЛОВЕ СПИСКА НУЖНО УКАЗАТЬ ЧИСЛО АРГУМЕНТОВ В СПИСКЕ.

ПОСЛЕ ОКОНЧАНИЯ МАССИВА ДАННЫХ ДЛЯ ТЕСТОВ ДОЛЖНА БЫТЬ ЗАПИСАНА МАКРОКОМАНДА #DS\_ENDDATA, КОТОРАЯ ВЫРАБАТЫВАЕТ ДЛЯ СУПЕРВИЗОРА ПРИЗНАК ОКОНЧАНИЯ МАССИВА ДАННЫХ ДЛЯ ТЕСТА.

ЕСЛИ НЕТ НЕОБХОДИМОСТИ ЗАДАНИЯ МАССИВА ДАННЫХ ДЛЯ ТЕСТА, МАКРОКОМАНДЫ #DS\_BGNDDATA И #DS\_ENDDATA НЕ ИСПОЛЬЗУЮТСЯ.

ДАЛЕЕ ДОЛЖНА ЗАПИСЫВАТЬСЯ МАКРОКОМАНДА #DS\_BGNTEST, КОТОРАЯ УКАЗЫВАЕТ НАЧАЛО ТЕСТОВОЙ ПОДПРОГРАММЫ. ЭТА МАКРОКОМАНДА ФОРМИРУЕТ МЕТКУ ВХОДА В ТЕСТОВУЮ ПОДПРОГРАММУ, А ТАКЖЕ МАСКУ СОХРАНЕНИЯ РЕГИСТРОВ В ТОЧКЕ ВХОДА.

ПОСЛЕ МАКРОКОМАНДЫ #DS\_BGNTEST ДОЛЖНА СЛЕДОВАТЬ СОБСТВЕННО ТЕСТОВАЯ ПОДПРОГРАММА, В КОНЦЕ КОТОРОЙ ЗАПИСЫВАЕТСЯ МАКРОКОМАНДА #DS\_ENDTEST, ГЕНЕРИРУЮЩАЯ ВЫХОДНУЮ МЕТКУ ПОДПРОГРАММЫ И ИНСТРУКЦИЮ ВЫХОДА ИЗ ПОДПРОГРАММЫ (RET).

ЕСЛИ ТЕСТ РАЗБИВАЕТСЯ НА ПОДТЕСТЫ, ТО ГРАНИЦЫ КАЖДОГО ИЗ ПОДТЕСТОВ ЗАДАЮТСЯ МАКРОКОМАНДАМИ #DS\_BGNSUB (НАЧАЛО ПОДТЕСТА) И #DS\_ENDSUB (КОНЕЦ ПОДТЕСТА). ЭТИ МАКРОКОМАНДЫ МОГУТ ЗАПИСЫВАТЬСЯ ТОЛЬКО В ПРОСТРАНСТВЕ МЕЖДУ МАКРОКОМАНДАМИ #DS\_BGNTEST И #DS\_ENDTEST.

ПЕРЕД МАКРОКОМАНДАМИ #DS\_BGNSUB МОЖЕТ ЗАПИСЫВАТЬСЯ МАКРОКОМАНДА #DS\_BVTTL С УКАЗАНИЕМ НАЗВАНИЯ ПОДТЕСТА, ОДНАКО ИСПОЛЬЗОВАНИЕ ЕЕ В ЭТОМ СЛУЧАЕ НЕ ОБЯЗАТЕЛЬНО.

13.00078-01 33 01

4.3.4. ПРИМЕР ОФОРМЛЕНИЯ ТЕСТОВОГО МОДУЛЯ

```
.TITLE                ПРОВЕРКА УСТРОЙСТВА АВС
.IDENT  /3.4/
.LIST  MEB
.NLIST  CND
.LIBRARY  /SYS#LIBRARY:DIAG/           ;ВКЛЮЧЕНИЕ
                                           ;БИБЛИОТЕЧНЫХ ФАЙЛОВ
.LIBRARY  /SYS#LIBRARY:STARLET/
.....
                                           ;КРАТКОЕ ОПИСАНИЕ
                                           ;ФУНКЦИИ МОДУЛЯ

.PAGE
.SBTTL      ДЕКЛАРАЦИИ
; МАКРООПРЕДЕЛЕНИЯ :
.....
                                           ;МАКРООПРЕДЕЛЕНИЯ
                                           ;ДЛЯ МОДУЛЯ
                                           ;НАЧАЛО МОДУЛЯ
#DS_35NMOD      SER_FUNCTIONAL
; ОПРЕДЕЛЕНИЯ СИМВОЛИЧЕСКИХ ИМЕН
#DS_DSSDEF
.....
#DS_CHDEF
#DS_SECDER      FUNCTIONAL,MANUAL      ;НАЗВАНИЕ ТЕСТОВЫХ
                                           ;СЕКЦИИ

.PAGE
#DS_SBTTL <ПРОВЕРКА УСТРОЙСТВА АВС> ;НАЗВАНИЕ ПЕРВОГО ТЕСТА
.....
                                           ;КРАТКОЕ ОПИСАНИЕ
                                           ;ПЕРВОГО ТЕСТА

#DS_BGNDATA
.LONG  2,МЕТ1,МЕТ2
                                           ;ДАННЫЕ ДЛЯ ПЕРВОГО
                                           ;ВЫПОЛНЕНИЯ ТЕСТА1
.LONG  2,МЕТ3,МЕТ4
                                           ;ДАННЫЕ ДЛЯ ВТОРОГО
                                           ;ВЫПОЛНЕНИЯ ТЕСТА2

#DS_ENDDATA
#DS_BGNTTEST <FUNCTIONAL,DEFAULT>
#DS_35NSUB
ПОДТЕСТ1 ТЕСТА1
#DS_ENDSUB
#DS_SBTTL <КОНТРОЛЬ БЛОКА А>           ;ПОДЗАГОЛОВОК ПОДТЕСТА
#DS_35NSUB
ПОДТЕСТ2 ТЕСТА1
#DS_ENDSUB
#DS_ENDTEST
.PAGE
#DS_SBTTL <ПРОВЕРКА УСТРОЙСТВА <LM>> ;НАЗВАНИЕ ВТОРОГО ТЕСТА
.....
                                           ;КРАТКОЕ ОПИСАНИЕ ТЕСТА2
#DS_35NTTEST <MANUAL>
      ТЕСТ2
#DS_ENDTEST
#DS_ENDMOD
.END
```

В ПРИМЕРЕ МОДУЛЬ ВКЛЮЧАЕТ 2 ТЕСТА. ПЕРВЫИ ТЕСТ СОСТОИТ ИЗ ДВУХ ПОДТЕСТОВ. МЕЖДУ МАКРОКОМАНДАМИ #DS\_35NDATA И #DS\_ENDDATA УКАЗАНЫ ДВЕ ГРУППЫ ДАННЫХ, КОТОРЫЕ ИСПОЛЬЗУЮТСЯ ТЕСТОМ1. КАЖДАЯ ГРУППА СОСТОИТ ИЗ 2-Х ВЕЛИЧИН.



ВТОРОЙ ТЕСТ НЕ РАЗДЕЛЕН НА ПОДТЕСТЫ. ПОМИМО МАКРОКОМАНД, УКАЗАННЫХ В ПРИМЕРЕ, ПОДПРОГРАММЫ ТЕСТОВ И ПОДТЕСТОВ ОБЫЧНО ИСПОЛЬЗУЮТ И МНОГИЕ ДРУГИЕ МАКРОКОМАНДЫ ДИАГНОСТИЧЕСКОЙ И СИСТЕМНЫХ БИБЛИОТЕК.

## 5. ОПИСАНИЕ ПОДКЛЮЧЕНИЯ ПРОГРАММ ПОЛЬЗОВАТЕЛЯ

ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ СЛЕДУЕТ СТРОИТЬ ПО МОДУЛЬНОМУ ПРИНЦИПУ. МНОГИЕ ФУНКЦИИ ТЕСТОВЫХ ПРОГРАММ МОЖНО ИСКЛЮЧИТЬ ИЗ ОСНОВНОЙ ЧАСТИ ПРОГРАММЫ И ПРЕДСТАВИТЬ В ВИДЕ ПОДПРОГРАММ. ПРАВИЛЬНОЕ ИСПОЛЬЗОВАНИЕ ПОДПРОГРАММ МОЖЕТ ПРИВЕСТИ К ЗНАЧИТЕЛЬНОМУ УМЕНЬШЕНИЮ РАЗМЕРОВ ОБЪЕКТНОГО КОДА.

ПОДПРОГРАММЫ, КОТОРЫЕ МОГУТ ИСПОЛЬЗОВАТЬСЯ РАЗЛИЧНЫМИ МОДУЛЯМИ, НЕОБХОДИМО ДЕЛАТЬ ГЛОБАЛЬНЫМИ И ПОМЕЩАТЬ В ОТДЕЛЬНЫЙ МОДУЛЬ, А В СЛУЧАЕ, ЕСЛИ ОБЩИЙ РАЗМЕР ПОДПРОГРАММ НЕВЕЛИК, ИХ МОЖНО РАЗМЕСТИТЬ В ГОЛОВНОМ МОДУЛЕ.

ПОДПРОГРАММА АСИНХРОННОГО СИСТЕМНОГО ЗАХВАТА (AST) И ПРОГРАММЫ ОБСЛУЖИВАНИЯ ПРЕРЫВАНИЯ ТАКЖЕ ДОЛЖНЫ БЫТЬ ГЛОБАЛЬНЫМИ.

ДЛЯ УДОБСТВА ПОЛЬЗОВАНИЯ ПРОГРАММОЙ И УПРОЩЕНИЯ И УСКОРЕНИЯ ПОИСКА ОШИБКИ НЕОБХОДИМО, ЧТОБЫ ОТЧЕТЫ (СООБЩЕНИЯ ОБ ОШИБКАХ) ВЫДАВАЛИСЬ В ОСНОВНОЙ ЧАСТИ ТЕСТОВОЙ ПРОГРАММЫ (А НЕ ИЗ ПОДПРОГРАММЫ).

ПОДПРОГРАММЫ ДОЛЖНЫ ОБРАЩАТЬСЯ С ОШИБКАМИ ТАК ЖЕ, КАК И СЛУЖЕБНЫЕ ПОДПРОГРАММЫ, Т.Е. ВОЗВРАЩАТЬ КОДЫ СОСТОЯНИЯ. В ОСНОВНОЙ ЧАСТИ ПРОГРАММЫ ВОЗВРАЩАЕМЫЕ КОДЫ СОСТОЯНИЯ ДОЛЖНЫ ПРОВЕРЯТЬСЯ ПОСЛЕ ВЫХОДА ИЗ ПОДПРОГРАММ ДЛЯ ВЕТВЛЕНИЯ ПРОГРАММЫ.

В НАЧАЛЕ КАЖДОЙ ПОДПРОГРАММЫ СЛЕДУЕТ ИСПОЛЬЗОВАТЬ МАС-

13.00078-01 33 01

КУ СОХРАНЕНИЯ РЕГИСТРОВ, ИСПОЛЪЗУЕМЫХ В ПОДПРОГРАММЕ.

ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ СЛЕДУЕТ СТРОИТЬ ТАКИМ ОБРАЗОМ, ЧТОБЫ ПЕРИОДИЧЕСКИ (НЕ РЕЖЕ ОДНОГО РАЗА В НЕСКОЛЬКО СЕКУНД) НА ТЕРМИНАЛ ВЫДАВАЛАСЬ ИНФОРМАЦИЯ С ТЕМ, ЧТОБЫ ОПЕРАТОР МОГ ЗНАТЬ, ЧТО ПРОГРАММА ПРОДОЛЖАЕТ РАБОТАТЬ.

В ПРИЛОЖЕНИИ 2 ДАН ПРИМЕР ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ.

### 5.1. ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ УРОВНЯ 2

ДЛЯ ОБРАБОТКИ ПЕРЕДАЧ ВВОДА-ВЫВОДА НА ПРОВЕРЯЕМОЕ УСТРОЙСТВО ПРОГРАММЫ УРОВНЯ 2 ИСПОЛЪЗУЮТ СИСТЕМНЫЕ ДРАЙВЕРЫ ОПЕРАЦИОННОЙ СИСТЕМЫ МОС ВП ИЛИ ДИАГНОСТИЧЕСКИЕ ДРАЙВЕРЫ СУПЕРВИЗОРА. РЕГИСТРЫ УСТРОЙСТВА НЕПОСРЕДСТВЕННО ВЫБИРАТЬСЯ НЕ МОГУТ. СЛУЖЕБНЫЕ СРЕДСТВА КАЧАЛА СУПЕРВИЗОРА НЕ ДОЛЖНЫ ИСПОЛЪЗОВАТЬСЯ.

ПРОГРАММНЫЕ ДРАЙВЕРЫ ВВОДА-ВЫВОДА ПЕРЕДАЮТ СОДЕРЖИМОЕ РЕГИСТРОВ УСТРОЙСТВ В БУФЕР, УКАЗАННЫЙ ПРОГРАММИСТОМ. ФОРМАТ ВЫВОДА ЗАВИСИТ ОТ КОНКРЕТНОГО УСТРОЙСТВА.

ДЛЯ СЛУЖЕБНЫХ СРЕДСТВ ДОЛЖНЫ ИСПОЛЪЗОВАТЬСЯ ФЛАГИ СОБЫТИЯ В ДИАПАЗОНЕ 32-63.

ПРОГРАММЫ УРОВНЯ 2 ДОЛЖНЫ ИСПОЛЪЗОВАТЬ ФИЗИЧЕСКИЙ ВВОД-ВЫВОД.

### 5.2. ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ УРОВНЯ 2R

ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ УРОВНЯ 2R ИСПОЛЪЗУЮТСЯ, В ЧАСТНОСТИ, В ТЕХ СЛУЧАЯХ, КОГДА НЕОБХОДИМО ПРОВЕРЯТЬ УСТРОЙСТВА, ДРАЙВЕРЫ КОТОРЫХ ОТСУТСТВУЮТ В СУПЕРВИЗОРЕ И ВХОДЯТ В ОПЕРАЦИОННУЮ СИСТЕМУ.

### 5.3. ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ УРОВНЯ 3

ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ УРОВНЯ 3 ИСПОЛЬЗУЮТ НЕПОСРЕДСТВЕННЫЙ ВВОД-ВЫВОД И ПРИВИЛЕГИРОВАННЫЕ ИНСТРУКЦИИ (НАПРИМЕР, MTRR). ЭТИ СРЕДСТВА НЕДОСТУПНЫ ДЛЯ ПРОГРАММ, КОТОРЫЕ ВЫПОЛНЯЮТСЯ ПОД УПРАВЛЕНИЕМ ОПЕРАЦИОННОЙ СИСТЕМЫ. ПРОГРАММЫ УРОВНЯ 3 МОГУТ ОПРЕДЕЛЯТЬ ОШИБКИ С БОЛЬШЕЙ ТОЧНОСТЬЮ, ЧЕМ ПРОГРАММЫ УРОВНЕЙ 2, 2R.

ПРИ ОБРАЩЕНИИ К ПЕРИФЕРИЙНОМУ УСТРОЙСТВУ ПРОГРАММЫ УРОВНЯ 3 ДОЛЖНЫ ИСПОЛЬЗОВАТЬ СЛУЖЕБНЫЕ СРЕДСТВА КАНАЛА СУПЕРВИЗОРА. ПРИ НЕОБХОДИМОСТИ РАСПРЕДЕЛЕНИЯ ПАМЯТИ, СЛЕДУЕТ ПОЛЬЗОВАТЬСЯ МАКРОКОМАНДАМИ #DS\_GETBUF\_X И #DS\_RELBUF\_X.

### 5.4. ТЕСТИРОВАНИЕ РЕГИСТРОВ

ДЛЯ ПРОВЕРКИ РЕГИСТРОВ ОБЫЧНО ИСПОЛЬЗУЮТСЯ ТЕСТЫ, ОБНАРУЖИВАЮЩИЕ НЕИСПРАВНОСТИ ТИПА УСТАНОВКА БИТА В "ТОЖДЕСТВЕННЫЙ НУЛЬ" (S\_A\_0) ИЛИ В "ТОЖДЕСТВЕННУЮ ЕДИНИЦУ" (S\_A\_1), А ТАКЖЕ ЗАМЫКАНИЕ РАЗРЯДОВ МЕЖДУ СОБОЙ. ТАКИЕ ТЕСТЫ ОБЫЧНО ИСПОЛЬЗУЮТ ЗАПИСЬ И СЧИТЫВАНИЕ КОМБИНАЦИЙ С ПЛАВАЮЩИМ НУЛЕМ И ПЛАВАЮЩЕЙ ЕДИНИЦЕЙ. ТАКИЕ КОМБИНАЦИИ УКАЗЫВАЮТ ОТКАЗЫ НАГЛЯДНЫМ ДЛЯ ОПЕРАТОРА СПОСОБОМ. ОБЩЕЕ ЧИСЛО КОМБИНАЦИЙ ПРИ ЭТОМ РАВНО  $N*2$ , ГДЕ N-ЧИСЛО БИТОВ В РЕГИСТРЕ.

ДЛЯ СОКРАЩЕНИЯ ЧИСЛА КОМБИНАЦИЙ И СООТВЕТСТВЕННО ТРЕБУЕМОГО ВРЕМЕНИ ПРИ ПРОВЕРКЕ РЕГИСТРА МОГУТ БЫТЬ ИСПОЛЬЗОВАНЫ КОМБИНАЦИИ, КОТОРЫЕ МЫ РАССМОТРИМ НА ПРИМЕРЕ ПРОВЕРКИ 16-БИТОВОГО РЕГИСТРА.



## 5. МАКРОКОМАНДЫ ДИАГНОСТИЧЕСКИХ ПРОГРАММ

МАКРОКОМАНДЫ, КОТОРЫЕ ИСПОЛЗУЮТСЯ ДИАГНОСТИЧЕСКИМИ ПРОГРАММАМИ, МОЖНО РАЗДЕЛИТЬ НА 3 ЧАСТИ:

- 1) МАКРОКОМАНДЫ ДИАГНОСТИЧЕСКОЙ СИСТЕМЫ МСПД;
- 2) МАКРОКОМАНДЫ ОПЕРАЦИОННОЙ СИСТЕМЫ МОС ВП;
- 3) СОБСТВЕННЫЕ МАКРОКОМАНДЫ ДИАГНОСТИЧЕСКИХ ПРОГРАММ.

МАКРОКОМАНДЫ ДИАГНОСТИЧЕСКОЙ СИСТЕМЫ МСПД РАЗМЕЩАЮТСЯ В МАКРОБИБЛИОТЕКЕ DIAG.MLB.

МАКРОКОМАНДЫ ОПЕРАЦИОННОЙ СИСТЕМЫ МОС ВП РАЗМЕЩАЮТСЯ В ДВУХ МАКРОБИБЛИОТЕКАХ: STARLET.MLB И LIB.MLB.

СОБСТВЕННЫЕ МАКРОКОМАНДЫ ДИАГНОСТИЧЕСКИХ ПРОГРАММ ОПРЕДЕЛЯЮТСЯ СОСТАВИТЕЛЕМ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ И РАЗМЕЩАЮТСЯ ИЛИ В БИБЛИОТЕЧНОМ ФАЙЛЕ КАТАЛОГА, ИЛИ В ТЕЛЕ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ.

АЛФАВИТНЫЙ УКАЗАТЕЛЬ МАКРОКОМАНД ДАН В ПРИЛОЖЕНИИ 1.

## 7. МАКРОКОМАНДЫ ДИАГНОСТИЧЕСКОЙ СИСТЕМЫ МСПД

ОПИСЫВАЕМЫЕ В НАСТОЯЩЕМ ДОКУМЕНТЕ МАКРОКОМАНДЫ ДИАГНОСТИЧЕСКОЙ СИСТЕМЫ МОЖНО РАЗДЕЛИТЬ НА СЛЕДУЮЩИЕ КАТЕГОРИИ:

- 1) МАКРОКОМАНДЫ ФОРМАТА ПРОГРАММЫ;
- 2) МАКРОКОМАНДЫ УПРАВЛЕНИЯ ПРОГРАММОЙ;
- 3) МАКРОКОМАНДЫ АППАРАТНОЙ Р-ТАБЛИЦЫ;
- 4) МАКРОКОМАНДЫ ДИАЛОГА МЕЖДУ ПРОГРАММОЙ И ОПЕРАТОРОМ;
- 5) МАКРОКОМАНДЫ УПРАВЛЕНИЯ ПАМЯТЬЮ;
- 6) МАКРОКОМАНДЫ СООБЩЕНИЙ ОБ ОШИБКАХ;
- 7) МАКРОКОМАНДЫ ОПРЕДЕЛЕНИЯ СИМВОЛИЧЕСКИХ ИМЕН.

### 7.1. КОДИРОВКА МАКРОКОМАНД ДИАГНОСТИЧЕСКОЙ СИСТЕМЫ

НАЗВАНИЕ МАКРОКОМАНД ДИАГНОСТИЧЕСКОЙ СИСТЕМЫ ИМЕЕТ ФОРМУ `ADS_ИМЯ`, `ADS_ИМЯ_DEF`, `ADS_ИМЯ_L` ИЛИ `ADS_ИМЯ_X`, ГДЕ X - ПРИНИМАЕТ ЗНАЧЕНИЯ S И G.

ФОРМА `ADS_ИМЯ_DEF` ИСПОЛЬЗУЕТСЯ ДЛЯ МАКРОКОМАНД, ОПРЕДЕЛЯЮЩИХ СИМВОЛИЧЕСКИЕ ИМЕНА.

ФОРМА `ADS_ИМЯ_L` ИСПОЛЬЗУЕТСЯ ДЛЯ МАКРОКОМАНД, ГЕНЕРИРУЮЩИХ СПИСОК АРГУМЕНТОВ.

ФОРМА `ADS_ИМЯ_X` ИСПОЛЬЗУЕТСЯ В ТЕХ МАКРОКОМАНДАХ, КОТОРЫЕ ВЫЗЫВАЮТ СЛУЖЕБНЫЕ ПОДПРОГРАММЫ СУПЕРВИЗОРА. ПРИЧЕМ, ФОРМА `ADS_ИМЯ_S` ПРИМЕНЯЕТСЯ В ТЕХ СЛУЧАЯХ, КОГДА ВЫЗОВ ПОДПРОГРАММЫ ПРОИСХОДИТ С ПОМОЩЬЮ ИНСТРУКЦИИ `CALLS`, А `ADS_ИМЯ_G` - ПРИ ВЫЗОВЕ ИНСТРУКЦИЕЙ `CALLG`.

ФОРМА `ADS_ИМЯ` ИСПОЛЬЗУЕТСЯ В МАКРОКОМАНДАХ, ГЕНЕРИРУЮЩИХ ПРОГРАММНЫЕ СЕКЦИИ, МЕТКИ, ДИРЕКТИВЫ И ИНСТРУКЦИИ АССЕМБЛЕРА, И РЯДЕ ДРУГИХ.

13.00078-01 33 01

ВЫЗОВ МАКРОКОМАНД ИМЕЕТ СЛЕДУЮЩИИ ФОРМАТ:

НАЗВАНИЕ\_МАКРОКОМАНДЫ СПИСОК\_АРГУМЕНТОВ

ПРИ ОПИСАНИИ МАКРОКОМАНД НЕОБЯЗАТЕЛЬНЫЕ АРГУМЕНТЫ ЗАКЛЮЧАЮТСЯ В КВАДРАТНЫЕ СКОБКИ. ЕСЛИ В ВЫЗОВЕ МАКРОКОМАНДЫ НЕ УКАЗАНЫ НЕОБЯЗАТЕЛЬНЫЕ АРГУМЕНТЫ, МАКРОГЕНЕРАТОР ОБЕСПЕЧИВАЕТ ИХ ЗНАЧЕНИЯ ПО УМОЛЧАНИЮ.

ФОРМАТ МАКРОКОМАНДЫ ЗАГОЛОВКА ВЫГЛЯДИТ СЛЕДУЮЩИМ ОБРАЗОМ:

```
#DS_HEADER PNAME,REV[,UPDATE],[,NUNIT][,ERRTYR],[,STAT]
```

В МАКРОКОМАНДЕ АРГУМЕНТЫ PNAME И REV ЯВЛЯЮТСЯ ОБЯЗАТЕЛЬНЫМИ, ОСТАЛЬНЫЕ - НЕТ.

В МАКРОВЫЗОВЕ АРГУМЕНТЫ ДОЛЖНЫ УКАЗЫВАТЬСЯ ИЛИ ПО ПОЗИЦИЯМ ИЛИ С ПОМОЩЬЮ КЛЮЧЕВЫХ СЛОВ.

ПОКАЗАННЫЙ ПРИМЕР ИЛЛЮСТРИРУЕТ ВЫЗОВ ПО ПОЗИЦИЯМ:

```
#DS_HEADER <SAMPLE TEST>,01,,8,6
```

В ПРИМЕРЕ УКАЗАНЫ АРГУМЕНТЫ PNAME, REV, NUNIT И ERRTYR, А АРГУМЕНТЫ UPDATE И STAT ОПУЩЕНЫ. ОПУЩЕННЫЕ АРГУМЕНТЫ УКАЗЫВАЮТСЯ С ПОМОЩЬЮ ЗАПЯТЫХ, ЕСЛИ ОНИ НЕ ЯВЛЯЮТСЯ ПОСЛЕДНИМИ.

В ПРИМЕРЕ ВМЕСТО ПРОПУЩЕННОГО АРГУМЕНТА UPDATE ЗАПИСАНА ЗАПЯТАЯ, А ВМЕСТО ПРОПУЩЕННОГО АРГУМЕНТА STAT - НЕТ, Т.К. АРГУМЕНТ STAT ЯВЛЯЕТСЯ ПОСЛЕДНИМ В СПИСКЕ АРГУМЕНТОВ. ВМЕСТО ОПУЩЕННЫХ АРГУМЕНТОВ В МАКРОРАСШИРЕНИИ БУДУТ УСТАНОВЛЕНЫ ИХ ЗНАЧЕНИЯ ПО УМОЛЧАНИЮ, КОТОРЫЕ УКАЗАНЫ В ОПРЕДЕЛЕНИИ МАКРОКОМАНДЫ.

ТАКОЙ ЖЕ СПИСОК АРГУМЕНТОВ МОЖЕТ БЫТЬ ЗАДАН С ПОМОЩЬЮ КЛЮЧЕВЫХ СЛОВ СЛЕДУЮЩИМ ОБРАЗОМ:

```
#DS_HEADER PNAME=<SAMPLE TEST>, REV=01, NUNIT=8, ERRTYR=6
```

ЕСЛИ СПИСОК АРГУМЕНТОВ ПРОДОЛЖАЕТСЯ НА СЛЕДУЮЩИХ СТРОКАХ, ТО КАЖДАЯ СТРОКА, ЗА ИСКЛЮЧЕНИЕМ ПОСЛЕДНЕЙ, ДОЛЖНА

13.00078-01 33 01

ЗАКАНЧИВАТЬСЯ ДЕФИСОМ (-).

В МАКРОКОМАНДАХ #DS\_ИМЯ\_X И #DS\_ИМЯ\_L СПИСОК АРГУМЕНТОВ РАСПОЛАГАЕТСЯ СЛЕДУЮЩИМ ОБРАЗОМ: В ПЕРВОМ СЛОВЕ УКАЗЫВАЕТСЯ ЧИСЛО АРГУМЕНТОВ, А В ПОСЛЕДУЮЩИХ ДЛИННЫХ СЛОВАХ РАСПОЛАГАЮТСЯ ПОСЛЕДОВАТЕЛЬНО АРГУМЕНТЫ МАКРОКОМАНДЫ.

НАПРИМЕР, ДЛЯ ВЫЗОВА МАКРОКОМАНДЫ:

#DS\_RELBUF\_X PAGECNT, RETADR, REGION

СПИСОК АРГУМЕНТОВ В ПАМЯТИ БУДЕТ РАСПОЛОЖЕН В СЛЕДУЮЩЕМ ПОРЯДКЕ (КАЖДАЯ ВЕЛИЧИНА ЗАНИМАЕТ ОДНО ДЛИННОЕ СЛОВО):

3  
PAGECNT  
RETA DR  
REGION

## 7.2. ИСПОЛЬЗОВАНИЕ МАКРОКОМАНД #DS\_ИМЯ\_G И

#DS\_ИМЯ\_L

ФОРМАТ ВЫЗОВА МАКРОКОМАНДЫ #DS\_ИМЯ\_G УДОБНО ИСПОЛЬЗОВАТЬ В ТЕХ СЛУЧАЯХ, КОГДА ПОДПРОГРАММА СУПЕРВИЗОРА ДОЛЖНА ВЫЗЫВАТЬСЯ МНОГОКРАТНО С ОДИНАКОВЫМ (ИЛИ ПОЧТИ ОДИНАКОВЫМ) СПИСОКОМ АРГУМЕНТОВ ИЛИ ПРИ ОТСУТСТВИИ ПОСЛЕДНЕГО. ДЛЯ СОЗДАНИЯ СПИСКА АРГУМЕНТОВ МАКРОКОМАНДЫ #DS\_ИМЯ\_G НЕОБХОДИМО ИСПОЛЬЗОВАТЬ МАКРОКОМАНДУ #DS\_ИМЯ\_L С МЕТКОЙ ПЕРЕД НЕЙ. ЭТА МЕТКА ИСПОЛЬЗУЕТСЯ В МАКРОКОМАНДЕ #DS\_ИМЯ\_G В КАЧЕСТВЕ АРГУМЕНТА.

В ОБЩЕМ СЛУЧАЕ МАКРОКОМАНДА #DS\_ИМЯ\_L ДОЛЖНА ЗАПИСЫВАТЬСЯ В ОТДЕЛЕ ДАННЫХ ГОЛОВНОГО МОДУЛЯ.

РАССМОТРИМ ПРИМЕР ИСПОЛЬЗОВАНИЯ УКАЗАННЫХ МАКРОКОМАНД.



1Э.00078-01 33 01

ПРИМЕР.

```
LIST: : #DS_RELBUF_L PAGESNT=10,RETADR=BUFLIM,REGION=0
      .....
BUFLIM: .BLKL 2
      .....
      #DS_RELBUF_G LIST
```

В ПРИМЕРЕ МАКРОКОМАНДА #DS\_RELBUF\_L СОЗДАЕТ СПИСОК АРГУМЕНТОВ С МЕТКОЙ LIST, А МАКРОКОМАНДА #DS\_RELBUF\_G ВЫЗЫВАЕТ ПОДПРОГРАММУ DS#RELBUF СУПЕРВИЗОРА И ПЕРЕДАЕТ ЕЙ АДРЕС МЕТКИ LIST, ПО КОТОРОМУ ПОДПРОГРАММА СЧИТЫВАЕТ НЕОБХОДИМЫЕ ДАННЫЕ.

ИНОГДА ВОЗНИКАЕТ НЕОБХОДИМОСТЬ ИЗМЕНЕНИЯ НЕКОТОРЫХ АРГУМЕНТОВ, ПЕРЕДАВАЕМЫХ ПОДПРОГРАММЕ СУПЕРВИЗОРА. ДЛЯ ЭТОГО НУЖНО ВЫПОЛНИТЬ НЕСКОЛЬКО ШАГОВ. РАССМОТРИМ ИХ НА ПРИМЕРЕ ПОДПРОГРАММЫ DS#RELBUF.

ПРЕДПОЛОЖИМ, ЧТО АРГУМЕНТ PAGESNT НУЖНО ЗАМЕНИТЬ НА 7.

В РАЗДЕЛЕ ДАННЫХ ТЕСТОВОГО МОДУЛЯ ДОЛЖНА БЫТЬ ИСПОЛЬЗОВАНА МАКРОКОМАНДА #DS\_RELBUFDDEF, КОТОРАЯ СОЗДАЕТ СЛЕДУЮЩИЕ СИМВОЛИЧЕСКИЕ ИМЕНА И СМЕЩЕНИЯ:

```
RELBUFD#_NARGS=3
RELBUFD#_PAGESNT=4
RELBUFD#_RETADR=3
RELBUFD#_REGION=12
```

НУЖНО ЗАМЕНИТЬ СТАРУЮ ВЕЛИЧИНУ В СПИСКЕ АРГУМЕНТОВ (PAGESNT=10) НА НОВУЮ (PAGESNT=7):

```
MOVAL LIST,R2 ;БАЗОВЫЙ АДРЕС СПИСКА
                ;АРГУМЕНТОВ
MOVL #7,RELBUFD#_PAGESNT(R2) ;ЗАМЕНА АРГУМЕНТА PAGESNT НА 7
```

ЗАТЕМ НУЖНО ВЫЗВАТЬ ПОДПРОГРАММУ СУПЕРВИЗОРА DS#RELBUF С ПОМОЩЬЮ МАКРОКОМАНДЫ #DS\_RELBUF\_G LIST.

13.00078-01 33 01

### 7.3. ИСПОЛЬЗОВАНИЕ МАКРОКОМАНДЫ ФОРМАТА

#DS\_ИМЯ\_S

ФОРМАТ УДОБНО ИСПОЛЬЗОВАТЬ, ЕСЛИ СЛУЖЕБНАЯ ПОДПРОГРАММА СУПЕРВИЗОРА ДОЛЖНА ВЫЗЫВАТЬСЯ РЕДКО, А ТАКЖЕ В ТЕХ СЛУЧАЯХ, КОГДА ПОДПРОГРАММА СУПЕРВИЗОРА ДОЛЖНА ВЫЗЫВАТЬСЯ МНОГОКРАТНО И КАЖДЫЙ РАЗ С ДРУГИМ СПИСОКОМ АРГУМЕНТОВ.

КАК УЖЕ БЫЛО СКАЗАНО, МАКРОКОМАНДА ФОРМАТА #DS\_ИМЯ\_S ФОРМИРУЕТ ИНСТРУКЦИЮ CALLS.

СПИСОК АРГУМЕНТОВ ЗАДАЕТСЯ В ВЫЗОВЕ МАКРОКОМАНДЫ И ПРИ АССЕМБЛИРОВАНИИ МАКРОКОМАНДА ГЕНЕРИРУЕТ ИНСТРУКЦИИ ЗАСЫЛКИ АРГУМЕНТОВ В СТЕК.

МАКРОРАСШИРЕНИЕ #DS\_RELBUF #ARGCNT=10, #ETADR=BUFLIM, REGION=0

БУДЕТ СОСТОЯТЬ ИЗ СЛЕДУЮЩИХ ИНСТРУКЦИЙ:

```
PUSHL    #0
PUSHAQ   BUFLIM
PUSHL    #10
CALLS    #3,#DS#RELBUF
```

### 7.4. ВЫДАЧА КОДОВ СОСТОЯНИЯ

БОЛЬШИНСТВО МАКРОКОМАНД, ВЫЗЫВАЮЩИХ ПОДПРОГРАММЫ СУПЕРВИЗОРА, ОБЕСПЕЧИВАЮТ ВЫДАЧУ КОДА СОСТОЯНИЯ В РЕГИСТРЕ R0 ПРИ ЗАВЕРШЕНИИ ВЫПОЛНЕНИЯ МАКРОКОМАНДЫ. ПРИ УСПЕШНОМ ВЫПОЛНЕНИИ УСТАНОВЛИВАЕТСЯ МЛАДШИЙ БИТ РЕГИСТРА R0.

ТРИ МЛАДШИХ БИТА СОВМЕСТНО УКАЗЫВАЮТ СТЕПЕНЬ СЕРЬЕЗНОСТИ ОШИБКИ СЛЕДУЮЩИМ ОБРАЗОМ:

```
0 - ПРЕДУПРЕЖДЕНИЕ
1 - УСПЕШНОЕ ОКОНЧАНИЕ
2 - ОШИБКА
4 - СЕРЬЕЗНАЯ ОШИБКА
```

ОСТАЛЬНЫЕ КОМБИНАЦИИ ТРЕХ МЛАДШИХ БИТОВ НЕ ИСПОЛЬЗУЮТСЯ.

13.00078-01 33 01

БИТЫ 3-15 РЕГИСТРА RO КОНКРЕТИЗИРУЮТ УСЛОВИЕ ВОЗВРАТА. КАЖДЫЙ ВОЗВРАЩАЕМЫЙ КОД СОСТОЯНИЯ ИМЕЕТ ОДНОЗНАЧНОЕ СИМВОЛИЧЕСКОЕ ИМЯ В ФОРМАТЕ DSN\_КОД.

ДЛЯ ТОГО, ЧТОБЫ ВЫЯСНИТЬ УСПЕШНО ЛИ ВЫПОЛНИЛАСЬ ТА ИЛИ ИНАЯ МАКРОКОМАНДА ВЫЗОВА ПОДПРОГРАММЫ СУПЕРВИЗОРА, В ДИАГНОСТИЧЕСКОЙ ПРОГРАММЕ АНАЛИЗИРУЕТСЯ МЛАДШИИ БИТ РЕГИСТРА RO, НАПРИМЕР:

```
BLVC    RD,ERRLABEL      ;ПЕРЕХОД К ПОДПРОГРАММЕ ERRLABEL
                          ;ПРИ НАЛИЧИИ ОШИБКИ (МЛ.БИТ RO=0)
```

ДЛЯ ТОГО, ЧТОБЫ ВЫЯСНИТЬ, КАКАЯ ИМЕННО ОШИБКА ПРОИЗОШЛА, ИСПОЛЬЗУЮТСЯ ИНСТРУКЦИИ СРАВНЕНИЯ СОДЕРЖИМОГО РЕГИСТРА RO С ВОЗМОЖНЫМИ КОДАМИ СОСТОЯНИЯ, НАПРИМЕР:

```
CMRW    RD,DSN_INWE      ;ИСХОДНАЯ ОШИБКА ОБОРУДОВАНИЯ?
```

#### 7.5. МАКРОКОМАНДЫ ФОРМАТА ПРОГРАММЫ

МАКРОКОМАНДЫ ФОРМАТА ПРОГРАММЫ, ГЕНЕРИРУЮТ ДИРЕКТИВЫ АССЕМБЛЕРА И ПРОГРАММЫ СВЯЗИ, КОТОРЫЕ ОРГАНИЗУЮТ ИНТЕРФЕЙС МЕЖДУ ДИАГНОСТИЧЕСКОЙ ПРОГРАММОЙ И СУПЕРВИЗОРОМ.

МАКРОКОМАНДЫ ВЫПОЛНЯЮТ СЛЕДУЮЩИЕ ФУНКЦИИ:

- 1) ФОРМИРОВАНИЕ ПОДЗАГОЛОВКОВ;
- 2) ФОРМИРОВАНИЕ ПРОГРАММНЫХ СЕКЦИЙ;
- 3) ФОРМИРОВАНИЕ ТОЧЕК ВХОДА ПОДПРОГРАММ И МАСОК;
- 4) ФОРМИРОВАНИЕ НАЧАЛЬНЫХ И КОНЕЧНЫХ МЕТОК (ПРИЗНАКОВ);
- 5) ГЕНЕРАЦИЯ АДРЕСОВ ПОДПРОГРАММ И ОБЛАСТЕЙ ПАМЯТИ;
- 6) ФОРМИРОВАНИЕ ДИРЕКТИВ УПРАВЛЕНИЯ РАСПЕЧАТКОЙ ЛИСТИНГОВ.

СУПЕРВИЗОР, ПОДПРОГРАММЫ И ДРУГИЕ МАКРОКОМАНДЫ ИСПОЛЬЗУЮТ ВЫРАБАТЫВАЕМЫЕ МАКРОКОМАНДАМИ ДАННЫЕ ДЛЯ УПРАВЛЕНИЯ ВЫПОЛНЕНИЕМ ПРОГРАММЫ И ДЛЯ ДОСТУПА К ОБЛАСТЯМ ПАМЯТИ ДАН-

ных.

### 7.5.1. МАКРОКОМАНДЫ ПРОГРАММНОГО МОДУЛЯ

В ГРУППУ ВХОДЯТ ДВЕ МАКРОКОМАНДЫ: #DS\_BGNMOD И #DS\_ENDMOD. ЭТИ МАКРОКОМАНДЫ ОПРЕДЕЛЯЮТ ГРАНИЦЫ (НАЧАЛО И КОНЕЦ) ПРОГРАММНОГО МОДУЛЯ.

#### 7.5.1.1. #DS\_BGNMOD - НАЧАЛО ПРОГРАММНОГО МОДУЛЯ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_BGNMOD ENV[;TN] L;ST]

ГДЕ ENV - СРЕДА (УСЛОВИЯ) РАБОТЫ ПРОГРАММЫ.

УКАЗЫВАЕТСЯ ОДНО ИЗ СЛЕДУЮЩИХ НАИМЕНОВАНИЙ:

SER\_FUNCTIONAL, SER\_REPAIR, SER\_FUNCTIONAL,  
SER\_REPAIR.

TN - НОМЕР ПЕРВОГО ТЕСТА В МОДУЛЕ (ПО УМОЛЧАНИЮ TN УСТАНАВЛИВАЕТСЯ В ЕДИНИЦУ);

ST - НОМЕР ПЕРВОГО ПОДТЕСТА В МОДУЛЕ (ПО УМОЛЧАНИЮ - 1).

МАКРОКОМАНДА #DS\_BGNMOD ОБЯЗАТЕЛЬНО ИСПОЛЬЗУЕТСЯ В НАЧАЛЕ КАЖДОГО МОДУЛЯ. ОНА ЗАДАЕТ СРЕДУ (УСЛОВИЯ) РАБОТЫ ПРОГРАММЫ. В ТЕСТОВЫХ МОДУЛЯХ МАКРОКОМАНДА ОПРЕДЕЛЯЕТ НОМЕР ПЕРВОГО ТЕСТА И ПОДТЕСТА В МОДУЛЕ.

ВОЗМОЖНЫ СЛЕДУЮЩИЕ СРЕДЫ РАБОТЫ ПРОГРАММЫ:

SER\_FUNCTIONAL - ФУНКЦИОНАЛЬНАЯ ДИАГНОСТИКА ГРУППЫ ЦЕНТРАЛЬНОГО ПРОЦЕССОРА (ТОЛЬКО УРОВЕНЬ 3);

SER\_REPAIR - РЕМОНТНАЯ ДИАГНОСТИКА ГРУППЫ ЦЕНТРАЛЬНОГО ПРОЦЕССОРА (ТОЛЬКО УРОВЕНЬ 3);

SER\_FUNCTIONAL - ФУНКЦИОНАЛЬНАЯ ДИАГНОСТИКА ПОЛЬЗОВАТЕЛЯ ИЛИ СИСТЕМЫ (УРОВЕНЬ 2, 2R ИЛИ 3);

13.00078-01 33 01

SER\_REPAIR - РЕМОНТНАЯ ДИАГНОСТИКА ПОЛЬЗОВАТЕЛЯ ИЛИ СИСТЕМЫ (УРОВЕНЬ 2, 2R ИЛИ 3).

#### 7.5.1.2. ADS\_ENDMOD - КОНЕЦ ПРОГРАММНОГО МОДУЛЯ

ФОРМАТ МАКРОКОМАНДЫ:

ADS\_ENDMOD

В МАКРОКОМАНДЕ АРГУМЕНТОВ НЕТ.

МАКРОКОМАНДА ДОЛЖНА БЫТЬ УКАЗАНА В САМОМ КОНЦЕ КАЖДОГО ПРОГРАММНОГО МОДУЛЯ.

#### 7.5.2. ADS\_HEADER - ФОРМИРОВАНИЕ ЗАГОЛОВКА ПРОГРАММНОЙ СЕКЦИИ

ФОРМАТ МАКРОКОМАНДЫ:

ADS\_HEADER PNAME,REV[,UPDATE][,NUNIT][,ERRTYP][,STAT]

ГДЕ PNAME - ИМЯ ПРОГРАММЫ;

REV - НОМЕР УРОВНЯ ВЫПУСКА ПРОГРАММЫ. ПО УМОЛЧАНИЮ 0;

UPDATE - НОМЕР УРОВНЯ ИЗМЕНЕНИЙ. ПО УМОЛЧАНИЮ 0;

NUNIT - МАКСИМАЛЬНОЕ ЧИСЛО УСТРОЙСТВ, КОТОРЫЕ МОГУТ ТЕСТИРОВАТЬСЯ ОДНОВРЕМЕННО. ПО УМОЛЧАНИЮ 0;

ERRTYP - ЧИСЛО ТИПОВ ОШИБОК, КОТОРЫЕ МОГУТ ВСТРЕТИТЬСЯ НА Тестируемом устройстве;

STAT - УКАЗАТЕЛЬ НАЛИЧИЯ СТАТИСТИЧЕСКОЙ ТАБЛИЦЫ.

ЕСЛИ В ПРОГРАММЕ ИМЕЕТ МЕСТО СТАТИСТИКА, ТО В КАЧЕСТВЕ АРГУМЕНТА STAT ДОЛЖНО БЫТЬ УКАЗАНО СЛОВО STATISTIC, В ПРОТИВНОМ СЛУЧАЕ АРГУМЕНТ STAT ОПУСКАЕТСЯ.

МАКРОКОМАНДА ADS\_HEADER ИСПОЛЬЗУЕТСЯ В НАЧАЛЕ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ ДЛЯ ФОРМИРОВАНИЯ ТАБЛИЦЫ, КОТОРАЯ ОПРЕДЕЛЯЕТ ДЛЯ СУПЕРВИЗОРА СТРУКТУРУ ПРОГРАММЫ. НАРЯДУ СО СВЕДЕНИЯМИ, УКАЗАННЫМИ В АРГУМЕНТАХ МАКРОКОМАНДЫ, ФОРМИРУЕМАЯ

19.00078-01 33 01

ТАБЛИЦА СОДЕРЖИТ ТАКЖЕ СЛЕДУЮЩИЕ ДАННЫЕ:

- 1) УКАЗАТЕЛЬ ТАБЛИЦЫ ДИСПЕТЧЕРА ТЕСТОВ;
- 2) УКАЗАТЕЛЬ СПИСКА ТИПОВ УСТРОЙСТВ;
- 3) УКАЗАТЕЛЬ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ;
- 4) УКАЗАТЕЛЬ ПОДПРОГРАММЫ ОЧИСТКИ;
- 5) УКАЗАТЕЛЬ СПИСКА АДРЕСОВ ИМЕН СЕКЦИИ .И НЕКОТОРЫЕ ДРУГИЕ.

РАСШИРЕНИЕ МАКРОКОМАНДЫ ADS\_HEADER В ДИАГНОСТИЧЕСКОМ ПРОГРАММЕ EVA000 ПОКАЗАНО В ПРИМЕРЕ.

ПРИМЕР.

```
ADS_HEADER <EVA000 LINE PRINTER DIAGNOSTIC>,5,3,10
          .SAVE
          .PSECT #HEADER,PAGE,NOEXE,NOWRT

LEAL_HEADLENGTH: .LONG      A_HEADEND-    ;ДЛИНА БЛОКА
                                     ;ДАННЫХ ЗАГОЛОВКА
LEAL_ENVIRON:     .LONG      #ENV          ;СРЕДА РАБОТЫ
                                     ;ПРОГРАММЫ
LEAL_NAME:        .ADDRESS    T_NAME      ;АДРЕС ТЕКСТА
                                     ;ИМЕНИ ПРОГРАММЫ
LEAL_REV:         .LONG      5           ;УРОВЕНЬ ВЫПУСКА
                                     ;ПРОГРАММЫ
LEAL_UPDATE:     .LONG      3           ;УРОВЕНЬ ИЗМЕНЕНИИ
LEAL_LASTAD:     .ADDRESS    LASTAD      ;ПЕРВАЯ СВОБОДНАЯ
                                     ;ЯЧЕЙКА ПОСЛЕ
                                     ;ПРОГРАММЫ
LEAL_DTP:        .ADDRESS    DISPATCH    ;УКАЗАТЕЛЬ ТАБЛИЦЫ
                                     ;ДИСПЕТЧЕРА ТЕСТОВ
LEAL_DEVTP:      .ADDRESS    AL_DEVTP    ;УКАЗАТЕЛЬ СПИСКА
                                     ;ТИПОВ УСТРОЙСТВ
LEAL_UNIT:       .LONG      10          ;ЧИСЛО УСТРОЙСТВ,
                                     ;КОТОРЫЕ МОГУТ БЫТЬ
                                     ;ПРОВЕРЕНЫ
LEAL_DREG:       .ADDRESS    DEV_REG     ;УКАЗАТЕЛЬ ТАБЛИЦЫ
                                     ;СОДЕРЖИМОГО
                                     ;РЕГИСТРОВ УСТРОЙСТВ
                                     .LONG      28
                                     .LONG      014J
LEAL_ICP:        .ADDRESS    INITIALIZE  ;АДРЕС ПОДПРОГРАММЫ
                                     ;ИНИЦИАЛИЗАЦИИ
LEAL_CCP:        .ADDRESS    CLEAN_UP    ;АДРЕС ПОДПРОГРАММЫ
                                     ;ОЧИСТКИ
LEAL_REPP:       .ADDRESS    SUMMARY    ;АДРЕС ПОДПРОГРАММЫ
                                     ;ИТОГОВОГО ОТЧЕТА
LEAL_STATAB:     .ADDRESS    0          ;УКАЗАТЕЛЬ
                                     ;СТАТИСТИЧЕСКОЙ
                                     ;ТАБЛИЦЫ
LEAL_ERRTP:      .LONG      0           ;ЧИСЛО ТИПОВ ОШИБОК
LEAL_SECNAM:     .ADDRESS    SECTION    ;УКАЗАТЕЛЬ СТЕКА ИМЕН
```

12.00078-01 33 01

		ТЕСТОВЫХ СЕКЦИЙ
LPA_TSTCNT:	.ADDRESS	L_TSTCNT
A_HEADEND:		
T_NAME:	.ASCII	VEVAAA LINE PRINTER DIAGNOSTIC
	.PSECT	_LAST,PAGE
LASTAD:	.PSECT	BTSTCNT,NOEXE,NOVRT,OVR,LONG
L_TSTCNT:	.RESTORE	

### 7.5.3. #DS\_DISPATCH - ИНИЦИАЛИЗАЦИЯ ТАБЛИЦЫ ДИСПЕТЧЕРА

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_DISPATCH

В МАКРОКОМАНДЕ АРГУМЕНТОВ НЕТ.

МАКРОКОМАНДА #DS\_DISPATCH ДОЛЖНА РАСПОЛАГАТЬСЯ ПОСЛЕ МАКРОКОМАНДЫ #DS\_HEADER. ОНА СОХРАНЯЕТ КОНТЕКСТ ТЕКУЩЕЙ ПРОГРАММНОЙ СЕКЦИИ И ГЕНЕРИРУЕТ ПРОГРАММНУЮ СЕКЦИЮ ДЛЯ ТАБЛИЦЫ ДИСПЕТЧЕРА (DISPATCH). КРОМЕ ТОГО, МАКРОКОМАНДА #DS\_HEADER ФОРМИРУЕТ МЕТКУ (DISPATCH) НАЧАЛА ТАБЛИЦЫ ДИСПЕТЧЕРА И ПРИЗНАК КОНЦА ТАБЛИЦЫ ДИСПЕТЧЕРА (С ДЛИННЫХ СЛОВ С НУЛЯМИ). ЗАПИСИ В ТАБЛИЦЕ ДИСПЕТЧЕРА ФОРМИРУЮТСЯ МАКРОКОМАНДАМИ #DS\_BGNTEST.

КАЖДАЯ МАКРОКОМАНДА #DS\_BGNTEST ЗАПИСЫВАЕТ В ТАБЛИЦУ ДИСПЕТЧЕРА СЛЕДУЮЩУЮ ИНФОРМАЦИЮ:

- 1) АДРЕС СЛОВА, В КОТОРОМ НАХОДИТСЯ НОМЕР ТЕСТА;
- 2) АДРЕС СТРОКИ, СОДЕРЖАЩЕЙ НАЗВАНИЕ ТЕСТА;
- 3) АДРЕС МАССИВА ДАННЫХ ДЛЯ ТЕСТА;
- 4) АДРЕС ТОЧКИ ВХОДА В ТЕСТОВУЮ ПОДПРОГРАММУ;
- 5) МАСКА СЕКЦИЙ, В КОТОРЫЕ ВХОДИТ ДАННЫЙ ТЕСТ.

#### 7.5.4. МАКРОКОМАНДЫ СТАТИСТИЧЕСКОЙ ТАБЛИЦЫ

В ГРУППУ ВХОДЯТ МАКРОКОМАНДЫ #DS\_BGNSTAT И #DS\_ENDSTAT, ЗАДАЮЩИЕ ГРАНИЦЫ ОБЛАСТИ ПАМЯТИ, В КОТОРОЙ ЗАПИСЫВАЮТСЯ СВЕДЕНИЯ ОБ АППАРАТНЫХ И ПРОГРАММНЫХ ОШИБКАХ. ЭТИ СВЕДЕНИЯ ИСПОЛЬЗУЮТСЯ ПОДПРОГРАММОЙ ЗАКЛЮЧИТЕЛЬНОГО ОТЧЕТА ПРИ РАСПЕЧАТКЕ СТАТИСТИЧЕСКОГО ОТЧЕТА.

##### 7.5.4.1. #DS\_BGNSTAT - НАЧАЛО СТАТИСТИЧЕСКОЙ ТАБЛИЦЫ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_BGNSTAT

АРГУМЕНТОВ У МАКРОКОМАНДЫ НЕТ.

МАКРОКОМАНДА #DS\_BGNSTAT ИСПОЛЬЗУЕТСЯ В ТЕХ СЛУЧАЯХ, КОГДА ДИАГНОСТИЧЕСКАЯ ПРОГРАММА ДОЛЖНА ИМЕТЬ ВОЗМОЖНОСТЬ ВЫДАЧИ ЗАКЛЮЧИТЕЛЬНОГО (ИТОГОВОГО) ОТЧЕТА.

МАКРОКОМАНДА ГЕНЕРИРУЕТ МЕТКУ НАЧАЛА СТАТИСТИЧЕСКОЙ ТАБЛИЦЫ STATISTIC. СТАТИСТИЧЕСКАЯ ТАБЛИЦА ФОРМИРУЕТСЯ СОСТАВИТЕЛЕМ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ ПУТЕМ ЗАПИСИ ИНФОРМАЦИИ ОБ ОШИБКАХ В ЯЧЕЙКАХ, СМЕЩЕННЫХ ОТНОСИТЕЛЬНО МЕТКИ STATISTIC.

##### 7.5.4.2. #DS\_ENDSTAT - КОНЕЦ СТАТИСТИЧЕСКОЙ ТАБЛИЦЫ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_ENDSTAT

АРГУМЕНТОВ У МАКРОКОМАНДЫ НЕТ.

МАКРОКОМАНДА ОБОЗНАЧАЕТ КОНЕЦ СТАТИСТИЧЕСКОЙ ТАБЛИЦЫ И ЗАПИСЫВАЕТСЯ ПОСЛЕ ЕЕ ОКОНЧАНИЯ.



### 7.5.5. МАКРОКОМАНДЫ ОПРЕДЕЛЕНИЯ ТЕСТОВЫХ СЕКЦИЙ

ДЛЯ ОПРЕДЕЛЕНИЯ ТЕСТОВЫХ СЕКЦИЙ ИСПОЛЬЗУЮТСЯ МАКРОКОМАНДЫ:

#DS\_SECTION И #DS\_SECTDEF

ПЕРВАЯ ПРИМЕНЯЕТСЯ В ГОЛОВНОМ МОДУЛЕ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ, А ВТОРАЯ - В ТЕСТОВЫХ МОДУЛЯХ.

#### 7.5.5.1. #DS\_SECTION - ОПРЕДЕЛЕНИЕ ТЕСТОВЫХ СЕКЦИЙ В ГОЛОВНОМ МОДУЛЕ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_SECTION A [B/C/D.....N/O/P]

ГДЕ A/B/C/... - НАЗВАНИЕ ТЕСТОВЫХ СЕКЦИЙ.

МАКРОКОМАНДА ИСПОЛЬЗУЕТСЯ ДЛЯ ОПРЕДЕЛЕНИЯ НАЗВАНИЙ ТЕСТОВЫХ СЕКЦИЙ. НАРЯДУ С СЕКЦИЯМИ, УКАЗАННЫМИ АРГУМЕНТАМИ A/B/C/... МАКРОКОМАНДА #DS\_SECTION ОПРЕДЕЛЯЕТ ТАКЖЕ ТЕСТОВУЮ СЕКЦИЮ DEFAULT. НАЗВАНИЯ СЕКЦИЙ, ОПРЕДЕЛЕННЫЕ МАКРОКОМАНДОЙ #DS\_SECTION, ИСПОЛЬЗУЮТСЯ МАКРОКОМАНДОЙ #DS\_BGNTTEST ДЛЯ УКАЗАНИЯ ТОГО, КАКОЙ СЕКЦИИ ПРИНАДЛЕЖИТ ТЕСТ.

МАКРОКОМАНДА #DS\_SECTION ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ В ПРОГРАММНОЙ СЕКЦИИ ГОЛОВНОГО МОДУЛЯ.

#### 7.5.5.2. #DS\_SECTDEF - ОПРЕДЕЛЕНИЕ ТЕСТОВЫХ СЕКЦИЙ В ТЕСТОВЫХ МОДУЛЯХ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_SECTDEF A [B/C/D.....N/O/P]

ГДЕ A/B/C/... - НАЗВАНИЕ ТЕСТОВЫХ СЕКЦИЙ.

МАКРОКОМАНДА ИСПОЛЬЗУЕТСЯ ДЛЯ ОПРЕДЕЛЕНИЯ ТЕСТОВЫХ

13.00078-01 33 01

СЕКЦИИ В КАЖДОМ МОДУЛЕ ПРОГРАММЫ, ЗА ИСКЛЮЧЕНИЕМ ГОЛОВНОГО МОДУЛЯ. НАЗВАНИЯ СЕКЦИИ, ЗАДАВАЕМЫЕ В КАЧЕСТВЕ АРГУМЕНТОВ МАКРОКОМАНДЫ #DS\_SECTION, ДОЛЖНЫ ПРИВОДИТЬСЯ В ТОМ ЖЕ ПОРЯДКЕ, КАК И В МАКРОКОМАНДЕ #DS\_SECTION В ГОЛОВНОМ МОДУЛЕ.

#### 7.5.6. МАКРОКОМАНДЫ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ

В ГРУППУ ВХОДЯТ МАКРОКОМАНДЫ #DS\_BGNINIT И #DS\_ENDINIT, КОТОРЫЕ ЗАПИСЫВАЮТСЯ СООТВЕТСТВЕННО В НАЧАЛЕ И КОНЦЕ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ.

##### 7.5.6.1. #DS\_BGNINIT - НАЧАЛО ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ

ФОРМАТ МАКРОКОМАНДЫ:

```
#DS_BGNINIT [REGMASK] [,PSECT]
```

ГДЕ REGMASK - МАСКА СОХРАНЕНИЯ РЕГИСТРОВ В ТОЧКЕ ВХОДА В ПОДПРОГРАММУ ИНИЦИАЛИЗАЦИИ;

PSECT - ЛЮБЫЕ ДОПУСТИМЫЕ АРГУМЕНТЫ ДЛЯ ДИРЕКТИВЫ .PSECT. ПО УМОЛЧАНИЮ PSECT = <INITIALIZE,LONG>.

МАКРОКОМАНДА #DS\_BGNINIT ЗАПИСЫВАЕТСЯ В НАЧАЛЕ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ. ОНА ГЕНЕРИРУЕТ МЕТКУ (INITIALIZE) ВХОДА И МАСКУ СОХРАНЕНИЯ РЕГИСТРОВ В ТОЧКЕ ВХОДА В ПОДПРОГРАММУ ИНИЦИАЛИЗАЦИИ. КРОМЕ ТОГО, ДАННАЯ МАКРОКОМАНДА СОХРАНЯЕТ КОНТЕКСТ ТЕКУЩЕЙ ПРОГРАММНОЙ СЕКЦИИ И ОПРЕДЕЛЯЕТ ПРОГРАММНУЮ СЕКЦИЮ ИНИЦИАЛИЗАЦИИ.

13.00078-01 33 01

7.5.6.2. #DS\_ENDINIT - КОНЕЦ ПОДПРОГРАММЫ  
ИНИЦИАЛИЗАЦИИ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_ENDINIT

АРГУМЕНТОВ У МАКРОКОМАНДЫ НЕТ.

МАКРОКОМАНДА #DS\_ENDINIT ЗАПИСЫВАЕТСЯ В КОНЦЕ ПОДПРОГ-  
РАММЫ ИНИЦИАЛИЗАЦИИ. ОНА ГЕНЕРИРУЕТ МЕТКУ ВЫХОДА  
INITIALIZE\_X, ВЫЗОВ ПОДПРОГРАММЫ СУПЕРВИЗОРА DS#BREAK (ДЛЯ  
ПРОВЕРКИ КОМАНДЫ CONTROL C) И ИНСТРУКЦИЮ ВОЗВРАТА RET. МАК-  
РОКОМАНДА #DS\_ENDINIT ТАКЖЕ ВОССТАНАВЛИВАЕТ ПРОГРАММНУЮ  
СЕКЦИЮ, КОНТЕКСТ КОТОРОЙ БЫЛ СОХРАНЕН МАКРОКОМАНДОЙ  
#DS\_BGNINIT.

7.5.7. МАКРОКОМАНДЫ ПОДПРОГРАММЫ ОЧИСТКИ

В ГРУППУ ВХОДЯТ МАКРОКОМАНДЫ #DS\_BGNCLEAN И  
#DS\_ENDCLEAN, КОТОРЫЕ ЗАПИСЫВАЮТСЯ СООТВЕТСТВЕННО В НАЧАЛЕ  
И КОНЦЕ ПОДПРОГРАММЫ ОЧИСТКИ.

7.5.7.1. #DS\_BGNCLEAN - НАЧАЛО ПОДПРОГРАММЫ  
ОЧИСТКИ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_BGNCLEAN [REGMASK] [,PSECT]

ГДЕ REGMASK - МАСКА СОХРАНЕНИЯ РЕГИСТРОВ В ТОЧКЕ ВХОДА В  
ПОДПРОГРАММУ ОЧИСТКИ;

PSECT - ЛЮБЫЕ ДОПУСТИМЫЕ АРГУМЕНТЫ ДЛЯ ДИРЕКТИВЫ  
.PSECT. ПО УМОЛЧАНИЮ PSECT=<CLEANUP, LONG>.

МАКРОКОМАНДА #DS\_BGNCLEAN ЗАПИСЫВАЕТСЯ В НАЧАЛЕ ПОДП-  
ПРОГРАММЫ ОЧИСТКИ. ОНА ГЕНЕРИРУЕТ МЕТКУ ВХОДА CLEAN\_UP И

19.00078-01 33 01

МАСКУ СОХРАНЕНИЯ РЕГИСТРОВ В ТУЧКЕ ВХОДА В ПОДПРОГРАММУ  
ОЧИСТКИ. КРОМЕ ТОГО, ДАННАЯ МАКРОКОМАНДА СОХРАНЯЕТ КОНТЕКСТ  
ТЕКУЩЕЙ ПРОГРАММНОЙ СЕКЦИИ И ОПРЕДЕЛЯЕТ ПРОГРАММНУЮ СЕКЦИЮ  
ОЧИСТКИ.

7.5.7.2. #DS\_ENDCLEAN - КОНЕЦ ПОДПРОГРАММЫ  
ОЧИСТКИ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_ENDCLEAN

АРГУМЕНТОВ У МАКРОКОМАНДЫ НЕТ.

МАКРОКОМАНДА #DS\_ENDCLEAN ЗАПИСЫВАЕТСЯ В КОНЦЕ ПРОГ-  
РАММЫ ОЧИСТКИ. ОНА ГЕНЕРИРУЕТ МЕТКУ ВЫХОДА CLEAN\_UP\_X,  
ВЫЗОВ ПОДПРОГРАММЫ СУПЕРВИЗОРА DS#BREAK (ДЛЯ ПРОВЕРКИ  
КОМАНДЫ ОПЕРАТОРА CONTROL C) И ИНСТРУКЦИЮ ВОЗВРАТА RET.  
МАКРОКОМАНДА #DS\_ENDCLEAN ТАКЖЕ ВОССТАНАВЛИВАЕТ ПРОГРАММНУЮ  
СЕКЦИЮ, КОНТЕКСТ КОТОРОЙ БЫЛ СОХРАНЕН МАКРОКОМАНДОЙ  
#DS\_BGNCLEAN.

7.5.8. МАКРОКОМАНДЫ ПОДПРОГРАММЫ  
ЗАКЛЮЧИТЕЛЬНОГО ОТЧЕТА

В ГРУППУ ВХОДЯТ МАКРОКОМАНДЫ #DS\_BGNSUMMARY И  
#DS\_ENDSUMMARY, КОТОРЫЕ ДОЛЖНЫ ЗАПИСЫВАТЬСЯ СООТВЕТСТВЕННО  
В НАЧАЛЕ И КОНЦЕ ПОДПРОГРАММЫ ЗАКЛЮЧИТЕЛЬНОГО ОТЧЕТА.

7.5.8.1. #DS\_BGNSUMMARY - НАЧАЛО ПОДПРОГРАММЫ  
ПЕЧАТИ ИТОГОВОГО ОТЧЕТА

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_BGNSUMMARY [REGMASK] [,PSECT]

ГДЕ REGMASK - МАСКА СОХРАНЕНИЯ РЕГИСТРОВ В ТОЧКЕ ВХОДА  
В ПОДПРОГРАММУ ПЕЧАТИ ИТОГОВОГО ОТЧЕТА;

PSECT - ЛЮБЫЕ ДОПУСТИМЫЕ АРГУМЕНТЫ ДЛЯ ДИРЕКТИВЫ  
АССЕМБЛЕРА .PSECT. ПО УМОЛЧАНИЮ PSECT=<SUMARRY, LONG>.

МАКРОКОМАНДА #DS\_BGNSUMMARY ЗАПИСЫВАЕТСЯ В НАЧАЛЕ  
ПОДПРОГРАММЫ ПЕЧАТИ ИТОГОВОГО ОТЧЕТА. ОНА ГЕНЕРИРУЕТ МЕТКУ  
ВХОДА SUMMARY И МАСКУ СОХРАНЕНИЯ РЕГИСТРОВ В ТОЧКЕ ВХОДА В  
ПОДПРОГРАММУ ПЕЧАТИ ИТОГОВОГО ОТЧЕТА. КРОМЕ ТОГО, ДАННАЯ  
МАКРОКОМАНДА СОХРАНЯЕТ КОНТЕКСТ ТЕКУЩЕЙ ПРОГРАММНОЙ СЕКЦИИ  
И ОПРЕДЕЛЯЕТ ПРОГРАММНУЮ СЕКЦИЮ ПЕЧАТИ ИТОГОВОГО ОТЧЕТА.

7.5.8.2. #DS\_ENDSUMMARY - КОНЕЦ ПОДПРОГРАММЫ  
ПЕЧАТИ ИТОГОВОГО ОТЧЕТА

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_ENDSUMMARY

АРГУМЕНТОВ У МАКРОКОМАНДЫ НЕТ.

МАКРОКОМАНДА #DS\_ENDSUMMARY ЗАПИСЫВАЕТСЯ В КОНЦЕ ПОДП-  
РОГРАММЫ ПЕЧАТИ ИТОГОВОГО ОТЧЕТА. ОНА ГЕНЕРИРУЕТ МЕТКУ  
ВЫХОДА SUMMARY\_X, ВЫЗОВ ПОДПРОГРАММЫ СУПЕРВИЗОРА DS#BREAK  
(ДЛЯ ПРОВЕРКИ КОМАНДЫ ОПЕРАТОРА CONTROL C) И ИНСТРУКЦИЮ  
ВОЗВРАТА RET. МАКРОКОМАНДА #DS\_ENDSUMMARY ТАКЖЕ ВОССТАНАВ-  
ЛИВАЕТ ПРОГРАММНУЮ СЕКЦИЮ, КОНТЕКСТ КОТОРОЙ БЫЛ СОХРАНЕН  
МАКРОКОМАНДОЙ #DS\_BGNSUMMARY.

13.00078-01 33 01

7.5.9. МАКРОКОМАНДЫ ПОДПРОГРАММЫ СООБЩЕНИИ ОБ  
ОШИБКАХ

В ГРУППУ ВХОДЯТ МАКРОКОМАНДЫ #DS\_BGNMESSAGE И  
#DS\_ENDMESSAGE, КОТОРЫЕ ЗАДАЮТ СООТВЕТСТВЕННО НИЖНЮЮ И  
ВЕРХНЮЮ ГРАНИЦЫ ПОДПРОГРАММЫ ПЕЧАТИ СООБЩЕНИИ ОБ ОШИБКАХ.

7.5.9.1. #DS\_BGNMESSAGE - НАЧАЛО ПОДПРОГРАММЫ  
ПЕЧАТИ СООБЩЕНИИ ОБ ОШИБКАХ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_BGNMESSAGE [REGMASK]

ГДЕ REGMASK - МАСКА СОХРАНЕНИЯ РЕГИСТРОВ В ТОЧКЕ ВХОДА  
В ПОДПРОГРАММУ ПЕЧАТИ СООБЩЕНИИ ОБ ОШИБКАХ.

МАКРОКОМАНДА #DS\_BGNMESSAGE ЗАПИСЫВАЕТСЯ В НАЧАЛЕ  
ПОДПРОГРАММЫ ПЕЧАТИ СООБЩЕНИИ ОБ ОШИБКАХ. ОНА ГЕНЕРИРУЕТ  
МАСКУ СОХРАНЕНИЯ РЕГИСТРА В ТОЧКЕ ВХОДА В ПОДПРОГРАММУ  
ПЕЧАТИ СООБЩЕНИИ ОБ ОШИБКАХ.

7.5.9.2. #DS\_ENDMESSAGE - КОНЕЦ ПОДПРОГРАММЫ  
ПЕЧАТИ СООБЩЕНИИ ОБ ОШИБКАХ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_ENDMESSAGE

АРГУМЕНТОВ У МАКРОКОМАНДЫ НЕТ.

МАКРОКОМАНДА #DS\_ENDMESSAGE ЗАПИСЫВАЕТСЯ В КОНЦЕ ПОДПРО-  
ГРАММЫ ПЕЧАТИ СООБЩЕНИИ ОБ ОШИБКАХ. ОНА ГЕНЕРИРУЕТ ИНСТРУКЦИЮ  
ВОЗВРАТА RET.

13.00078-01 33 01

### 7.5.10. МАКРОКОМАНДЫ ПОДЗАГОЛОВКА

В ГРУППУ ВХОДЯТ ДВЕ МАКРОКОМАНДЫ: #DS\_SBTTL И #DS\_PAGE. МАКРОКОМАНДА #DS\_SBTTL ФОРМИРУЕТ ПОДЗАГОЛОВОК ТЕСТА ИЛИ ПОДТЕСТА, КОТОРЫЙ РАСПЕЧАТЫВАЕТСЯ НА ВСЕХ СТРАНИЦАХ ЛИСТИНГА, ОТНОСЯЩИХСЯ К ТЕСТУ ИЛИ ПОДТЕСТУ. МАКРОКОМАНДА #DS\_PAGE ЗАПИСЫВАЕТСЯ ПЕРЕД МАКРОКОМАНДОЙ #DS\_SBTTL (НО НЕ ОБЯЗАТЕЛЬНО). ОНА ОБЕСПЕЧИВАЕТ ПЕРЕХОД НА СЛЕДУЮЩУЮ СТРАНИЦУ И ОТМЕНУ РАСПЕЧАТКИ ВЫЗОВА МАКРОКОМАНДЫ #DS\_SBTTL.

#### 7.5.10.1. #DS\_STTTL - ФОРМИРОВАНИЕ ПОДЗАГОЛОВКА ТЕСТА ИЛИ ПОДТЕСТА

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_SBTTL ASCII[,ALIGN]

ГДЕ ASCII - СТРОКА СИМВОЛОВ, СОДЕРЖАЩАЯ НАЗВАНИЕ ТЕСТА ИЛИ ПОДТЕСТА. МАКСИМАЛЬНО ДОПУСТИМОЕ ЧИСЛО СИМВОЛОВ 50. СТРОКА ДОЛЖНА БЫТЬ ЗАКЛЮЧЕНА В УГЛОВЫЕ СКОБКИ < >; ALIGN - УРОВЕНЬ ВЫРАВНИВАНИЯ ПРОГРАММНОЙ СЕКЦИИ ТЕСТА. ВОЗМОЖНЫ СЛЕДУЮЩИЕ УРОВНИ ВЫРАВНИВАНИЯ: BYTE (БАЙТ), WORD (СЛОВО), LONG (ДЛИННОЕ СЛОВО), QUAD (ЧЕТВЕРНОЕ СЛОВО), PAGE (СТРАНИЦА). ПО УМОЛЧАНИЮ - PAGE. АРГУМЕНТ ALIGN НЕ ИСПОЛЬЗУЕТСЯ ПРИ ФОРМИРОВАНИИ ПОДЗАГОЛОВКА ПОДТЕСТА.

МАКРОКОМАНДА #DS\_SBTTL ОБЯЗАТЕЛЬНО ДОЛЖНА БЫТЬ УКАЗАНА В НАЧАЛЕ КАЖДОГО ТЕСТА. ОНА МОЖЕТ БЫТЬ ИСПОЛЬЗОВАНА (НО НЕ ОБЯЗАТЕЛЬНО) В НАЧАЛЕ ПОДТЕСТА.

МАКРОКОМАНДА #DS\_SBTTL ФОРМИРУЕТ ПОДЗАГОЛОВОК, В КОТОРОМ УКАЗЫВАЕТСЯ НОМЕР ТЕСТА (А ДЛЯ ПОДТЕСТОВ И НОМЕР ПОДТЕСТА) И НАЗВАНИЕ ТЕСТА ИЛИ ПОДТЕСТА. ПРИ ТРАНСЛЯЦИИ ПОДЗАГОЛОВОК РАСПЕЧАТЫВАЕТСЯ.

13.00078-01 33 01

7.5.10.2. #DS\_PAGE - ИСКЛЮЧЕНИЕ РАСПЕЧАТКИ ВЫЗОВА  
МАКРОКОМАНДЫ #DS\_SBTTL

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_PAGE [NUM]

ГДЕ NUM - ЧИСЛО 0 ИЛИ 1. 1 УКАЗЫВАЕТ, ЧТО ДОЛЖНА

ГЕНЕРИРОВАТЬСЯ ДИРЕКТИВА МАКРОАССЕМБЛЕРА .PAGE (ПЕРЕХОД НА НОВУЮ СТРАНИЦУ). ЕСЛИ NUM=0 - ДИРЕКТИВА .PAGE НЕ ГЕНЕРИРУЕТСЯ. ПО УМОЛЧАНИЮ NUM=0.

МАКРОКОМАНДА #DS\_PAGE ИСПОЛЬЗУЕТСЯ В СОЧЕТАНИИ С МАКРОКОМАНДОЙ #DS\_SBTTL. ПРИ ЕЕ НАПИСАНИИ ПЕРЕД МАКРОКОМАНДОЙ #DS\_SBTTL ИСКЛЮЧАЕТСЯ РАСПЕЧАТКА В ЛИСТИНГЕ ВЫЗОВА МАКРОКОМАНДЫ #DS\_SBTTL.

ЕСЛИ АРГУМЕНТ NUM=1, ТО СЛЕДУЮЩАЯ СТРОКА ЛИСТИНГА ПЕЧАТАЕТСЯ С НОВОЙ СТРАНИЦЫ.

7.5.11. МАКРОКОМАНДЫ ТЕСТОВОЙ ПОДПРОГРАММЫ

В ГРУППУ ВХОДЯТ МАКРОКОМАНДЫ #DS\_BGNTEST И #DS\_ENDTEST, КОТОРЫЕ ЗАПИСЫВАЮТСЯ СООТВЕТСТВЕННО В НАЧАЛЕ И КОНЦЕ КАЖДОЙ ТЕСТОВОЙ ПОДПРОГРАММЫ.

7.5.11.1. #DS\_BGNTEST - НАЧАЛО ТЕСТОВОЙ  
ПОДПРОГРАММЫ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_BGNTEST [SECTION] L,REGMASK] L,ALIGN]

ГДЕ SECTION - НАЗВАНИЕ ТЕСТОВЫХ СЕКЦИЙ, В КОТОРЫЕ ВХОДИТ ДАННЫЙ ТЕСТ;

REGMASK - МАСКА СОХРАНЕНИЯ РЕГИСТРОВ В ТОЧКЕ ВХОДА В ТЕСТОВУЮ ПОДПРОГРАММУ;



13.00078-01 33 01

ALIGN - УРОВЕНЬ ВЫРАВНИВАНИЯ. ПО УМОЛЧАНИЮ LONG (ДЛИННЫЕ СЛОВА).

МАКРОКОМАНДА #DS\_BGNTEST ЗАПИСЫВАЕТСЯ В НАЧАЛЕ ТЕСТОВОЙ ПОДПРОГРАММЫ. ОНА ФОРМИРУЕТ МЕТКУ ВХОДА В ТЕСТОВУЮ ПОДПРОГРАММУ TEST\_N (ГДЕ N-ТРЕХЗНАЧНЫЙ НОМЕР ТЕСТА С ВЕДУЩИМИ НУЛЯМИ), А ТАКЖЕ МАСКУ СОХРАНЕНИЯ РЕГИСТРОВ В ТОЧКЕ ВХОДА. В ТЕХ СЛУЧАЯХ, КОГДА НЕ ЗАДАЕТСЯ СПИСОК АРГУМЕНТОВ ДЛЯ ДАННОГО ТЕСТА (НА ОСНОВЕ МАКРОКОМАНД #DS\_BGNDDATA И #DS\_ENDDDATA), МАКРОКОМАНДА #DS\_BGNTEST ФОРМИРУЕТ ПЕРЕД МЕТКОЙ TEST\_N МЕТКУ DATA\_N (ГДЕ N-ТРЕХЗНАЧНЫЙ НОМЕР ТЕСТА С ВЕДУЩИМИ НУЛЯМИ) И ДЛИННОЕ СЛОВО С НУЛЯМИ.

КРОМЕ ТОГО, МАКРОКОМАНДА #DS\_BGNTEST ГЕНЕРИРУЕТ В СЕКЦИИ ДИСПЕТЧЕРА (DISPATCH) СЛЕДУЮЩУЮ ИНФОРМАЦИЮ:

- 1) АДРЕС СЛОВА, В КОТОРОМ СОДЕРЖИТСЯ НОМЕР ТЕСТА;
- 2) АДРЕС СТРОКИ С НАЗВАНИЕМ ДАННОГО ТЕСТА;
- 3) АДРЕС МАССИВА ДАННЫХ ДЛЯ ТЕСТА;
- 4) АДРЕС ТОЧКИ ВХОДА В ТЕСТОВУЮ ПОДПРОГРАММУ;
- 5) МАСКУ СЕКЦИЙ, В КОТОРЫЕ ВХОДИТ ДАННЫЙ ТЕСТ;

7.5.11.2. #DS\_ENDTEST - КОНЕЦ ТЕСТОВОЙ ПОДПРОГРАММЫ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_ENDTEST

АРГУМЕНТОВ У МАКРОКОМАНДЫ НЕТ.

МАКРОКОМАНДА #DS\_ENDTEST ЗАПИСЫВАЕТСЯ В КОНЦЕ ТЕСТОВОЙ ПОДПРОГРАММЫ. ОНА ФОРМИРУЕТ ИНСТРУКЦИЮ ЗАСЫЛКИ 1 В РЕГИСТРЕ ДЛЯ УКАЗАНИЯ НОРМАЛЬНОГО ЗАВЕРШЕНИЯ ТЕСТА, МЕТКУ ВЫХОДА TEST\_N\_X (ГДЕ N-ТРЕХЗНАЧНЫЙ НОМЕР ТЕСТА С ВЕДУЩИМИ НУЛЯМИ), ВЫЗОВ ПОДПРОГРАММЫ СУПЕРВИЗОРА DSABREAK (ДЛЯ ПРОВЕРКИ

КОМАНДЫ ОПЕРАТОРА CONTROL C) И ИНСТРУКЦИЮ ВОЗВРАТА RET.

#### 7.5.12. МАКРОКОМАНДЫ ДАННЫХ ДЛЯ ТЕСТА

В ГРУППУ ВХОДЯТ МАКРОКОМАНДЫ #DS\_BGNDATA И #DS\_ENDDATA, ОПРЕДЕЛЯЮЩИЕ ГРАНИЦЫ ДЛЯ ЗАПИСИ ВХОДНЫХ ДАННЫХ ТЕСТА. ЭТИ МАКРОКОМАНДЫ ИСПОЛЬЗУЕТСЯ В ТЕХ СЛУЧАЯХ, КОГДА ОДИН И ТОТ ЖЕ ТЕСТ ИСПОЛЬЗУЕТСЯ НЕСКОЛЬКО РАЗ С РАЗЛИЧНЫМИ ДАННЫМИ.

##### 7.5.12.1. #DS\_BGNDATA - НАЧАЛО МАССИВА ДАННЫХ ДЛЯ ТЕСТА

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_BGNDATA [ALIGN]

ГДЕ ALIGN - УРОВЕНЬ ВЫРАВНИВАНИЯ МАССИВА ДАННЫХ.

ВОЗМОЖНЫ УРОВНИ: BYTE (БАЙТ), WORD (СЛОВО), LONG (ДЛИННОЕ СЛОВО), QUAD (ЧЕТВЕРТНОЕ СЛОВО), PAGE (СТРАНИЦА).

МАКРОКОМАНДА #DS\_BGNDATA ИСПОЛЬЗУЕТСЯ В ТЕХ СЛУЧАЯХ, КОГДА ОДИН И ТОТ ЖЕ ТЕСТ НЕОБХОДИМО ВЫПОЛНИТЬ НЕСКОЛЬКО РАЗ С РАЗЛИЧНЫМИ ДАННЫМИ. ЭТА МАКРОКОМАНДА ГЕНЕРИРУЕТ МЕТКУ НАЧАЛА МАССИВА ДАННЫХ DATA\_N, ГДЕ N-ТРЕХЗНАЧНЫЙ НОМЕР ТЕСТА (С ВЕДУЩИМИ НУЛЯМИ).

МАКРОКОМАНДА #DS\_BGNDATA ЗАПИСЫВАЕТСЯ ПОСЛЕ МАКРОКОМАНДЫ #DS\_SBTIL СООТВЕТСТВУЮЩЕГО ТЕСТА.

НЕПОСРЕДСТВЕННО ПОСЛЕ ВЫЗОВА МАКРОКОМАНДЫ #DS\_BGNDATA ДОЛЖЕН ЗАПИСЫВАТЬСЯ МАССИВ ДАННЫХ ДЛЯ ТЕСТА. МАССИВ СОСТОИТ ИЗ НЕСКОЛЬКИХ СПИСКОВ АРГУМЕНТОВ, ОПРЕДЕЛЯЕМЫХ СОСТАВИТЕЛЕМ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ. КАЖДОМУ СПИСКУ АРГУМЕНТОВ СООТВЕТСТВУЕТ ОДНО ВЫПОЛНЕНИЕ ТЕСТА.

13.00078-01 33 01

СПИСОК АРГУМЕНТОВ ДОЛЖЕН СОСТОЯТЬ ИЗ ДЛИННЫХ СЛОВ, ПРИЧЕМ В ПЕРВОМ ДЛИННОМ СЛОВЕ СПИСКА НЕОБХОДИМО УКАЗЫВАТЬ ЧИСЛО АРГУМЕНТОВ В СПИСКЕ. КОД ТЕСТА ДОЛЖЕН ОПРАШИВАТЬ СПИСКИ АРГУМЕНТОВ СО СМЕЩЕНИЕМ ОТ УКАЗАТЕЛЯ АРГУМЕНТОВ АР. ПРИ КАЖДОМ ПОВТОРЕНИИ ТЕСТА СУПЕРВИЗОР ПОДВИГАЕТ АР К НАЧАЛУ ОЧЕРЕДНОГО СПИСКА АРГУМЕНТОВ.

#### 7.5.12.2. #DS\_ENDDATA - КОНЕЦ МАССИВА ДАННЫХ ДЛЯ ТЕСТА

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_ENDDATA

АРГУМЕНТОВ У МАКРОКОМАНДЫ НЕТ.

ВЫЗОВ МАКРОКОМАНДЫ #DS\_ENDDATA ДОЛЖЕН ЗАПИСЫВАТЬСЯ НЕПОСРЕДСТВЕННО ПОСЛЕ ОКОНЧАНИЯ МАССИВА ДАННЫХ ДЛЯ ТЕСТА.

МАКРОКОМАНДА ГЕНЕРИРУЕТ ДЛИННОЕ СЛОВО, СОДЕРЖАЩЕЕ НУЛИ. ЭТО СЛОВО ЯВЛЯЕТСЯ ДЛЯ СУПЕРВИЗОРА ПРИЗНАКОМ КОНЦА МАССИВА ДАННЫХ ДЛЯ ТЕСТА.

КОГДА ПОСЛЕ ПОСЛЕДНЕГО СПИСКА АРГУМЕНТОВ СУПЕРВИЗОР ОПРАШИВАЕТ ГЕНЕРИРУЕМОЕ МАКРОКОМАНДОЙ #DS\_ENDDATA ДЛИННОЕ СЛОВО С НУЛЯМИ, ДИСПЕТЧЕР СУПЕРВИЗОРА ПЕРЕДАЕТ УПРАВЛЕНИЕ НОВОМУ ТЕСТУ.

РАССМОТРИМ ПРИМЕР ЗАПИСИ ДАННЫХ ДЛЯ ТЕСТА.

ПРИМЕР.

```
ADS_SBTTL          <НАЗВАНИЕ_ТЕСТА>
.....
ADS_BGNDATA
.LONG      4,  A/B/C/D  :ПЕРВЫЙ СПИСОК АРГУМЕНТОВ
.LONG      4,  K/B/L/D  :ВТОРОЙ СПИСОК АРГУМЕНТОВ
.LONG      4,  N/M/L/D  :ТРЕТИЙ СПИСОК АРГУМЕНТОВ
ADS_ENDDATA
ADS_BGNTTEST
ТЕСТОВАЯ ПОДПРОГРАММА
ADS_ENDTEST
```

В ПРИМЕРЕ СУПЕРВИЗОР ВЫЗЫВАЕТ ТЕСТ ТРИ РАЗА, ПО ОДНОМУ РАЗУ С КАЖДЫМ СПИСОКОМ АРГУМЕНТОВ. ЗАТЕМ, КОГДА СУПЕРВИЗОР ВСТРЕЧАЕТ ДЛИННОЕ СЛОВО С НУЛЯМИ, ФОРМИРУЕМОЕ МАКРОКОМАНДОЙ ADS\_ENDDATA, УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ ДИСПЕТЧЕРУ СУПЕРВИЗОРА.

#### 7.6. МАКРОКОМАНДЫ УПРАВЛЕНИЯ ПРОГРАММОЙ

МАКРОКОМАНДЫ УПРАВЛЕНИЯ ПРОГРАММОЙ ГЕНЕРИРУЮТ ИНСТРУКЦИИ, КОТОРЫЕ ПРИ ОПРЕДЕЛЕННЫХ УСЛОВИЯХ МОГУТ ИЗМЕНИТЬ ХОД ВЫПОЛНЕНИЯ ПРОГРАММЫ.

НЕКОТОРЫЕ ИЗ ЭТИХ МАКРОКОМАНД ГЕНЕРИРУЮТ ИНСТРУКЦИИ БЕЗ ВЫЗОВА КАКОЙ-ЛИБО ПОДПРОГРАММЫ, ДРУГИЕ - ФОРМИРУЮТ ВЫЗОВЫ ПОДПРОГРАММ СУПЕРВИЗОРА.

ПОДПРОГРАММЫ СУПЕРВИЗОРА, ВЫЗЫВАЕМЫЕ РАССМАТРИВАЕМЫМИ МАКРОКОМАНДАМИ, НЕ ВОЗВРАЩАЮТ КОДЫ СОСТОЯНИЯ.

##### 7.6.1. МАКРОКОМАНДЫ ПЕРЕДАЧИ УПРАВЛЕНИЯ В ПОДПРОГРАММЕ ИНИЦИАЛИЗАЦИИ

В ГРУППУ ВХОДЯТ МАКРОКОМАНДЫ ADS\_BPASSU, ADS\_BNPASSU И ADS\_ENDPASS\_X. ОНИ ПОЗВОЛЯЮТ ПРОИЗВОДИТЬ ВЕТВЛЕНИЕ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ В ЗАВИСИМОСТИ ОТ ТОГО, ВЫПОЛНЯЕТСЯ ОНА ПЕРВЫЙ РАЗ ИЛИ НЕТ, И УКАЗЫВАТЬ СУПЕРВИЗОРУ ОКОНЧАНИЕ ПРОХОДА.

13.00078-01 33 01

ОБЫЧНО ПОДПРОГРАММА ИНИЦИАЛИЗАЦИИ ИСПОЛЬЗУЕТ ОДНУ ИЗ МАКРОКОМАНД (#DS\_VPASSO ИЛИ #DS\_VNPASSO), КОТОРАЯ РАБОТАЕТ СОВМЕСТНО СО СЛУЖЕБНОЙ МАКРОКОМАНДОЙ #DS\_ENDPASS\_X, ВЫЗЫВАЮЩЕЙ ПОДПРОГРАММУ СУПЕРВИЗОРА (#DSX#ENDPASS) ДЛЯ ОТМЕТКИ КОНЦА ПРОХОДА ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ.

ПОСЛЕ ТОГО КАК БУДЕТ ВЫПОЛНЕНО ЗАДАННОЕ ОПЕРАТОРОМ ЧИСЛО ПРОХОДОВ, УКАЗАННАЯ ПОДПРОГРАММА ВЫЗЫВАЕТ ВЫПОЛНЕНИЕ ПОДПРОГРАММ ОЧИСТКИ И ИТОГОВОГО ОТЧЕТА.

РАССМОТРИМ ПРИМЕР, В КОТОРОМ СОДЕРЖИТСЯ ЧАСТЬ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ.

ПРИМЕР.

```

1      .SBTTL  INIZIALIZATION CODE      ;ПОДЗАГОЛОВОК
                                           ;ПОДПРОГРАММЫ
                                           ;ИНИЦИАЛИЗАЦИИ
2      #DS_BGNINIT                       ;НАЧАЛО ПОДПРОГРАММЫ
                                           ;ИНИЦИАЛИЗАЦИИ
3      #DS_VNPASSO  10#                  ;ПЕРВЫЙ ПРОХОД?
4      CLR  LOG_UNIT                     ;ДА. НАЧАЛО ПРОВЕРКИ
                                           ;ПЕРВОГО УСТРОЙСТВА
5      BRB  20#                          ;
6  10#: INCL  LOG_UNIT                    ;НЕТ. НАЧАЛО РАБОТЫ
                                           ;СО СЛЕДУЮЩИМ
                                           ;УСТРОЙСТВОМ
7      CMPL  #SA#GL_UNITS,LOG_UNIT       ;ПОСЛЕДНЕЕ УСТРОЙСТВО
                                           ;ПРОВЕРЕНО?
8      BNEQ  20#                          ;НЕТ
9      #DS_ENDPASS_G                      ;ДА. ВЫПОЛНЕН ЛИ
                                           ;ПОСЛЕДНИЙ ПРОХОД?
10     CLR  LOG_UNIT                      ;ПОСЛЕДНИЙ ПРОХОД
                                           ;НЕ ВЫПОЛНЕН
11  20#: #DS_GPHARD...                   ;ПОЛУЧЕНИЕ АДРЕСА
                                           ;Р-ТАБЛИЦЫ
12     #ASSIGN...                          ;НАЗНАЧЕНИЕ КАНАЛА
      .
      .
      .
17     CLRQ  CSRW...                       ;ИНИЦИАЛИЗАЦИЯ
                                           ;ПРОВЕРЯЕМОГО
                                           ;УСТРОЙСТВА

```

В ПРИМЕРЕ ПРИ ПЕРВОМ ВЫПОЛНЕНИИ ПЕРВОГО ПРОХОДА ПРОГРАММЫ НОМЕР УСТРОЙСТВА УСТАНАВЛИВАЕТСЯ В 0 (СТРОКА4), И УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ СТРОКЕ11. ПРИ ВТОРОМ ВЫПОЛНЕНИИ ПЕРВОГО ПРОХОДА ФЛАГ ПРОХОДА 0 (#SA#V\_PASSO) ОЧИЩАЕТСЯ И

13.00078-01 33 07

УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ ОТ СТРОКИ 3 К СТРОКЕ 6, В КОТОРОЙ ЛОГИЧЕСКИЙ НОМЕР УСТРОЙСТВА УВЕЛИЧИВАЕТСЯ НА 1. КОГДА ПОСЛЕДНЕЕ ВЫБРАННОЕ УСТРОЙСТВО БУДЕТ ПРОВЕРЕНО, Т.Е. НОМЕР ЛОГИЧЕСКОГО УСТРОЙСТВА (LOG\_UNIT) БУДЕТ РАВЕН ЧИСЛУ ПРОВЕРЯЕМЫХ УСТРОЙСТВ (DSAPGL\_UNITS), ВЫПОЛНЯЕТСЯ МАКРОКОМАНДА #DS\_ENDPASS\_6, КОТОРАЯ СООБЩАЕТ СУПЕРВИЗОРУ О ЗАВЕРШЕНИИ ОЧЕРЕДНОГО ПРОХОДА. ПРИ ЗАВЕРШЕНИИ ПОСЛЕДНЕГО ПРОХОДА ПОДПРОГРАММА СУПЕРВИЗОРА, ВЫЗЫВАЕМАЯ МАКРОКОМАНДОЙ #DS\_ENDPASS\_6, ПЕРЕДАЕТ УПРАВЛЕНИЕ ПОДПРОГРАММАМ ОЧИСТКИ И ИТОГОВОГО ОТЧЕТА.

ЕСЛИ ЖЕ ПРОХОДЫ НЕ ЗАКОНЧИЛИСЬ, ТО УПРАВЛЕНИЕ ВОЗВРАЩАЕТСЯ К СТРОКЕ 10 ДЛЯ ВЫБОРА ПЕРВОГО УСТРОЙСТВА (LOG\_UNIT=0) И ВЫПОЛНЕНИЯ ЕЩЕ ОДНОГО ПРОХОДА.

КОГДА ЛОГИЧЕСКИЙ НОМЕР УСТРОЙСТВА ПОЛУЧАЕТ ПРИРАЩЕНИЕ (СТРОКА 6), А ПОСЛЕДНЕЕ УСТРОЙСТВО ЕЩЕ НЕ ПРОВЕРЕНО, ТО ОСУЩЕСТВЛЯЕТСЯ ПЕРЕХОД К СТРОКЕ 11 И НАЧИНАЕТСЯ ИНИЦИАЛИЗАЦИЯ ПРОВЕРЯЕМОГО УСТРОЙСТВА.

#### 7.6.1.1. #DS\_BRASSU - ПЕРЕДАЧА УПРАВЛЕНИЯ ПРИ ПЕРВОМ ВЫПОЛНЕНИИ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_BRASSU ADR

ГДЕ ADR - АДРЕС (МЕТКА) ПЕРЕДАЧИ УПРАВЛЕНИЯ ПРИ ПЕРВОМ ВЫПОЛНЕНИИ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ.

МАКРОКОМАНДА ИСПОЛЬЗУЕТСЯ В ПОДПРОГРАММЕ ИНИЦИАЛИЗАЦИИ ДЛЯ ПРОПУСКА ОПРЕДЕЛЕННЫХ ЧАСТЕЙ ПРОГРАММ ПРИ ПЕРВОМ ЕЕ ВЫПОЛНЕНИИ.

13.00078-01 33 01

7.6.1.2. `ADS_VNPASSO` - ПЕРЕДАЧА УПРАВЛЕНИЯ ПРИ  
НЕ ПЕРВОМ ВЫПОЛНЕНИИ ПОДПРОГРАММЫ  
ИНИЦИАЛИЗАЦИИ

ФОРМАТ МАКРОКОМАНДЫ:

`ADS_VNPASSO ADR`

ГДЕ `ADR` - АДРЕС (МЕТКА) ПЕРЕДАЧИ УПРАВЛЕНИЯ ПРИ НЕ ПЕРВОМ  
ВЫПОЛНЕНИИ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ.

МАКРОКОМАНДА ИСПОЛЬЗУЕТСЯ В ПОДПРОГРАММЕ ИНИЦИАЛИЗАЦИИ  
ДЛЯ ПРОПУСКА ОПРЕДЕЛЕННЫХ ЧАСТЕЙ ПРОГРАММЫ ПРИ НЕ ПЕРВОМ ЕЕ  
ВЫПОЛНЕНИИ.

7.6.1.3. `ADS_ENDPASS_X` - СООБЩЕНИЕ СУПЕРВИЗОРУ  
ОБ ОКОНЧАНИИ ПРОХОДА ДИАГНОСТИЧЕСКОЙ  
ПРОГРАММЫ

ФОРМАТ МАКРОКОМАНДЫ:

`ADS_ENDPASS_X`

АРГУМЕНТОВ В МАКРОКОМАНДЕ НЕТ.

МАКРОКОМАНДА ИСПОЛЬЗУЕТСЯ В ПОДПРОГРАММЕ ИНИЦИАЛИЗА-  
ЦИИ. ОНА ВЫЗЫВАЕТ ПОДПРОГРАММУ СУПЕРВИЗОРА ДЛЯ ОТМЕТКИ КОН-  
ЦА ПРОХОДА ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ. ЕСЛИ ВЫПОЛНЕНО ЗАДАН-  
НОЕ ОПЕРАТОРОМ ЧИСЛО ПРОХОДОВ, ПОДПРОГРАММА СУПЕРВИЗОРА  
ОБЕСПЕЧИВАЕТ ВЫПОЛНЕНИЕ ПОДПРОГРАММ ИТОГОВОГО ОТЧЕТА И  
ОЧИСТКИ.

### 7.6.2. МАКРОКОМАНДЫ ФЛАГА БЫСТРОЙ ПРОВЕРКИ

В ГРУППУ ВХОДЯТ МАКРОКОМАНДЫ #DS\_BQUICK И #DS\_BNQUICK, КОТОРЫЕ ПРОВЕРЯЮТ СОСТОЯНИЕ УПРАВЛЯЮЩЕГО ФЛАГА БЫСТРОЙ ПРОВЕРКИ QUICK. УКАЗАННЫЕ МАКРОКОМАНДЫ ДАЮТ ВОЗМОЖНОСТЬ ИСКЛЮЧИТЬ НЕКОТОРЫЕ ЧАСТИ ДЛИННЫХ ТЕСТОВ В ТЕХ СЛУЧАЯХ, ЕСЛИ ОПЕРАТОРУ ТРЕБУЕТСЯ УСКОРИТЬ ВЫПОЛНЕНИЕ ПРОГРАММЫ.

В ПРИМЕРЕ ПОКАЗАНО ИСПОЛЬЗОВАНИЕ МАКРОКОМАНДЫ #DS\_BNQUICK В СОЧЕТАНИИ С МАКРОКОМАНДАМИ #DS\_EXIT И #DS\_ENDTEST.

ПРИМЕР.

```
#DS_SBTTL TEST 3      ;ПОДЗАГЛОВОК ТЕСТА
#DS_BGNTEST ...      ;НАЧАЛО ТЕСТА
#DS_BNQUICK 104      ;ПЕРЕХОД ПРИ ОТСУТСТВИИ ФЛАГА
                    ;БЫСТРОЙ ПРОВЕРКИ
#DS_EXIT TEST        ;ПЕРЕХОД К МАКРОКОМАНДЕ
                    ;#DS_ENDTEST ДЛЯ ИСКЛЮЧЕНИЯ ТЕСТА 3
104:                CLR R6
;ТЕСТОВАЯ ПОДПРОГРАММА
#DS_ENDTEST
```

#### 7.6.2.1. #DS\_BQUICK - ПЕРЕХОД В РЕЖИМЕ БЫСТРОЙ ПРОВЕРКИ

ФОРМАТ МАКРОКОМАНДЫ:

```
#DS_BQUICK ADR
```

ГДЕ ADR - АДРЕС (МЕТКА) ПЕРЕДАЧИ УПРАВЛЕНИЯ В РЕЖИМЕ БЫСТРОЙ ПРОВЕРКИ.

МАКРОКОМАНДА ПРОВЕРЯЕТ СОСТОЯНИЕ УПРАВЛЯЮЩЕГО ФЛАГА БЫСТРОЙ ПРОВЕРКИ QUICK. ЕСЛИ ЭТОТ ФЛАГ УСТАНОВЛЕН, Т.Е. ТЕСТ ДОЛЖЕН ВЫПОЛНЯТЬСЯ В РЕЖИМЕ БЫСТРОЙ ПРОВЕРКИ, УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ ПО АДРЕСУ, УКАЗАННОМУ АРГУМЕНТОМ ADR. РЕЖИМ БЫСТРОЙ ПРОВЕРКИ УСТАНОВЛИВАЕТСЯ ОПЕРАТОРОМ.

МАКРОКОМАНДА #DS\_BQUICK ПОЗВОЛЯЕТ ПРОПУСКАТЬ ОПРЕДЕЛЕННЫЕ ЧАСТИ ТЕСТОВОЙ ПОДПРОГРАММЫ ПРИ НЕОБХОДИМОСТИ УСКО-



19.00078-01 33 01

РЕНИЯ ЕЕ ВЫПОЛНЕНИЯ.

### 7.6.2.2. ADS\_BNQUICK - ПЕРЕХОД ПРИ ОТСУТСТВИИ РЕЖИМА БЫСТРОЙ ПРОВЕРКИ

ФОРМАТ МАКРОКОМАНДЫ:

ADS\_BNQUICK ADR

ГДЕ ADR - АДРЕС (МЕТКА) ПЕРЕДАЧИ УПРАВЛЕНИЯ ПРИ  
ОТСУТСТВИИ РЕЖИМА БЫСТРОЙ ПРОВЕРКИ.

МАКРОКОМАНДА ADS\_BNQUICK ПРОВЕРЯЕТ СОСТОЯНИЕ УПРАВЛЯЮ-  
ЩЕГО ФЛАГА БЫСТРОЙ ПРОВЕРКИ QUICK. ЕСЛИ ЭТОТ ФЛАГ ОЧИЩЕН,  
Т.Е. ТЕСТ ДОЛЖЕН ВЫПОЛНЯТЬСЯ НЕ В РЕЖИМЕ БЫСТРОЙ ПРОВЕРКИ,  
ПРОИСХОДИТ ПЕРЕХОД ПО АДРЕСУ, УКАЗАННОМУ АРГУМЕНТОМ ADR.

МАКРОКОМАНДА ПОЗВОЛЯЕТ ПЕРЕСТРАИВАТЬ ТЕСТОВУЮ ПРОГРАМ-  
МУ ТАКИМ ОБРАЗОМ, ЧТОБЫ ПРИ НЕОБХОДИМОСТИ СОКРАТИТЬ ВРЕМЯ  
ЕЕ ВЫПОЛНЕНИЯ.

### 7.6.3. МАКРОКОМАНДЫ ФЛАГА ОПЕРАТОРА

В ЭТУ ГРУППУ ВХОДЯТ МАКРОКОМАНДЫ ADS\_VOPER И  
ADS\_VNOPER, КОТОРЫЕ ПОЗВОЛЯЮТ ВЫПОЛНЯТЬ ИЛИ ИСКЛЮЧАТЬ ОПРЕ-  
ДЕЛЕННЫЕ ТЕСТЫ В ЗАВИСИМОСТИ ОТ СОСТОЯНИЯ ФЛАГА УПРАВЛЕНИЯ  
ОПЕРАТОРА OPERATOR. В ЧАСТНОСТИ, УКАЗАННЫЕ МАКРОКОМАНДЫ  
МОГУТ ИСПОЛЬЗОВАТЬСЯ ПРИ ОТСУТСТВИИ ОПЕРАТОРА ДЛЯ ИСКЛЮЧЕ-  
НИЯ ТЕХ ЧАСТЕЙ ТЕСТОВ, КОТОРЫЕ ТРЕБУЮТ ВМЕШАТЕЛЬСТВА ОПЕРА-  
ТОРА.

ПРИМЕР.

```
.SBTTL      TEST4
ADS_BGNTEST
ADS_VOPER 104  ПЕРЕХОД ПРИ НАЛИЧИИ ОПЕРАТОРА
ADS_EXIT  TEST  ПЕРЕХОД НА МАКРОКОМАНДУ
.           ADS_ENDTEST ДЛЯ ВЫХОДА ИЗ
.           ТЕСТА 4 ПРИ ОТСУТСТВИИ ОПЕРАТОРА
104 ADS_BGNSUB
```

13.00078-01 33 01

.  
.  
.  
ADS\_ENDSUB  
.  
.  
.  
ADS\_ENDTEST

7.6.3.1. ADS\_BOPER - ПЕРЕХОД ПРИ НАЛИЧИИ  
ОПЕРАТОРА

ФОРМАТ МАКРОКОМАНДЫ:

ADS\_BOPER ADR

ГДЕ ADR - АДРЕС (МЕТКА) ПЕРЕДАЧИ УПРАВЛЕНИЯ ПРИ  
УСТАНОВКЕ ФЛАГА УПРАВЛЕНИЯ ОПЕРАТОРА.

МАКРОКОМАНДА ADS\_BOPER ПРОВЕРЯЕТ СОСТОЯНИЕ ФЛАГА  
УПРАВЛЕНИЯ ОПЕРАТОРА. ЕСЛИ ЭТОТ ФЛАГ УСТАНОВЛЕН (Т.Е.  
ВЫПОЛНЕНИЕ ПРОГРАММЫ ПРОИЗВОДИТСЯ ПОД УПРАВЛЕНИЕМ ОПЕ-  
РАТОРА), УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ ПО АДРЕСУ, УКАЗАННОМУ  
АРГУМЕНТОМ ADR.

МАКРОКОМАНДА ПОЗВОЛЯЕТ ВЫПОЛНЯТЬ ИЛИ ПРОПУСКАТЬ НЕКО-  
ТОРЫЕ ЧАСТИ ПРОГРАММЫ В ЗАВИСИМОСТИ ОТ СОСТОЯНИЯ ФЛАГА  
УПРАВЛЕНИЯ ОПЕРАТОРА.

7.6.3.2. ADS\_BNOPER - ПЕРЕХОД ПРИ ОТСУТСТВИИ  
ОПЕРАТОРА

ФОРМАТ МАКРОКОМАНДЫ:

ADS\_BNOPER ADR

ГДЕ ADR - АДРЕС (МЕТКА) ПЕРЕДАЧИ УПРАВЛЕНИЯ ПРИ  
ОЧИЩЕННОМ СОСТОЯНИИ ФЛАГА УПРАВЛЕНИЯ ОПЕРАТОРА.

МАКРОКОМАНДА ADS\_BNOPER ПРОВЕРЯЕТ СОСТОЯНИЕ ФЛАГА  
УПРАВЛЕНИЯ ОПЕРАТОРА. ЕСЛИ ЭТОТ ФЛАГ НЕ УСТАНОВЛЕН (Т.Е.  
ВЫПОЛНЕНИЕ ПРОГРАММЫ ПРОИЗВОДИТСЯ БЕЗ ВМЕШАТЕЛЬСТВА ОПЕРА-

13.00078-01 33 01

ГОРА), УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ ПО АДРЕСУ, УКАЗАННОМУ АРГУМЕНТОМ ADDR.

МАКРОКОМАНДА ДАЕТ ВОЗМОЖНОСТЬ ВЫПОЛНЯТЬ ИЛИ ПРОПУСКАТЬ НЕКОТОРЫЕ ЧАСТИ ТЕСТОВОЙ ПРОГРАММЫ В ЗАВИСИМОСТИ ОТ СОСТОЯНИЯ ФЛАГА УПРАВЛЕНИЯ ОПЕРАТОРА.

#### 7.6.4. МАКРОКОМАНДЫ УПРАВЛЕНИЯ ПРОГРАММНЫМ ПОДТЕСТОМ

В ГРУППУ ВХОДЯТ МАКРОКОМАНДЫ #DS\_BGN SUB И #DS\_END SUB, КОТОРЫЕ ЗАДАЮТ ГРАНИЦЫ (НАЧАЛЬНУЮ И КОНЕЧНУЮ) КАЖДОГО ПОДТЕСТА.

ПАРЫ МАКРОКОМАНД ЗАПИСЫВАЮТСЯ МЕЖДУ МАКРОКОМАНДАМИ #DS\_BGN TEST (НАЧАЛО ТЕСТА) И #DS\_END TEST (КОНЕЦ ТЕСТА), КАК ЭТО ПОКАЗАНО В ПРИМЕРЕ.

ПРИМЕР.

#DS_BGN TEST	;НАЧАЛО ТЕСТА
#DS_BGN SUB	;НАЧАЛО ПОДТЕСТА 1
ПОДТЕСТ 1	
#DS_END SUB	;КОНЕЦ ПОДТЕСТА 1
#DS_BGN SUB	;НАЧАЛО ПОДТЕСТА 2
ПОДТЕСТ 2	
#DS_END SUB	;КОНЕЦ ПОДТЕСТА 2
#DS_END TEST	;КОНЕЦ ТЕСТА

##### 7.6.4.1. #DS\_BGN SUB - НАЧАЛО ПОДТЕСТА

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_BGN SUB

АРГУМЕНТОВ В МАКРОКОМАНДЕ ЧЕТ.

МАКРОКОМАНДА #DS\_BGN SUB ЗАПИСЫВАЕТСЯ В НАЧАЛЕ КАЖДОГО ПОДТЕСТА В ТЕСТЕ. ОНА ГЕНЕРИРУЕТ МЕТКУ ВХОДА TN\_SM (ГДЕ N - НОМЕР ТЕСТА, M - НОМЕР ПОДТЕСТА В ТЕСТЕ). КРОМЕ ТОГО, ДАННАЯ МАКРОКОМАНДА ФОРМИРУЕТ В СЕКЦИИ ARG LIST ДЛИННЫЕ СЛОВА С НОМЕРАМИ ТЕСТА И ПОДТЕСТА.

13.00078-01 33 01

МАКРОКОМАНДА `ADS_BGN SUB` ВЫЗЫВАЕТ ПОДПРОГРАММУ СУПЕРВИЗОРА, КОТОРАЯ ПРОВЕРЯЕТ ПРОХОДИТ ЛИ ДИАГНОСТИЧЕСКАЯ ПРОГРАММА ПОСЛЕДОВАТЕЛЬНО В ПОРЯДКЕ ВОЗРАСТАНИЯ НОМЕРОВ ЧЕРЕЗ КАЖДЫЙ ПОДТЕСТ В ТЕСТЕ.

ЕСЛИ ПОДПРОГРАММА ОБНАРУЖИВАЕТ ОШИБКУ ПОСЛЕДОВАТЕЛЬНОСТИ, ОНА ИЗВЕЩАЕТ ОБ ЭТОМ ОПЕРАТОРА И ПЕРЕДАЕТ УПРАВЛЕНИЕ ИНТЕРПРЕТАТОРУ КОМАНДНОЙ СТРОКИ.

#### 7.6.4.2. `ADS_END SUB` - КОНЕЦ ПОДТЕСТА

ФОРМАТ МАКРОКОМАНДЫ:

`ADS_END SUB`

АРГУМЕНТОВ В МАКРОКОМАНДЕ ЧЕТ.

МАКРОКОМАНДУ СЛЕДУЕТ ПИСАТЬ В КОНЦЕ КАЖДОГО ПОДТЕСТА. ОНА ГЕНЕРИРУЕТ МЕТКУ ВЫХОДА `TN_SM_X` (`N` - НОМЕР ТЕСТА, `M` - НОМЕР ПОДТЕСТА). МАКРОКОМАНДА `ADS_END SUB` ВЫЗЫВАЕТ ПОДПРОГРАММУ СУПЕРВИЗОРА, КОТОРАЯ ВЫПОЛНЯЕТ ПРОВЕРКУ НОМЕРОВ ТЕСТОВ И НОМЕРОВ ПОДТЕСТА В ПРЕДЕЛАХ КАЖДОГО ТЕСТА НА ПРАВИЛЬНОСТЬ СЛЕДОВАНИЯ НОМЕРОВ ТЕСТОВ И ПОДТЕСТОВ В КАЖДОМ ПОДТЕСТЕ. ПРИ ОБНАРУЖЕНИИ ОШИБКИ СЛЕДОВАНИЯ, ПЕЧАТАЕТСЯ ИЗВЕЩЕНИЕ ОБ ЭТОМ И УПРАВЛЕНИЕ ВОЗВРАЩАЕТСЯ КОМАНДНОМУ РЕЖИМУ.

#### 7.6.5. МАКРОКОМАНДЫ ЗАЦИКЛИВАНИЯ

В ГРУППУ ВХОДЯТ МАКРОКОМАНДЫ `ADS_SK LOOP` И `ADS_IN LOOP`. ПЕРВАЯ ИЗ НИХ ДАЕТ ВОЗМОЖНОСТЬ ПРОИЗВОДИТЬ ЗАЦИКЛИВАНИЕ ТОЙ ЧАСТИ ПРОГРАММЫ, В КОТОРОЙ ОБНАРУЖЕНА ОШИБКА, ВТОРАЯ - ПРОВЕРЯЕТ, "ЗАЦИКЛИЛАСЬ" ЛИ ПРОГРАММА НА ОШИБКЕ.

#### 7.6.5.1. #DS\_SCKLOOP - УПРАВЛЕНИЕ ЗАЦИКЛИВАНИЕМ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_SCKLOOP LABEL

ГДЕ LABEL - АДРЕС ПЕРЕХОДА.

МАКРОКОМАНДА #DS\_SCKLOOP ВЫЗЫВАЕТ ПОДПРОГРАММУ СУПЕРВИЗОРА, КОТОРАЯ УПРАВЛЯЕТ МЕХАНИЗМОМ ЗАЦИКЛИВАНИЯ ПОДТЕСТА.

ЕСЛИ ОБНАРУЖЕНА ОШИБКА (УСТАНОВЛЕН ФЛАГ ОШИБКИ) И УСТАНОВЛЕН УПРАВЛЯЮЩИЙ ФЛАГ ЦИКЛА (LOOP), ПОДПРОГРАММА СУПЕРВИЗОРА ВЫЗЫВАЕТ ПЕРЕХОД ПО АДРЕСУ, УКАЗАННОМУ АРГУМЕНТОМ LABEL, ТАКИМ ОБРАЗОМ ПРОИСХОДИТ ЗАЦИКЛИВАНИЕ ОПРЕДЕЛЕННОЙ ЧАСТИ ПРОГРАММЫ, ЧТО СОЗДАЕТ ВОЗМОЖНОСТЬ ПРОСМОТРА НА ОСЦИЛЛОГРАФЕ СИГНАЛОВ В РАЗЛИЧНЫХ ТОЧКАХ СХЕМЫ ДЛЯ ВЫЯСНЕНИЯ ПРИЧИНЫ ОШИБКИ.

МАКРОКОМАНДА #DS\_SCKLOOP ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ В СОЧЕТАНИИ СО СЛЕДУЮЩИМИ МАКРОКОМАНДАМИ:

#DS\_BGNTEST, #DS\_ENDTEST, #DS\_BGN SUB И #DS\_ENDSUB.

#### 7.6.5.2. #DS\_INLOOP\_X - ПРОВЕРКА ЗАЦИКЛИВАНИЯ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_INLOOP\_X

АРГУМЕНТОВ У МАКРОКОМАНДЫ НЕТ.

МАКРОКОМАНДА ВЫЗЫВАЕТ СЛУЖЕБНУЮ ПОДПРОГРАММУ СУПЕРВИЗОРА, КОТОРАЯ ПРОВЕРЯЕТ "ЗАЦИКЛИЛАСЬ" ЛИ ПРОГРАММА НА ОШИБКЕ ИЛИ НЕТ. ЕСЛИ ПРОГРАММА НАХОДИТСЯ В ЦИКЛЕ, ТО МЛАДШИЙ БИТ РЕГИСТРА RD УСТАНАВЛИВАЕТСЯ, ЕСЛИ НЕТ - ТО ОЧИЩАЕТСЯ.

МАКРОКОМАНДА #DS\_INLOOP\_X ВОЗВРАЩАЕТ СЛЕДУЮЩИЕ КОДЫ СОСТОЯНИЯ:

- 1) #DS\_NORMAL - ПРОГРАММА НАХОДИТСЯ В ЦИКЛЕ.
- 2) #DS\_ERROR - ПРОГРАММА НЕ НАХОДИТСЯ В ЦИКЛЕ.

7.6.5.3. ПРИМЕР ИСПОЛЬЗОВАНИЯ МАКРОКОМАНДЫ  
ЗАЦИКЛИВАНИЯ

```
#DS_SBTTL TESTS
#DS_BGNTST
#DS_BGNSUB
LABEL_1:
    ; ПЕРВАЯ ЧАСТЬ ПОДТЕСТА 1
#DS_CKLOOP LABEL_1
LABEL_2:
    ; ВТОРАЯ ЧАСТЬ ПОДТЕСТА 1
#DS_CKLOOP LABEL_2
#DS_ENDSUB
#DS_BGNSUB
LABEL_3:
    ; ПОДТЕСТ 2
#DS_CKLOOP LABEL_3
#DS_ENDSUB
    .
    .
    .
#DS_ENDTEST
```

ЕСЛИ ПРОИСХОДИТ ОШИБКА В ПЕРВОЙ ЧАСТИ ПОДТЕСТА 1, ТО ЦИКЛ БУДЕТ ВЫПОЛНЯТЬСЯ ОТ ПЕРВОЙ МАКРОКОМАНДЫ #DS\_CKLOOP ОБРАТНО К МЕТКЕ LABEL\_1.

АНАЛОГИЧНО, ЕСЛИ ОШИБКА БУДЕТ ВО ВТОРОЙ ЧАСТИ ПОДТЕСТА 1, ТО ЦИКЛ БУДЕТ ВЫПОЛНЯТЬСЯ ОТ ВТОРОЙ МАКРОКОМАНДЫ #DS\_CKLOOP ОБРАТНО К МЕТКЕ LABEL\_2.

СЛЕДУЕТ ОТМЕТИТЬ, ЧТО АРГУМЕНТ, УКАЗАННЫЙ В МАКРОКОМАНДЕ #DS\_CKLOOP, ДОЛЖЕН УКАЗЫВАТЬ АДРЕС В ПРЕДЕЛАХ ТОГО ЖЕ ПОДТЕСТА. ЕСЛИ ТЕСТ НЕ ИМЕЕТ ПОДТЕСТОВ, ТО ЗАЦИКЛИВАНИЕ ДОЛЖНО ПРОИСХОДИТЬ ВНУТРИ ОДНОГО ТЕСТА.

7.6.5. МАКРОКОМАНДЫ ВЫХОДА ИЗ ПРОГРАММЫ И  
ПОДПРОГРАММЫ

В ГРУППУ ВХОДЯТ ТРИ МАКРОКОМАНДЫ:

#DS\_EXIT, #DS\_ESCAPE И #DS\_ABORT.

МАКРОКОМАНДЫ ПОЗВОЛЯЮТ ИСКЛЮЧАТЬ ОПРЕДЕЛЕННЫЕ ЧАСТИ ПРОГРАММЫ ПРИ ФИКСАЦИИ ОШИБКИ ИЛИ ПО ТРЕБОВАНИЮ ОПЕРАТОРА.

19.00078-01 33 01

МАКРОКОМАНДА #DS\_EXIT ВЫПОЛНЯЕТ БЕЗУСЛОВНЫЙ ПЕРЕХОД К ПОСЛЕДНЕМУ ОПЕРАТОРУ ПОДПРОГРАММЫ ИЛИ ПОДТЕСТА.

МАКРОКОМАНДА #DS\_ESCAPE ОБЕСПЕЧИВАЕТ УСЛОВНЫЙ ВЫХОД ИЗ ТЕСТА ИЛИ ПОДТЕСТА В СООТВЕТСТВИИ С СОСТОЯНИЕМ ОШИБКИ.

МАКРОКОМАНДА #DS\_AБОРТ ОТМЕНЯЕТ ВЫПОЛНЕНИЕ ТЕСТА ИЛИ ВСЕЙ ПРОГРАММЫ.

#### 7.6.6.1. #DS\_ESCAPE - УСЛОВНЫЙ ВЫХОД ИЗ ТЕСТА ИЛИ ПОДТЕСТА

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_ESCAPE ARG

ГДЕ ARG - SUB ИЛИ TEST

МАКРОКОМАНДА ОБЕСПЕЧИВАЕТ УСЛОВНЫЙ ВЫХОД ИЗ ТЕСТА (АРГУМЕНТ ARG РАВЕН TEST) ИЛИ ПОДТЕСТА (АРГУМЕНТ ARG РАВЕН SUB) ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ.

МАКРОКОМАНДА #DS\_ESCAPE ВЫЗЫВАЕТ ПОДПРОГРАММУ СУПЕРВИЗОРА, КОТОРАЯ ПРОВЕРЯЕТ СОСТОЯНИЕ ФЛАГА ОШИБКИ. ЕСЛИ ФЛАГ УСТАНОВЛЕН, ТО УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ В КОНЕЦ ПОДТЕСТА ИЛИ ТЕСТА В ЗАВИСИМОСТИ ОТ УКАЗАННОГО АРГУМЕНТА. МАКРОКОМАНДА ДОЗВОЛЯЕТ ИСКЛЮЧИТЬ ВЫПОЛНЕНИЕ ОПРЕДЕЛЕННЫХ ЧАСТЕЙ ПРОГРАММЫ, ЕСЛИ В ПРЕДШЕСТВУЮЩЕЙ ЧАСТИ ПРОГРАММЫ ОБНАРУЖЕНА ТАКАЯ ОШИБКА, ЧТО ДАЛЬНЕЙШЕЕ ВЫПОЛНЕНИЕ ТЕСТА ИЛИ ПОДТЕСТА БЕСПОЛЕЗНО.

ИСПОЛНЕНИЕ МАКРОКОМАНДЫ #DS\_ESCAPE НЕ ИЗМЕНЯЕТ СОДЕРЖИМОГО РЕГИСТРА RO.

ЕСЛИ МАКРОКОМАНДА #DS\_ESCAPE ДОЛЖНА ВЫПОЛНЯТЬСЯ В ТОЙ ЧАСТИ ТЕСТА, КОТОРАЯ НАХОДИТСЯ ВНУТРИ ЦИКЛА, ОБРАЗУЕМОГО МАКРОКОМАНДОЙ #DS\_SCKLOOP, ТО ОНА (МАКРОКОМАНДА #DS\_ESCAPE) ДОЛЖНА БЫТЬ РАЗМЕЩЕНА ПОСЛЕ МАКРОКОМАНДЫ #DS\_SCKLOOP, КАК

13.00078-01 33 01

ПОКАЗАНО В ПРИМЕРЕ.

ПРИМЕР.

```
#DS_SBTTL TESTA
#DS_BGNTST
.
.
#DS_BGNSUB
LABEL_1:
    ПЕРВАЯ ЧАСТЬ ПОДТЕСТА1
LABEL_2:
    ВТОРАЯ ЧАСТЬ ПОДТЕСТА1
#DS_CKLOOP LABEL_2
#DS_ESCAPE SUB
LABEL_3:
    ТРЕТЬЯ ЧАСТЬ ПОДТЕСТА1
#DS_ENDSUB
#DS_BGNSUB
    ПОДТЕСТ 2
#DS_ENDSUB
#DS_ENDTEST
```

ЕСЛИ ОШИБКА ПРОИСХОДИТ ВО ВТОРОЙ ЧАСТИ ПОДТЕСТА1 И УСТАНОВЛЕН ФЛАГ ЦИКЛА, ТО БУДЕТ ПРОИСХОДИТЬ ЗАЦИКЛИВАНИЕ ОТ МАКРОКОМАНДЫ #DS\_CKLOOP НА МЕТКУ LABEL\_2. НО ЕСЛИ ОШИБКА ПРОИСХОДИТ ВО ВТОРОЙ ЧАСТИ ПОДТЕСТА1, КОГДА ФЛАГ ЦИКЛА НЕ УСТАНОВЛЕН, ТО МАКРОКОМАНДА #DS\_ESCAPE БУДЕТ ПЕРЕДАВАТЬ УПРАВЛЕНИЕ ВТОРОМУ ПОДТЕСТУ, ИСКЛЮЧАЯ ТАКИМ ОБРАЗОМ ВЫПОЛНЕНИЕ ТРЕТЬЕЙ ЧАСТИ ПОДТЕСТА1.

#### 7.6.6.2. #DS\_EXIT - БЕЗУСЛОВНЫЙ ВЫХОД ИЗ ПРОГРАММНОЙ ФАЗЫ

ФОРМАТ МАКРОКОМАНДЫ:

```
#DS_EXIT ARG
```

ГДЕ ARG - TEST, SUB, INIT ИЛИ CLEAN.

МАКРОКОМАНДА ГЕНЕРИРУЕТ КОМАНДУ БЕЗУСЛОВНОГО ПЕРЕХОДА К ПОСЛЕДНЕМУ ОПЕРАТОРУ ПРОГРАММНОЙ ФАЗЫ В ЗАВИСИМОСТИ ОТ АРГУМЕНТА ARG. ЕСЛИ АРГУМЕНТ TEST-УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ В КОНЕЦ СЕКЦИИ ТЕКУЩЕГО ТЕСТА, SUB-УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ В КОНЕЦ ТЕКУЩЕГО ПОДТЕСТА, INIT-В КОНЕЦ СЕКЦИИ ИНИЦИАЛИЗАЦИИ,



13.00078-01 33 01

CLEAN-СЕКЦИИ ОЧИСТКИ.

МАКРОКОМАНДА #DS\_EXIT ИСПОЛЬЗУЕТСЯ ДЛЯ ПРОПУСКА ВЫПОЛНЕНИЯ ЧАСТИ ПОДПРОГРАММЫ ПРИ НЕКОТОРЫХ УСЛОВИЯХ.

ПРИМЕР.

```
#DS_SBTTL TESTB
#DS_BGNTST
#DS_BNQUICK 2#
#DS_EXIT TEST
2#:
```

ТЕСТОВАЯ ПОДПРОГРАММА  
#DS\_ENDTEST

В ПРИМЕРЕ ПРИ ОТСУТСТВИИ ФЛАГА БЫСТРОЙ ПРОВЕРКИ (QUICK) МАКРОКОМАНДА #DS\_BNQUICK ПЕРЕДАЕТ УПРАВЛЕНИЕ К МЕТКЕ 2# И ВЫПОЛНЯЕТСЯ ТЕСТОВАЯ ПОДПРОГРАММА.

ЕСЛИ ЖЕ УСТАНОВЛЕН ФЛАГ БЫСТРОЙ ПРОВЕРКИ, ТО СРАБАТЫВАЕТ МАКРОКОМАНДА #DS\_EXIT, КОТОРАЯ ОСУЩЕСТВЛЯЕТ ПЕРЕХОД В КОНЕЦ ТЕСТА К МАКРОКОМАНДЕ #DS\_ENDTEST. ТАКИМ ОБРАЗОМ ИСКЛЮЧАЕТСЯ ВЫПОЛНЕНИЕ ТЕСТОВОЙ ПОДПРОГРАММЫ.

#### 7.6.6.3. #DS\_ABORT - ОТМЕНА ПРОГРАММЫ ИЛИ ТЕСТА

ФОРМАТ МАКРОКОМАНДЫ:

```
#DS_ABORT ARG
```

ГДЕ ARG - УРОВЕНЬ ОТМЕНЫ: PROGRAM (ПРОГРАММА) ИЛИ TEST (ТЕСТ).

ПО УМОЛЧАНИЮ ИСПОЛЬЗУЕТСЯ АРГУМЕНТ PROGRAM.

МАКРОКОМАНДА #DS\_ABORT ПОЗВОЛЯЕТ ОТМЕНЯТЬ ВЫПОЛНЕНИЕ ТЕКУЩЕГО ТЕСТА ИЛИ ВСЕЙ ПРОГРАММЫ В ЗАВИСИМОСТИ ОТ АРГУМЕНТА ARG. ЕСЛИ УКАЗАН АРГУМЕНТ PROGRAM, ТО ВЫЗЫВАЕТСЯ ПОДПРОГРАММА СУПЕРВИЗОРА, КОТОРАЯ ПЕЧАТАЕТ СООБЩЕНИЕ ОБ ОТМЕНЕ ПРОГРАММЫ, ВЫПОЛНЯЕТ ПОДПРОГРАММУ ОЧИСТКИ И ПЕРЕДАЕТ УПРАВЛЕНИЕ СУПЕРВИЗОРУ.

ЕСЛИ АРГУМЕНТОМ ЯВЛЯЕТСЯ TEST, ТО МАКРОКОМАНДА

13.00078-01 33 01

ADS\_ABORT ОЧИЩАЕТ РЕГИСТР RD (ЧТО СВИДЕТЕЛЬСТВУЕТ О СБОЕ В ТЕСТЕ), ЗАТЕМ ВЫПОЛНЯЕТ ИНСТРУКЦИЮ ВОЗВРАТА RET, ЧТО ПРИВОДИТ К ПЕРЕДАЧЕ УПРАВЛЕНИЯ ДИСПЕТЧЕРУ СУПЕРВИЗОРА, КОТОРЫЙ ЗАПУСКАЕТ СЛЕДУЮЩИЙ ТЕСТ.

МАКРОКОМАНДА ADS\_ABORT ДОЛЖНА ИСПОЛЬЗОВАТЬСЯ В ТОМ СЛУЧАЕ, КОГДА ОБНАРУЖИВАЕТСЯ ОШИБКА, НЕ ПОЗВОЛЯЮЩАЯ ПРАВИЛЬНО ВЫПОЛНИТЬ ПРОГРАММУ ИЛИ ТЕСТ. НАПРИМЕР, ЕСЛИ В ПОДПРОГРАММЕ ИНИЦИАЛИЗАЦИИ ОБНАРУЖИВАЕТСЯ СИСТЕМНАЯ ОШИБКА, ТО ПРОГРАММА ДОЛЖНА БЫТЬ ОТМЕНЕНА.

РАССМОТРИМ ПРИМЕР, ИЛЛЮСТРИРУЮЩИЙ ИСПОЛЬЗОВАНИЕ МАКРОКОМАНДЫ ADS\_ABORT.

ПРИМЕР.

```
*****
ADS_BPHARD_S      ;ПОЛУЧЕНИЕ АДРЕСА Р-ТАБЛИЦЫ
ADS_BNERROR SUP   ;ЕСТЬ ОШИБКА ПРИ ВЫПОЛНЕНИИ ПРЕДЫДУЩЕЙ
                  ;ПОДПРОГРАММЫ
ADS_ERRSYS_S      ;ДА. ПЕЧАТЬ СООБЩЕНИЯ О ФАТАЛЬНОЙ
                  ;ОШИБКЕ
ADS_ABORT PROGRAM ;ОТМЕНА ВЫПОЛНЕНИЯ ПРОГРАММЫ
SUP: MOVL A,R2    ;ПРОДОЛЖЕНИЕ ВЫПОЛНЕНИЯ ПРОГРАММЫ
```

7.6.5.4. ADS\_BREAK - ВКЛИНИВАНИЕ В

ДИАГНОСТИЧЕСКУЮ ПРОГРАММУ

ФОРМАТ МАКРОКОМАНДЫ:

ADS\_BREAK

АРГУМЕНТОВ У МАКРОКОМАНДЫ НЕТ.

МАКРОКОМАНДА ОБЕСПЕЧИВАЕТ ВЫЗОВ ПОДПРОГРАММЫ СУПЕРВИЗОРА DS\_BREAK ДЛЯ ПРОВЕРКИ НАЛИЧИЯ КОМАНДЫ ОПЕРАТОРА CONTROL C.

ЕСЛИ ОПЕРАТОР ВВЕЛ С КОНСОЛИ КОМАНДУ CONTROL C, ТО ПОДПРОГРАММА DSBREAK ПРЕРВЕТ ВЫПОЛНЕНИЕ ПРОГРАММЫ И ПЕРЕДАСТ УПРАВЛЕНИЕ ИНТЕРПРЕТАТОРУ КОМАНДНОЙ СТРОКИ (CLI).

МАКРОКОМАНДА ADS\_BREAK ПОЗВОЛИТ ОПЕРАТОРУ ПРЕРВАТЬ

13.00078-01 33 01

ВЫПОЛНЕНИЕ ПРОГРАММЫ КОМАНДОЙ CONTROL C. ЕЕ НЕОБХОДИМО ИСПОЛЬЗОВАТЬ ВНУТРИ ДЛИННЫХ ЦИКЛОВ С ТЕМ, ЧТОБЫ ОПЕРАТОРУ НЕ БЫЛО НЕОБХОДИМОСТИ ДОЛГО ЖДАТЬ ПРЕРЫВАНИЯ ПРОГРАММЫ ПРИ ПОДАЧЕ КОМАНДЫ CONTROL C.

ПОДПРОГРАММА DSABREAK ВЫЗЫВАЕТСЯ ПРИ ОБРАЩЕНИИ К ЛЮБОЙ СЛУЖЕБНОЙ ПОДПРОГРАММЕ СУПЕРВИЗОРА.

ОНА ЯВЛЯЕТСЯ ЕДИНСТВЕННЫМ МЕХАНИЗМОМ СУПЕРВИЗОРА, КОТОРЫЙ ПОЗВОЛЯЕТ ПРОВЕРИТЬ СОСТОЯНИЕ ФЛАГА CONTROL C.

ЕСЛИ В ДИАГНОСТИЧЕСКОЙ ПРОГРАММЕ ИМЕЮТСЯ ЦИКЛЫ, ВЫПОЛНЕНИЕ КОТОРЫХ ЗАНИМАЕТ ДЛИТЕЛЬНОЕ ВРЕМЯ И В ЭТИХ ЦИКЛАХ НЕТ ОБРАЩЕНИЯ К ПОДПРОГРАММАМ СУПЕРВИЗОРА, ТО ОПЕРАТОР НЕ СМОЖЕТ ЗА КОРОТКОЕ ВРЕМЯ ПРЕРВАТЬ ПРОГРАММУ С ПОМОЩЬЮ КОМАНДЫ CONTROL C. ПОЭТОМУ ЛЮБОЙ ПОТЕНЦИАЛЬНЫЙ ЦИКЛ, НЕ ВКЛЮЧАЮЩИЙ ОБРАЩЕНИЕ К ПОДПРОГРАММЕ СУПЕРВИЗОРА, ДОЛЖЕН СОДЕРЖАТЬ ВЫЗОВ МАКРОКОМАНДЫ #DS\_BREAK.

#### 7.5.7. МАКРОКОМАНДЫ ПЕРЕХОДА ПО ОКОНЧАНИИ ВЫПОЛНЕНИЯ ПОДПРОГРАММЫ

В ГРУППУ ВХОДЯТ МАКРОКОМАНДЫ #DS\_VCOMPLETE И #DS\_VNCOMPLETE. ОНИ ПРОИЗВОДЯТ ПРОВЕРКУ СОСТОЯНИЯ МЛАДШЕГО РАЗРЯДА РЕГИСТРА RD И В ЗАВИСИМОСТИ ОТ ЕГО ЗНАЧЕНИЯ МОГУТ ВЫПОЛНИТЬ УСЛОВНЫЙ ПЕРЕХОД.

ПО ПРИНЯТОМУ СОГЛАШЕНИЮ ПРИ УСПЕШНОМ ВЫПОЛНЕНИИ ПОДПРОГРАММЫ МЛАДШИЙ БИТ РЕГИСТРА RD УСТАНОВЛИВАЕТСЯ, В ПРОТИВНОМ СЛУЧАЕ СНИЖАЕТСЯ. В ДРУГИХ БИТАХ ЭТОГО РЕГИСТРА МОГУТ РАСПОЛАГАТЬСЯ РАЗЛИЧНЫЕ КОМБИНАЦИИ В ЗАВИСИМОСТИ ОТ ПРИЧИНЫ СБОЯ.

13.00078-01 33 01

ПРИМЕР.

ADS_CHANNEL_S			ЗВЫЗОВ ПОДПРОГРАММЫ
			ЗОБСЛУЖИВАНИЯ КАНАЛА
ADS_VSCOMPLETE	ZUP		ЗПОДПРОГРАММА ВЫПОЛНЕНА
			ЗУСПЕШНО
ADS_ERRSYS_G	LIST		ЗНЕТ. ПЕЧАТЬ СООБЩЕНИЯ
			ЗОБ ОШИБКЕ
ADS_ABORT TEST			ЗВЫХОД ИЗ ТЕСТА
Z04:	INCL	R>	ЗДА. ПРОДОЛЖЕНИЕ
			ЗВЫПОЛНЕНИЯ ТЕСТА

7.6.7.1. ADS\_VSCOMPLETE - ПЕРЕХОД ПРИ УСПЕШНОМ  
ВЫПОЛНЕНИИ ПОДПРОГРАММЫ

ФОРМАТ МАКРОКОМАНДЫ:

ADS\_VSCOMPLETE ADR

ГДЕ ADR - АДРЕС (МЕТКА) ПЕРЕХОДА ПРИ УСПЕШНОМ ВЫПОЛНЕНИИ  
ПОДПРОГРАММЫ.

МАКРОКОМАНДА ПРОВЕРЯЕТ МЛАДШИИ БИТ РЕГИСТРА R0. ЕСЛИ  
УКАЗАННЫЙ БИТ УСТАНОВЛЕН, ЧТО ОЗНАЧАЕТ УСПЕШНОЕ ВЫПОЛНЕНИЕ  
ПРЕШЕСТВУЮЩЕЙ ПОДПРОГРАММЫ, ОСУЩЕСТВЛЯЕТСЯ ПЕРЕХОД ПО АДРЕ-  
СУ, УКАЗАННОМУ АРГУМЕНТОМ ADR.

МАКРОКОМАНДА ADS\_VSCOMPLETE ЗАПИСЫВАЕТСЯ ПОСЛЕ ОБРАЩЕ-  
НИЯ К ПОДПРОГРАММЕ В ТЕХ СЛУЧАЯХ, КОГДА В ЗАВИСИМОСТИ ОТ  
РЕЗУЛЬТАТА ЕЕ ВЫПОЛНЕНИЯ (УСПЕШНО ИЛИ НЕТ) НЕОБХОДИМО  
ПРОИЗВЕСТИ ВЕТВЛЕНИЕ ПРОГРАММЫ. ЗАМЕТИМ, ЧТО ПО ПРИНЯТОМУ  
СОГЛАШЕНИЮ ПРИ УСПЕШНОМ ВЫПОЛНЕНИИ ПОДПРОГРАММЫ В МЛАДШИИ  
БИТ РЕГИСТРА R0 ЗАСЫЛАЕТСЯ 1, А В ПРОТИВНОМ СЛУЧАЕ - 0.

7.6.7.2. ADS\_VNSCOMPLETE - ПЕРЕХОД ПРИ НЕУСПЕШНОМ  
ВЫПОЛНЕНИИ ПОДПРОГРАММЫ

ФОРМАТ МАКРОКОМАНДЫ:

ADS\_VNSCOMPLETE ADR

ГДЕ ADR - АДРЕС (МЕТКА) ПЕРЕХОДА ПРИ НЕУСПЕШНОМ

19.00078-01 33 01

ВЫПОЛНЕНИИ ПОДПРОГРАММЫ.

МАКРОКОМАНДА ПРОВЕРЯЕТ МЛАДШИЙ БИТ РЕГИСТРА RO. ЕСЛИ УКАЗАННЫЙ БИТ ОЧИЩЕН, ЧТО ОЗНАЧАЕТ НЕУСПЕШНОЕ ВЫПОЛНЕНИЕ ПРЕДШЕСТВУЮЩЕЙ ПОДПРОГРАММЫ, ОСУЩЕСТВЛЯЕТСЯ ПЕРЕХОД ПО АДРЕСУ, УКАЗАННОМУ АРГУМЕНТОМ ADR.

МАКРОКОМАНДА #DS\_VNCOMPLETE ЗАПИСЫВАЕТСЯ ПОСЛЕ ВЫЗОВА ПОДПРОГРАММЫ, ЕСЛИ В ЗАВИСИМОСТИ ОТ РЕЗУЛЬТАТА ЕЕ ВЫПОЛНЕНИЯ (УСПЕШНО ИЛИ НЕТ) НЕОБХОДИМО РАЗВЕТВИТЬ ПРОГРАММУ.

#### 7.6.8. МАКРОКОМАНДЫ ПЕРЕХОДА ПРИ ОШИБКЕ

В ГРУППУ ВХОДЯТ МАКРОКОМАНДЫ #DS\_BERROR И #DS\_VNERROR. ЭТИ МАКРОКОМАНДЫ ГЕНЕРИРУЮТ ИНСТРУКЦИИ, КОТОРЫЕ ПРОВЕРЯЮТ СОДЕРЖИМОЕ РЕГИСТРА RO И В ЗАВИСИМОСТИ ОТ ЕГО ЗНАЧЕНИЯ МОГУТ ВЫПОЛНИТЬ УСЛОВНЫЙ ПЕРЕХОД.

ЕСЛИ ПРИ ВЫПОЛНЕНИИ ПОДПРОГРАММЫ НЕ ОБНАРУЖЕНО ОШИБКИ, ТО МЛАДШИЙ БИТ РЕГИСТРА RO УСТАНОВЛИВАЕТСЯ, В ПРОТИВНОМ СЛУЧАЕ - ОЧИЩАЕТСЯ.

##### 7.6.8.1. #DS\_BERROR - ПЕРЕХОД ПРИ ОШИБКЕ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_BERROR ADR

ГДЕ ADR - АДРЕС (МЕТКА) ПЕРЕДАЧИ УПРАВЛЕНИЯ ПРИ ОБНАРУЖЕНИИ ОШИБКИ.

МАКРОКОМАНДА ПРОВЕРЯЕТ МЛАДШИЙ БИТ РЕГИСТРА RO. ЕСЛИ УКАЗАННЫЙ БИТ ОЧИЩЕН, ЧТО УКАЗЫВАЕТ НА ОБНАРУЖЕНИЕ ОШИБКИ ПРИ ВЫПОЛНЕНИИ ПРЕДШЕСТВУЮЩЕЙ ПОДПРОГРАММЫ, ТО ПРОИСХОДИТ ПЕРЕХОД ПО АДРЕСУ, УКАЗАННОМУ АРГУМЕНТОМ ADR.

МАКРОКОМАНДА #DS\_BERROR ЗАПИСЫВАЕТСЯ ПОСЛЕ ВЫЗОВА СЛУЖБНОЙ ПОДПРОГРАММЫ СУПЕРВИЗОРА ИЛИ ОБРАЩЕНИЯ К ТЕСТОВОЙ

19.00078-01 33 01

ПОДПРОГРАММЕ В ТЕХ СЛУЧАЯХ, КОГДА В ЗАВИСИМОСТИ ОТ РЕЗУЛЬТАТА ЕЕ ВЫПОЛНЕНИЯ (ОБНАРУЖЕНА ОШИБКА ИЛИ НЕТ) НЕОБХОДИМО ПРОИЗВЕСТИ ВЕТВЛЕНИЕ ПРОГРАММЫ.

ПО ПРИНЯТОМУ СОГЛАШЕНИЮ ПРИ ОБНАРУЖЕНИИ ОШИБКИ В МЛАДШИИ БИТ РЕГИСТРА R0 ЗАСЫЛАЕТСЯ 0, В ПРОТИВНОМ СЛУЧАЕ - 1.

#### 7.6.8.2. #DS\_3NERROR - ПЕРЕХОД ПРИ ОТСУТСТВИИ ОШИБКИ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_3NERROR ADR

ГДЕ ADR - АДРЕС (МЕТКА) ПЕРЕДАЧИ УПРАВЛЕНИЯ ПРИ ОТСУТСТВИИ ОШИБКИ.

МАКРОКОМАНДА ПРОВЕРЯЕТ МЛАДШИИ БИТ РЕГИСТРА R0. ЕСЛИ УКАЗАННЫЙ БИТ УСТАНОВЛЕН, ЧТО УКАЗЫВАЕТ НА ОТСУТСТВИЕ ОШИБКИ ПРИ ВЫПОЛНЕНИИ ПРЕДШЕСТВУЮЩЕЙ ПОДПРОГРАММЫ, ТО ПРОИСХОДИТ ПЕРЕХОД ПО АДРЕСУ, УКАЗАННОМУ АРГУМЕНТОМ ADR.

МАКРОКОМАНДА #DS\_3NERROR ЗАПИСЫВАЕТСЯ ПОСЛЕ ВЫЗОВА СЛУЖЕБНОЙ ПОДПРОГРАММЫ СУПЕРВИЗОРА ИЛИ ТЕСТОВОЙ ПОДПРОГРАММЫ В ТЕХ СЛУЧАЯХ, КОГДА В ЗАВИСИМОСТИ ОТ РЕЗУЛЬТАТА ЕЕ ВЫПОЛНЕНИЯ (ОБНАРУЖЕНА ОШИБКА ИЛИ НЕТ) НЕОБХОДИМО ПРОИЗВЕСТИ ВЕТВЛЕНИЕ ПРОГРАММЫ.

### 7.7. МАКРОКОМАНДЫ P-ТАБЛИЦЫ

#### 7.7.1. ОБЩИЕ СВЕДЕНИЯ О P-ТАБЛИЦЕ

ДИАГНОСТИЧЕСКАЯ ПРОГРАММА ДОЛЖНА ЗНАТЬ ХАРАКТЕРИСТИКИ ТЕСТИРУЕМОГО УСТРОЙСТВА. НЕКОТОРЫЕ ХАРАКТЕРИСТИКИ ЯВЛЯЮТСЯ СПЕЦИФИЧНЫМИ, ПОЭТОМУ ИХ НЕВОЗМОЖНО ЖЕСТКО ОПРЕДЕЛИТЬ В

13.00078-01 33 01

ДИАГНОСТИЧЕСКОЙ ПРОГРАММЕ. ДЛЯ ТОГО ЧТОБЫ НЕОБХОДИМЫЕ ХАРАКТЕРИСТИКИ МОГЛИ БЫТЬ ОПРЕДЕЛЕНЫ ПРИ ИСПОЛНЕНИИ ПРОГРАММЫ, СИСТЕМА МСПД ИСПОЛЬЗУЕТ ТАК НАЗЫВАЕМУЮ Р-ТАБЛИЦУ. В Р-ТАБЛИЦЕ СОДЕРЖИТСЯ ИНФОРМАЦИЯ ОБ УСТРОЙСТВЕ, КОТОРАЯ ТРЕБУЕТСЯ ПРИ ОБРАЩЕНИИ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ К УСТРОЙСТВУ.

Р-ТАБЛИЦЫ СТРОЯТСЯ МСПД ПРИ ВЫПОЛНЕНИИ КОМАНД АТТАСН, ЗАДАВАЕМЫХ ПОЛЬЗОВАТЕЛЕМ. ПО КАЖДОЙ КОМАНДЕ АТТАСН СОЗДАЕТСЯ НОВАЯ Р-ТАБЛИЦА, К КОТОРОЙ ДИАГНОСТИЧЕСКАЯ СИСТЕМА ОБРАЩАЕТСЯ ПРИ ТЕСТИРОВАНИИ УСТРОЙСТВА. ТАКИМ ОБРАЗОМ ОПРЕДЕЛЕНИЕ ХАРАКТЕРИСТИК УСТРОЙСТВА ОТДЕЛЯЕТСЯ ОТ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ.

ПРИ ЗАДАНИИ ПОЛЬЗОВАТЕЛЕМ КОМАНДЫ АТТАСН ПРИСОЕДИНЕНИЯ УСТРОЙСТВА ОДНИМ ИЗ ПАРАМЕТРОВ ЯВЛЯЕТСЯ ЗВЕНО СВЯЗИ. ЭТО ЗВЕНО ПРЕДСТАВЛЯЕТ СОБОЙ ЧАСТЬ АППАРАТУРЫ, К КОТОРОЙ ПОДСОЕДИНЯЕТСЯ ОЧЕРЕДНОЕ УСТРОЙСТВО, ВХОДЯЩЕЕ В ПРОВЕРЯЕМОЕ СОЕДИНЕНИЕ УСТРОЙСТВ. ЗВЕНО СВЯЗИ, В СВОЮ ОЧЕРЕДЬ, ДОЛЖНО БЫТЬ ПРЕДВАРИТЕЛЬНО ОПРЕДЕЛЕНО С ПОМОЩЬЮ ДРУГОЙ КОМАНДЫ АТТАСН ТАК, ЧТОБЫ СООТВЕТСТВУЮЩАЯ Р-ТАБЛИЦА УЖЕ СУЩЕСТВОВАЛА. НАБОР КОМАНД АТТАСН СОЗДАЕТ В РЕЗУЛЬТАТЕ ДЕРЕВООБРАЗНУЮ СТРУКТУРУ СОЕДИНЕНИЙ УСТРОЙСТВ. КОРНЕМ ЭТОГО ДЕРЕВА ЯВЛЯЕТСЯ ПСЕВДОУСТРОЙСТВО HUB. ЭТО ПСЕВДОУСТРОЙСТВО ВВЕДЕНО В СВЯЗИ С ТЕМ, ЧТО РЕАЛЬНОЕ ВЗАИМОДЕЙСТВИЕ УСТРОЙСТВ ЗАВИСИТ ОТ ТИПА ПРОЦЕССОРА. В ОБЩЕМ СЛУЧАЕ ПРОЦЕССОРЫ И ШИНЫ ПОДСОЕДИНЯЮТСЯ К ПСЕВДОУСТРОЙСТВУ HUB, КОНТРОЛЛЕРЫ ПОДСОЕДИНЯЮТСЯ К ШИНАМ, А УСТРОЙСТВА - К КОНТРОЛЛЕРАМ.

Р-ТАБЛИЦА ДЛЯ КОНКРЕТНОГО УСТРОЙСТВА ВКЛИЧАЕТ ИНФОРМАЦИЮ ИЗ АРГУМЕНТОВ КОМАНДЫ АТТАСН, А ТАКЖЕ СОДЕРЖИТ СЛЕДУЮЩУЮ СТАНДАРТНУЮ ИНФОРМАЦИЮ:

13.00078-01 33 01

1) НАЗВАНИЕ УСТРОЙСТВА. ЭТО НАЗВАНИЕ ТЕХНИЧЕСКОГО ПРОДУКТА, НАПРИМЕР, РК06 ИЛИ TMD3;

2) ТИП УСТРОЙСТВА. ЭТО ИМЯ, ПО КОТОРОМУ МЫ ОБРАЩАЕМСЯ К УСТРОЙСТВУ, НАПРИМЕР, DRBT ИЛИ DMA0;

3) АДРЕС Р-ТАБЛИЦЫ ДЛЯ ЗВЕНА СВЯЗИ УСТРОЙСТВА;

4) ХАРАКТЕРИСТИКИ УСТРОЙСТВА. ИНФОРМАЦИЯ, КОТОРАЯ ДОЛЖНА БЫТЬ ВКЛЮЧЕНА В Р-ТАБЛИЦУ ДЛЯ ДОСТАТОЧНОГО ОПИСАНИЯ УСТРОЙСТВА НЕЗАВИСИМО ОТ ТИПА УСТРОЙСТВА И ЕГО ЗВЕНА СВЯЗИ. НАПРИМЕР, ДЛЯ УСТРОЙСТВА, ПОДСОЕДИНЯЕМОГО К UNIBUS ТРЕБУЕТСЯ ЗАДАТЬ АДРЕС РЕГИСТРА CSR НА UNIBUS, УРОВЕНЬ ЗАПРОСА И АДРЕС ВЕКТОРА ПРЕРЫВАНИЯ.

#### 7.7.2. ФОРМАТ Р-ТАБЛИЦЫ

Р-ТАБЛИЦЫ ИМЕЮТ СТАНДАРТНЫЙ ФОРМАТ. КАЖДАЯ Р-ТАБЛИЦА СОСТОИТ ИЗ ДВУХ СЕКЦИИ. ПЕРВАЯ СЕКЦИЯ ВКЛЮЧАЕТ ПОЛЯ, КОТОРЫЕ НЕ ЗАВИСЯТ ОТ УСТРОЙСТВА И ВХОДЯТ В Р-ТАБЛИЦЫ ВСЕХ УСТРОЙСТВ. КАЖДОЕ ТАКОЕ ПОЛЕ ИМЕЕТ СИМВОЛИЧЕСКОЕ ИМЯ, ПО КОТОРОМУ К НЕМУ ОБРАЩАЕТСЯ ДИАГНОСТИЧЕСКАЯ ПРОГРАММА. ВТОРАЯ СЕКЦИЯ РАЗЛИЧНА ДЛЯ КАЖДОГО ТИПА УСТРОЙСТВА И СОДЕРЖИТ ИНФОРМАЦИЮ, КОТОРАЯ ЗАВИСИТ ОТ УСТРОЙСТВА.

ПЕРВАЯ СЕКЦИЯ СОДЕРЖИТ СЛЕДУЮЩИЕ ПОЛЯ:

NRPW\_DEVICE - ЧЕТЫРЕХСЛОВНЫЙ ОПИСАТЕЛЬ СТРОКИ ИМЕНИ УСТРОЙСТВА (ПЕРВОЕ ДВОИНОЕ СЛОВО СОДЕРЖИТ ЧИСЛО СИМВОЛОВ В СТРОКЕ ИМЕНИ УСТРОЙСТВА NRPW\_DEVICE, ВТОРОЕ ДВОИНОЕ СЛОВО-АДРЕС ЭТОЙ СТРОКИ);

NRPW\_SIZE - РАЗМЕР Р-ТАБЛИЦЫ В БАЙТАХ (РАЗМЕР ОБЕИХ СЕКЦИИ Р-ТАБЛИЦЫ);

NRPW\_FLAGS - ФЛАГИ, ИСПОЛЬЗУЕМЫЕ ДИАГНОСТИЧЕСКОЙ СИСТЕМОЙ ПРИ ИНИЦИАЛИЗАЦИИ УСТРОЙСТВА. ИСПОЛЬЗУЮТСЯ 2 ФЛАГА:



13.00078-01 33 01

НРАМ\_ALLOC (БИТ 0) И НРАМ\_WASSAL (БИТ 1). ПЕРВЫЙ ФЛАГ, ЕСЛИ УСТАНОВЛЕН, УКАЗЫВАЕТ, ЧТО ПЕРЕД ТЕСТИРОВАНИЕМ В РЕЖИМЕ ПОЛЬЗОВАТЕЛЯ НУЖНО ЗАРЕЗЕРВИРОВАТЬ УСТРОЙСТВО ДЛЯ МОНОПОЛЬНОГО ПРИМЕНЕНИЯ. ВТОРОЙ ФЛАГ (НРАМ\_WASSAL) УСТАНОВЛИВАЕТСЯ СУПЕРВИЗОРОМ, ЕСЛИ УСТРОЙСТВО ЗАРЕЗЕРВИРОВАНО ДЛЯ МОНОПОЛЬНОГО ИСПОЛЬЗОВАНИЯ;

НРАВ\_DRIVE - НОМЕР УСТРОЙСТВА. ЭТО ЧИСЛО В КОНЦЕ ИМЕНИ УСТРОЙСТВА;

НРАТ\_DEVICE - ИМЯ УСТРОЙСТВА В КОДЕ КОИ-8 С НАЧАЛЬНЫМ СИМВОЛОМ "\_". НАПРИМЕР, "\_RNO";

НРАА\_DEVICE - САМЫЙ МЛАДШИЙ ВИРТУАЛЬНЫЙ АДРЕС РЕГИСТРОВ УСТРОЙСТВА В ПРОСТРАНСТВЕ P1 (ДЛЯ ОШ - ЭТО РЕГИСТР CSR, ДЛЯ МАГИСТРАЛИ SBI - РЕГИСТР КОНФИГУРАЦИИ И Т.Д.);

НРАА\_DVA - БАЗОВЫЙ АДРЕС ВИРТУАЛЬНОГО АДРЕСНОГО ПРОСТРАНСТВА, СВЯЗАННОГО С УСТРОЙСТВОМ. ДРУГИМ УСТРОЙСТВАМ, ПРИСОЕДИНЕННЫМ К ДАННОМУ, АДРЕСА УКАЗЫВАЮТСЯ ОТНОСИТЕЛЬНО ЭТОГО БАЗОВОГО АДРЕСА. КОГДА СОЗДАЕТСЯ НОВАЯ P-ТАБЛИЦА ДЛЯ УСТРОЙСТВА, ПРИСОЕДИНЕННОГО К ДАННОМУ, ЗНАЧЕНИЕ ЭТОГО ПОЛЯ ПРИСВАИВАЕТСЯ ПОЛЮ НРАА\_DEVICE ПРИСОЕДИНЕННОГО УСТРОЙСТВА. КОГДА АДРЕС НОВОГО УСТРОЙСТВА ПОСТУПАЕТ ОТ ПОЛЬЗОВАТЕЛЯ, ОН ДОЛЖЕН БЫТЬ ДОБАВЛЕН К АДРЕСУ В ПОЛЕ НРАА\_DEVICE. ПОЛЕ НРАА\_DVA МОЖЕТ СОДЕРЖАТЬ И ДРУГУЮ ИНФОРМАЦИЮ. НАПРИМЕР, ПРИ ИСПОЛЬЗОВАНИИ АДАПТЕРОВ ОШ КАЖДОЙ МАГИСТРАЛИ ОШ ПРИПИСЫВАЕТСЯ СВОЙ БАЗОВЫЙ АДРЕС. АДРЕС УСТРОЙСТВ, ПОДСОЕДИНЕННЫХ К ОТДЕЛЬНОЙ МАГИСТРАЛИ ОШ, СКЛАДЫВАЮТСЯ С БАЗОВЫМ АДРЕСОМ ОШ ДЛЯ ПОЛУЧЕНИЯ ДЕЙСТВИТЕЛЬНОГО ФИЗИЧЕСКОГО АДРЕСА. БАЗОВЫЙ АДРЕС ОШ ХРАНИТСЯ В ПОЛЕ НРАА\_DVA ДЛЯ P-ТАБЛИЦЫ АДАПТЕРА ОШ. КОГДА КОНТРОЛЛЕР ПОДСОЕДИНЯЕТСЯ К ОШ, ТО В ПОЛЕ НРАА\_DEVICE ЕГО P-ТАБЛИЦЫ БУДЕТ УСТАНОВЛЕНО ЗНАЧЕНИЕ, СОДЕРЖАЩЕЕСЯ В ПОЛЕ НРАА\_DVA P-ТАБЛИЦЫ АДАПТЕРА ОШ. А ПОЛЬ-

19.00078-01 33 01

ЗОВАТЕЛИ, СООТВЕТСТВЕННО, ПОДСКАЗЫВАЕТСЯ 18-РАЗРЯДНЫЙ АДРЕС КОНТРОЛЛЕРА. ЭТОТ АДРЕС МОЖЕТ ХРАНИТЬСЯ В 18 МЛАДШИХ РАЗРЯДАХ ПОЛЯ HPAA\_DEVICE, ДАВАЯ В РЕЗУЛЬТАТЕ ПОЛНЫЙ ФИЗИЧЕСКИЙ АДРЕС КОНТРОЛЛЕРА;

HPAA\_LINK - АДРЕС P-ТАБЛИЦЫ УСТРОЙСТВА, К КОТОРОМУ ПОДСОЕДИНЯЕТСЯ ДАННОЕ. ЕСЛИ ЭТО УСТРОЙСТВО ПОДСОЕДИНЯЕТСЯ К ПСЕВДОУСТРОЙСТВУ HUB, ТО ЗНАЧЕНИЕ ЭТОГО ПОЛЯ РАВНО 0;

HPAW\_VECTOR - АДРЕС ВЕКТОРА ПРЕРЫВАНИЯ УСТРОЙСТВА;

HPAT\_TYPE - СИМВОЛЬНАЯ СТРОКА ФИКСИРОВАННОЙ ДЛИНЫ, В КОТОРОЙ УКАЗАН ТИП УСТРОЙСТВА, НАПРИМЕР, DW730, RH730 ИЛИ RK611.

ПОЛЕ, СЛЕДУЮЩЕЕ ЗА ПЕРВОЙ СЕКЦИЕЙ, НАЗЫВАЕТСЯ HPAA\_DEPENDENT

ПОЛЯ HPAW\_SIZE, HPAA\_DEVICE, HPAB\_DRIVE, HPAT\_DEVICE, HPAA\_LINK И HPAT\_TYPE АВТОМАТИЧЕСКИ ЗАПОЛНЯЮТСЯ ДИАГНОСТИЧЕСКИМ СУПЕРВИЗОРОМ. ОСТАЛЬНЫЕ ПОЛЯ ЗАПОЛНЯЮТСЯ В СООТВЕТСТВИИ С УКАЗАНИЯМИ, СОДЕРЖАЩИМИСЯ В ОПИСАТЕЛЯХ P-ТАБЛИЦЫ.

ПОЛЯ HPAB\_FLAGS, HPAB\_DRIVE ЗАНИМАЮТ ПО ОДНОМУ БАЙТУ, ПОЛЯ HPAW\_SIZE, HPAW\_VECTOR - ПО ОДНОМУ СЛОВУ, HPAA\_DEVICE, HPAA\_DVA, HPAA\_LINK - ПО ДЛИННОМУ СЛОВУ, ПОЛЕ HPAA\_DEVICE - ДВОЙНОЕ СЛОВО, ПОЛЯ HPAT\_DEVICE, HPAT\_TYPE - ПО 12 БАЙТОВ.

### 7.7.3. ОБЩИЕ СВЕДЕНИЯ О МАКРОКОМАНДАХ

#### P-ТАБЛИЦЫ

В ГРУППУ ВХОДЯТ МАКРОКОМАНДЫ ОПИСАТЕЛИ P-ТАБЛИЦЫ И ОПРЕДЕЛЕНИЯ СТРУКТУРЫ, А ТАКЖЕ МАКРОКОМАНДЫ ADS\_DEVTYPE И ADS\_GPHARD.

ПРИ ПРОВЕРКЕ СПЕЦИАЛЬНЫХ УСТРОЙСТВ, КОТОРЫЕ НЕ ЯВЛЯЮТ-

19.00078-01 33 01

ся частью стандартной конфигурации SM-1700, СУПЕРВИЗОР НЕ ЗНАЕТ КАКИЕ ХАРАКТЕРИСТИКИ УСТРОЙСТВА ЕМУ НЕОБХОДИМО ЗАПРАШИВАТЬ ДЛЯ РАБОТЫ, Т.К. ОНИ МОГУТ БЫТЬ РАЗНЫМИ. ПОЭТОМУ ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ, КОТОРЫЕ ПРОВЕРЯЮТ СПЕЦИАЛЬНЫЕ УСТРОЙСТВА, ДОЛЖНЫ ВКЛЮЧАТЬ МАКРОКОМАНДЫ ОПИСАТЕЛЕЙ P-ТАБЛИЦЫ ДЛЯ КАЖДОГО ПРОВЕРЯЕМОГО УСТРОЙСТВА. С ПОМОЩЬЮ ЭТИХ МАКРОКОМАД ОПРЕДЕЛЯЮТСЯ ХАРАКТЕРИСТИКИ, ЗАВИСЯЩИЕ ОТ ТИПА УСТРОЙСТВА, ДЛЯ ЗАПИСИ ИХ В P-ТАБЛИЦУ. ПРИ ПОДГОТОВКЕ К ВЫПОЛНЕНИЮ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ, КОТОРАЯ ПРОВЕРЯЕТ СПЕЦИАЛЬНОЕ УСТРОЙСТВО, ОПЕРАТОР ВЫЧИСЛИТЕЛЬНОЙ МАШИНЫ ДОЛЖЕН ЗАГРУЗИТЬ УКАЗАННУЮ ПРОГРАММУ ДО ПОДКЛЮЧЕНИЯ ПРОВЕРЯЕМОГО УСТРОЙСТВА С ПОМОЩЬЮ КОМАНДЫ ATTACH. В ОТВЕТ НА ЭТУ КОМАНДУ СУПЕРВИЗОР ПРОСМАТРИВАЕТ ОПИСАТЕЛИ P-ТАБЛИЦЫ В ЗАГРУЖЕННОЙ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЕ И ЗАПРАШИВАЕТ У ОПЕРАТОРА ХАРАКТЕРИСТИКИ УСТРОЙСТВА В СООТВЕТСТВИИ С ИНФОРМАЦИЕЙ В ОПИСАТЕЛЕ P-ТАБЛИЦЫ.

КОГДА СУПЕРВИЗОР ОСУЩЕСТВЛЯЕТ ПОИСК ОПИСАТЕЛЯ P-ТАБЛИЦЫ, ОН СНАЧАЛА ПРОВЕРЯЕТ СПИСОК ОПИСАТЕЛЕЙ В ПРОГРАММЕ, ФОРМИРУЕМЫЙ МАКРОКОМАНДОЙ #DS\_DEVTYPE. ЕСЛИ ЭТОТ ПОИСК НЕУСПЕШЕН, ТО СУПЕРВИЗОР ПРОВЕРЯЕТ СВОЙ СОБСТВЕННЫЙ СПИСОК.

МАКРОКОМАНДА #DS\_BPARAM ВЫДАЕТ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЕ БАЗОВЫЙ АДРЕС P-ТАБЛИЦЫ ДЛЯ ЗАДАННОГО ЛОГИЧЕСКОГО НОМЕРА УСТРОЙСТВА. С ПОМОЩЬЮ БАЗОВОГО АДРЕСА И СИМВОЛИЧЕСКИХ ИМЕН СМЕЩЕНИЙ ПАРАМЕТРОВ P-ТАБЛИЦЫ (КОТОРЫЕ ВЫРАБАТЫВАЮТСЯ МАКРОКОМАНДОЙ #DS\_HPARAM), ДИАГНОСТИЧЕСКАЯ ПРОГРАММА ИМЕЕТ ВОЗМОЖНОСТЬ ОБРАЩАТЬСЯ К ЛЮБОЙ ОБЛАСТИ P-ТАБЛИЦЫ УСТРОЙСТВА.

#### 7.7.4. МАКРОКОМАНДЫ ОПИСАТЕЛЕЙ Р-ТАБЛИЦЫ

В ГРУППУ ВХОДЯТ МАКРОКОМАНДЫ:

ADS\_INITIALIZE, ADS\_DECIMAL, ADS\_OCTAL,  
ADS\_HEXADecimal, ADS\_STRING, ADS\_LITERAL, ADS\_FETCH,  
ADS\_STORE, ADS\_END.

ДЛЯ ОПИСАНИЯ Р-ТАБЛИЦЫ ДИАГНОСТИЧЕСКАЯ ПРОГРАММА ИСПОЛЬЗУЕТ ОПРЕДЕЛЕННУЮ КОМБИНАЦИЮ УКАЗАННЫХ МАКРОКОМАНД.

СУПЕРВИЗОР, ПРОСМАТРИВАЯ ОПИСАТЕЛИ Р-ТАБЛИЦЫ, ЗАПРАШИВАЕТ У ОПЕРАТОРА ВЕЛИЧИНЫ В СООТВЕТСТВИИ С ДАННЫМИ, СОДЕРЖАЩИМИСЯ В ОПИСАТЕЛЯХ, И ЗАПИСЫВАЕТ ПОЛУЧЕННЫЕ ВЕЛИЧИНЫ В ОБЛАСТИ Р-ТАБЛИЦЫ, УКАЗАННЫЕ В ОПИСАТЕЛЯХ.

МАКРОКОМАНДА ADS\_INITIALIZE ЗАДАЕТ ИМЯ УСТРОЙСТВА, ДЛИНУ Р-ТАБЛИЦЫ, МАКСИМАЛЬНЫЙ НОМЕР В ИМЕНИ УСТРОЙСТВА И ИМЯ ДРАЙВЕРА. ЭТА МАКРОКОМАНДА ДОЛЖНА БЫТЬ ПЕРВОЙ СРЕДИ ОПИСАТЕЛЕЙ Р-ТАБЛИЦЫ УСТРОЙСТВА.

ИНФОРМАЦИЯ, ЗАДАВАЕМАЯ МАКРОКОМАНДАМИ ADS\_DECIMAL, ADS\_OCTAL, ADS\_HEXADecimal, ADS\_STRING, ИСПОЛЬЗУЕТСЯ СУПЕРВИЗОРОМ ДЛЯ ЗАПРОСА У ОПЕРАТОРА СООТВЕТСТВЕННО ДЕСЯТИЧНОГО, ВОСЬМЕРИЧНОГО ИЛИ ШЕСТНАДЦАТЕРИЧНОГО ЧИСЛА ИЛИ СТРОКИ КОДА КОИ-8.

МАКРОКОМАНДА ADS\_LITERAL ПРИМЕНЯЕТСЯ ДЛЯ СОЗДАНИЯ КОНСТАНТЫ В СЛУЧАЕ, ЕСЛИ НЕ ТРЕБУЕТСЯ ПОЛУЧАТЬ ВЕЛИЧИНУ ОТ ОПЕРАТОРА.

МАКРОКОМАНДА ADS\_FETCH УКАЗЫВАЕТ СУПЕРВИЗОРУ НА НЕОБХОДИМОСТЬ ИЗВЛЕЧЕНИЯ ВЕЛИЧИНЫ ИЗ Р-ТАБЛИЦЫ.

МАКРОКОМАНДА ADS\_STORE УКАЗЫВАЕТ ОБЛАСТЬ Р-ТАБЛИЦЫ, В КОТОРУЮ ДОЛЖНА БЫТЬ ЗАПИСАНА ТЕКУЩАЯ ВЕЛИЧИНА, Т.Е. ТА ВЕЛИЧИНА, КОТОРАЯ БЫЛА ПОЛУЧЕНА С ПОМОЩЬЮ ПРЕДШЕСТВУЮЩЕЙ МАКРОКОМАНДЫ ОПИСАТЕЛЯ Р-ТАБЛИЦЫ.

13.00078-01 33 01'

МАКРОКОМАНДА #DS\_END ВЫРАБАТЫВАЕТ ДЛЯ СУПЕРВИЗОРА ПРИЗНАК КОНЦА ОПИСАТЕЛЯ Р-ТАБЛИЦЫ.

#### 7.7.4.1. #DS\_INITIALIZE - НАЧАЛО ОБРАБОТКИ ТАБЛИЦЫ Р

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_INITIALIZE DEVICE, LENGTH, MAX, DRIVER

ГДЕ DEVICE АППАРАТНОЕ ИМЯ УСТРОЙСТВА (НАПРИМЕР, RP06 ИЛИ DW780);

LENGTH - ОБЩАЯ ДЛИНА СООТВЕТСТВУЮЩЕЙ ТАБЛИЦЫ Р;

MAX - МАКСИМАЛЬНО ДОПУСТИМЫЙ НОМЕР УСТРОЙСТВА В ИМЕНИ УСТРОЙСТВА. ЕСЛИ У УСТРОЙСТВА НЕТ НОМЕРА, ТО АРГУМЕНТ MAX ДОЛЖЕН БЫТЬ РАВЕН 0;

DRIVER - ИМЯ ДРАЙВЕРА ОЧЕРЕДИ ВВОДА-ВЫВОДА УСТРОЙСТВА ИЗ ДВУХ СИМВОЛОВ. ЕСЛИ ДРАЙВЕР НЕ ПРИМЕНЯЕТСЯ, ТО ЭТОТ АРГУМЕНТ ДОЛЖЕН БЫТЬ НУЛЕМ.

МАКРОКОМАНДУ #DS\_INITIALIZE СЛЕДУЕТ УКАЗЫВАТЬ ПЕРВОЙ В МАССИВЕ ОПИСАТЕЛЕЙ ТАБЛИЦЫ Р. ОНА ЗАДАЕТ ИМЯ УСТРОЙСТВА, ОБЩУЮ ДЛИНУ ТАБЛИЦЫ Р, МАКСИМАЛЬНЫЙ НОМЕР УСТРОЙСТВА И ИМЯ ДРАЙВЕРА. ВСЕ ЭТИ ПАРАМЕТРЫ СУПЕРВИЗОР ЗАНОСИТ В ТАБЛИЦУ Р СООТВЕТСТВУЮЩЕГО УСТРОЙСТВА.

19.00078-01 33 01

7.7.4.2. ADS\_ADECIMAL - ЗАДАНИЕ ИНФОРМАЦИИ ДЛЯ  
ЗАПРОСА У ОПЕРАТОРА ДЕСЯТИЧНОЙ ВЕЛИЧИНЫ

ФОРМАТ МАКРОКОМАНДЫ:

ADS\_ADECIMAL PROMPT, LOW, HIGH

ГДЕ PROMPT - СТРОКА СИМВОЛОВ, ИСПОЛЬЗУЕМАЯ В КАЧЕСТВЕ  
ЗАПРОСА ОПЕРАТОРУ;

LOW - НИЖНЯЯ ГРАНИЦА ПАРАМЕТРА, ЗАДАВАЕМОГО ОПЕРАТО-  
РОМ;

HIGH - ВЕРХНЯЯ ГРАНИЦА ПАРАМЕТРА, ЗАДАВАЕМОГО ОПЕРАТО-  
РОМ.

ДАННУЮ МАКРОКОМАНДУ СЛЕДУЕТ ИСПОЛЬЗОВАТЬ ДЛЯ ЗАПРОСА  
ВВОДА ОПЕРАТОРОМ ДЕСЯТИЧНОЙ ВЕЛИЧИНЫ. ПОСЛЕ ТОГО КАК СУПЕР-  
ВИЗОР УСТАНОВИТ, ЧТО ПОЛУЧЕННАЯ ОТ ОПЕРАТОРА ВЕЛИЧИНА НАХО-  
ДИТСЯ В ЗАДАННЫХ ГРАНИЦАХ, ОНА СТАНОВИТСЯ ТЕКУЩЕЙ ВЕЛИЧИ-  
НОЙ.

7.7.4.3. ADS\_AOSTAL - ЗАДАНИЕ ИНФОРМАЦИИ ДЛЯ  
ЗАПРОСА У ОПЕРАТОРА ВОСЬМЕРИЧНОЙ  
ВЕЛИЧИНЫ

ФОРМАТ МАКРОКОМАНДЫ: .

ADS\_AOSTAL PROMPT, LOW, HIGH

ГДЕ PROMPT - СТРОКА СИМВОЛОВ, ИСПОЛЬЗУЕМАЯ В КАЧЕСТВЕ  
ЗАПРОСА ОПЕРАТОРУ;

LOW - НИЖНЯЯ ГРАНИЦА ПАРАМЕТРА, ЗАДАВАЕМОГО ОПЕРАТО-  
РОМ;

19.00078-01 33 01

HIGH - ВЕРХНЯЯ ГРАНИЦА ПАРАМЕТРА, ЗАДАВАЕМОГО ОПЕРАТОРОМ.

МАКРОКОМАНДУ СЛЕДУЕТ ИСПОЛЬЗОВАТЬ ДЛЯ ЗАПРОСА ВВОДА ОПЕРАТОРОМ ВОСЬМЕРИЧНОЙ ВЕЛИЧИНЫ. ПОСЛЕ ТОГО КАК СУПЕРВИЗОР УСТАНОВИТ, ЧТО ПОЛУЧЕННАЯ ОТ ОПЕРАТОРА ВЕЛИЧИНА НАХОДИТСЯ В ЗАДАННЫХ ГРАНИЦАХ, ОНА СТАНОВИТСЯ ТЕКУЩЕЙ ВЕЛИЧИНОЙ.

7.7.4.4. MDS\_HEXADecimal - ЗАДАНИЕ ИНФОРМАЦИИ  
ДЛЯ ЗАПРОСА У ОПЕРАТОРА  
ШЕСТНАДЦАТЕРИЧНОЙ ВЕЛИЧИНЫ

ФОРМАТ МАКРОКОМАНДЫ:

MDS\_HEXADecimal PROMPT, LOW, HIGH

ГДЕ PROMPT - СТРОКА СИМВОЛОВ, ИСПОЛЬЗУЕМАЯ СУПЕРВИЗОРОМ  
В КАЧЕСТВЕ ЗАПРОСА ОПЕРАТОРУ;

LOW - НИЖНЯЯ ГРАНИЦА ПАРАМЕТРА, ЗАДАВАЕМОГО ОПЕРАТОРОМ;

HIGH - ВЕРХНЯЯ ГРАНИЦА ПАРАМЕТРА, ЗАДАВАЕМОГО ОПЕРАТОРОМ.

МАКРОКОМАНДУ СЛЕДУЕТ ИСПОЛЬЗОВАТЬ ДЛЯ ЗАПРОСА ВВОДА ОПЕРАТОРОМ ШЕСТНАДЦАТЕРИЧНОЙ ВЕЛИЧИНЫ. ПОСЛЕ ТОГО КАК СУПЕРВИЗОР УСТАНОВИТ, ЧТО ПОЛУЧЕННАЯ ОТ ОПЕРАТОРА ВЕЛИЧИНА НАХОДИТСЯ В ЗАДАННЫХ ГРАНИЦАХ, ОНА СТАНОВИТСЯ ТЕКУЩЕЙ ВЕЛИЧИНОЙ.

19.00078-01 33 01

**7.7.4.5. ADS\_ASTRING - ЗАДАНИЕ ИНФОРМАЦИИ ДЛЯ  
ЗАПРОСА У ОПЕРАТОРА СТРОКИ СИМВОЛОВ**

**ФОРМАТ МАКРОКОМАНДЫ:**

**ADS\_ASTRING PROMPT, STRINGS**

**ГДЕ PROMPT - СТРОКА СИМВОЛОВ, ИСПОЛЬЗУЕМАЯ СУПЕРВИЗОРОМ  
В КАЧЕСТВЕ ЗАПРОСА ОПЕРАТОРУ;**

**STRINGS - СПИСОК ДОПУСТИМЫХ СТРОК, ЗАДАВАЕМЫХ ОПЕРАТО-  
РОМ.**

**МАКРОКОМАНДУ СЛЕДУЕТ ИСПОЛЬЗОВАТЬ ДЛЯ ЗАПРОСА ВВОДА  
ОПЕРАТОРОМ СТРОКИ СИМВОЛОВ. СУПЕРВИЗОР ПРОВЕРЯЕТ, ЯВЛЯЕТСЯ  
ЛИ ПОЛУЧЕННАЯ ОТ ОПЕРАТОРА СТРОКА СИМВОЛОВ ДОПУСТИМОЙ  
(Т.Е. ВХОДИТ ЛИ ЭТА СТРОКА В СПИСОК ДОПУСТИМЫХ СТРОК, ОПРЕ-  
ДЕЛЯЕМЫХ АРГУМЕНТОМ STRINGS).**

**7.7.4.6. ADS\_LITERAL - ОПРЕДЕЛЕНИЕ ЗНАЧЕНИЯ  
КОНСТАНТЫ**

**ФОРМАТ МАКРОКОМАНДЫ:**

**ADS\_LITERAL CONSTANT**

**ГДЕ CONSTANT - КОНСТАНТА.**

**МАКРОКОМАНДУ СЛЕДУЕТ ИСПОЛЬЗОВАТЬ ДЛЯ СОЗДАНИЯ КОНС-  
ТАНТЫ В КАЧЕСТВЕ ТЕКУЩЕЙ ВЕЛИЧИНЫ, ЕСЛИ ЕЕ НЕ ТРЕБУЕТСЯ  
ПОЛУЧАТЬ ОТ ОПЕРАТОРА.**



13.00078-01 33 01

#### 7.7.4.7. #DS\_FETCH - ИЗВЛЕЧЕНИЕ ПОЛЯ ИЗ ТАБЛИЦЫ

P

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_FETCH OFFSET, BIT, SIZE

ГДЕ OFFSET - СМЕЩЕНИЕ В БАЙТАХ ОТНОСИТЕЛЬНО БАЗЫ  
ТАБЛИЦЫ P. СМЕЩЕНИЕ МОЖЕТ БЫТЬ ОТ 0 ДО 65535;

BIT - СМЕЩЕНИЕ В БИТАХ ОТНОСИТЕЛЬНО НАЧАЛА ПАРАМЕТРА  
OFFSET. СМЕЩЕНИЕ МОЖЕТ БЫТЬ ОТ 0 ДО 255;

SIZE - РАЗМЕР ПОЛЯ (В БИТАХ), КОТОРОЕ НУЖНО ИЗВЛЕЧЬ ИЗ  
ТАБЛИЦЫ P. РАЗМЕР ПОЛЯ МОЖЕТ БЫТЬ ОТ 1 ДО 32.

МАКРОКОМАНДУ СЛЕДУЕТ ИСПОЛЬЗОВАТЬ ДЛЯ ФОРМИРОВАНИЯ  
ТЕКУЩЕЙ ВЕЛИЧИНЫ ПУТЕМ ИЗВЛЕЧЕНИЯ ПОЛЯ ИЗ ТАБЛИЦЫ P. ПАРА-  
МЕТРЫ OFFSET, BIT И SIZE УКАЗЫВАЮТ ПОЛЕ В ТАБЛИЦЕ P.

#### 7.7.4.8. #DS\_STORE - ВНЕСЕНИЕ ВЕЛИЧИНЫ В ТАБЛИЦУ P

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_STORE OFFSET, BIT, SIZE

ГДЕ OFFSET - СМЕЩЕНИЕ В БАЙТАХ ОТНОСИТЕЛЬНО БАЗЫ  
ТАБЛИЦЫ P. СМЕЩЕНИЕ МОЖЕТ БЫТЬ ОТ 0 ДО 65536;

BIT - СМЕЩЕНИЕ В БИТАХ ОТНОСИТЕЛЬНО НАЧАЛА ПАРАМЕТРА  
OFFSET. СМЕЩЕНИЕ МОЖЕТ БЫТЬ ОТ 0 ДО 255;

SIZE - РАЗМЕР ПОЛЯ (В БИТАХ), КОТОРОЕ НУЖНО ВНЕСТИ В  
ТАБЛИЦУ P. РАЗМЕР ПОЛЯ МОЖЕТ БЫТЬ ОТ 1 ДО 32.

1Э.00078-01 33 01

МАКРОКОМАНДУ СЛЕДУЕТ ИСПОЛЬЗОВАТЬ ДЛЯ ЗАНЕСЕНИЯ В ТАБЛИЦУ Р ТЕКУЩЕЙ ВЕЛИЧИНЫ, Т.Е. ТРИ ВЕЛИЧИНЫ, КОТОРАЯ БЫЛА ПОЛУЧЕНА С ПОМОЩЬЮ ПРЕДШЕСТВУЮЩЕЙ МАКРОКОМАНДЫ.

#### 7.7.4.9. #DS\_END - ОКОНЧАНИЕ ОБРАБОТКИ ТАБЛИЦЫ

Р

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_END

АРГУМЕНТОВ У МАКРОКОМАНДЫ ЧЕТ.

МАКРОКОМАНДУ СЛЕДУЕТ ИСПОЛЬЗОВАТЬ В КОНЦЕ МАССИВА ОПИСАТЕЛЕЙ ТАБЛИЦЫ Р УСТРОЙСТВА. ОНА ЯВЛЯЕТСЯ ДЛЯ СУПЕРВИЗОРА ПРИЗНАКОМ КОНЦА МАССИВА ОПИСАТЕЛЕЙ.

#### 7.7.5. #DS\_DEVTYPE - УКАЗАНИЕ ТИПОВ УСТРОЙСТВ И АДРЕСОВ ОПИСАТЕЛЕЙ Р-ТАБЛИЦЫ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_DEVTYPE {STRINGS}[,ADDRESSES]

ГДЕ STRINGS - ИМЕНА ТИПОВ УСТРОЙСТВ. ИМЕНА ДОЛЖНЫ

БЫТЬ ЗАКЛЮЧЕНЫ В УГЛОВЫЕ СКОБКИ < > И РАЗДЕЛЯТЬСЯ ЗАПЯТЫМИ;

ADDRESSES - АДРЕСА МАССИВОВ ОПИСАТЕЛЕЙ ТАБЛИЦЫ Р, УКАЗАННЫХ В ПРОГРАММЕ. АДРЕСА МАССИВОВ ДОЛЖНЫ ЗАКЛЮЧАТЬСЯ В УГЛОВЫЕ СКОБКИ < > И РАЗДЕЛЯТЬСЯ ЗАПЯТЫМИ.

МАКРОКОМАНДА #DS\_DEVTYPE ГЕНЕРИРУЕТ СТРОКИ ИМЕН ТИПОВ УСТРОЙСТВ И АДРЕСА МАССИВОВ ОПИСАТЕЛЕЙ ТАБЛИЦЫ Р, ЗАДАННЫХ В ПРОГРАММЕ. УКАЗАННЫЕ МАССИВЫ ИСПОЛЬЗУЮТСЯ СУПЕРВИЗОРОМ ДЛЯ ФОРМИРОВАНИЯ ТАБЛИЦЫ Р СООТВЕТСТВУЮЩИХ УСТРОЙСТВ.

13.00078-01 33 01

### 7.7.6. МАКРОКОМАНДЫ ОПРЕДЕЛЕНИЯ СТРУКТУРЫ

В ЭТУ ГРУППУ ВХОДЯТ МАКРОКОМАНДЫ #DEFINI, #DEF И #DEFEND, КОТОРЫЕ СЛЕДУЕТ ИСПОЛЬЗОВАТЬ ДЛЯ ОПРЕДЕЛЕНИЯ СТРУКТУРЫ ДОБАВЛЕНИЙ К ТАБЛИЦЕ P, КОТОРЫЕ НЕОБХОДИМО СОЗДАТЬ ПОЛЬЗОВАТЕЛЮ. УКАЗАННЫЕ МАКРОКОМАНДЫ НАХОДЯТСЯ В БИБЛИОТЕКЕ LIB.MLB ОПЕРАЦИОННОЙ СИСТЕМЫ МОС ВП.

#### 7.7.6.1. #DEFINI - НАЧАЛО СТРУКТУРЫ ДАННЫХ

ФОРМАТ МАКРОКОМАНДЫ:

#DEFINI STRUC, GBL, DOT

ГДЕ STRUC - ИМЯ СТРУКТУРЫ (НАПРИМЕР, ИМЯ ТАБЛИЦЫ P);

GBL - GLOBAL (ГЛОБАЛЬНЫЕ) ИЛИ LOCAL (ЛОКАЛЬНЫЕ). ПО УМОЛЧАНИЮ ГЕНЕРИРУЮТСЯ ЛОКАЛЬНЫЕ СИМВОЛИЧЕСКИЕ ИМЕНА.

МАКРОКОМАНДУ СЛЕДУЕТ ИСПОЛЬЗОВАТЬ ДЛЯ НАЧАЛА ОПРЕДЕЛЕНИЯ СТРУКТУРЫ ДАННЫХ.

#### 7.7.6.2. #DEF - ОПРЕДЕЛЕНИЕ ПОЛЕЙ В СТРУКТУРЕ

ФОРМАТ МАКРОКОМАНДЫ:

#DEF SYM, ALLOC, SIZ

ГДЕ SYM - ОПРЕДЕЛЯЕМОЕ СИМВОЛИЧЕСКОЕ ИМЯ;

ALLOC - ДИРЕКТИВА АССЕМБЛЕРА, ОПРЕДЕЛЯЮЩАЯ ИСПОЛЬЗУЕМЫЙ РАЗМЕР БЛОКА:

.BLKB

.BLKW

.BLKL

SIZ - ЧИСЛО БЛОКОВ.

МАКРОКОМАНДА ИСПОЛЬЗУЕТСЯ ДЛЯ ГЕНЕРАЦИИ СМЕЩЕНИЯ ДАННОГО СИМВОЛИЧЕСКОГО ИМЕНИ.

### 7.7.5.5. #DEFEND - ОКОНЧАНИЕ ОПРЕДЕЛЕНИЯ

ФОРМАТ МАКРОКОМАНД:

```
#DEFEND STRUC, GBL
```

ГДЕ STRUC - ИМЯ СТРУКТУРЫ;

GBL - GLOBAL (ГЛОБАЛЬНЫЕ) ИЛИ LOCAL (ЛОКАЛЬНЫЕ).

МАКРОКОМАНДУ СЛЕДУЕТ ИСПОЛЬЗОВАТЬ ДЛЯ ОЧИСТКИ ПРОЦЕССА ОПРЕДЕЛЕНИЯ СТРУКТУРЫ ДАННЫХ ПОСЛЕ ОПРЕДЕЛЕНИЯ ВСЕХ СИМВОЛИЧЕСКИХ ИМЕН.

### 7.7.7. ПРИМЕРЫ МАКРОКОМАНД ОПРЕДЕЛЕНИЯ СТРУКТУРЫ ДАННЫХ И ОПИСАТЕЛЕЙ P-ТАБЛИЦЫ

РАССМОТРИМ ПРИМЕР ИСПОЛЬЗОВАНИЯ МАКРОКОМАНД СТРУКТУРЫ ДАННЫХ ДЛЯ ОПРЕДЕЛЕНИЯ ЗАВИСЯЩИХ ОТ УСТРОЙСТВА СМЕЩЕНИИ ТАБЛИЦЫ ДЛЯ ИНТЕРФЕЙСА RH780.

#### ПРИМЕР 1.

```
#DEFINI    RH780, GLOBAL, HPPA_DEPENDENT
#DEF      RH780#B_TR, .BLKB, 1           ; НОМЕР TR АДАПТЕРА
#DEF      RH780#B_BR, .BLKB, 1           ; УРОВЕНЬ BR АДАПТЕРА
#DEF      RH780#K_LEN
#DEFEND    RH780, GLOBAL
```

В ПРИМЕРЕ 1 ПРОГРАММИСТ СОЗДАЛ СТРУКТУРУ ДАННЫХ С ИМЕНЕМ RH780, КОТОРАЯ ОПРЕДЕЛЯЕТ СИМВОЛИЧЕСКИЕ СМЕЩЕНИЯ P-ТАБЛИЦЫ, ОПИСЫВАЮЩИЕ ИНТЕРФЕЙС RH780. СИМВОЛИЧЕСКИЕ ИМЕНА ЯВЛЯЮТСЯ ГЛОБАЛЬНЫМИ, А ВЕЛИЧИНОЙ ПЕРВОГО ГЕНЕРИРУЕМОГО СИМВОЛИЧЕСКОГО ИМЕНИ HPPA\_DEPENDENT. ДЛЯ СИМВОЛИЧЕСКИХ ИМЕН

13.00078-01 33 01

RH7804B\_TR И RH7804B\_BR ВЫДЕЛЕНО ПО ОДНОМУ БАЙТУ ПАМЯТИ. СИМВОЛИЧЕСКОЕ ИМЯ RH7804K\_LEN УКАЗЫВАЕТ ВЕЛИЧИНУ, СОДЕРЖАЩУЮ ДЛИНУ ТАБЛИЦЫ P.

РАССМОТРИМ ПРИМЕР 2, В КОТОРОМ СУПЕРВИЗОР ИСПОЛЬЗУЕТ МАКРОКОМАНДЫ ОПИСАТЕЛЕЙ ТАБЛИЦЫ P ДЛЯ ОБЕСПЕЧЕНИЯ ЗАПОЛНЕНИЯ ТАБЛИЦЫ P ВО ВРЕМЯ ПРОГОНА.

ПРИМЕР 2.

```
1) ADS_INITIALIZE RH780,RH7804K_LEN,8 ; НАЧАЛО ОПИСАТЕЛЯ
; P-ТАБЛИЦЫ
2) ADS_DECIMAL TR,1,15 ; ЗАПРОС НОМЕРА TR
3) ADS_STORE RH7804B_TR,0,8 ; ЗАПИСЬ TR В
; RH7804B_TR
4) ADS_STORE HPA_DEVICE,13,4 ; ЗАПИСЬ TR В БИТАХ
; 13-15 АДРЕСА
; УСТРОЙСТВА
5) ADS_STORE HPAW_VECTOR,2,4 ; ЗАПИСЬ TR В БИТАХ
; 2-5 ВЕКТОРА
6) ADS_DECIMAL BR,4,7 ; ЗАПРОС BR
7) ADS_STORE RH7804B_BR,0,8 ; ЗАПИСЬ BR В
; RH7804B_BR
8) ADS_STORE HPAW_VECTOR,6,2 ; ЗАПИСЬ BR В БИТАХ
; 6-7 ВЕКТОРА
9) ADS_LITERAL 6 ; ПОЛУЧЕНИЕ
; ЛИТЕРАЛА 6
10) ADS_STORE HPA_DEVICE,28,4 ; УСТАНОВКА БИТОВ
; 28-31 АДРЕСА
; УСТРОЙСТВА
11) ADS_LITERAL 1 ; ПОЛУЧЕНИЕ
; ЛИТЕРАЛА 1
12) ADS_STORE HPAW_VECTOR,8,1 ; УСТАНОВКА БИТА 8
; В СМЕЩЕНИИ
; ВЕКТОРА SCB
13) ADS_END ; КОНЕЦ ОПИСАТЕЛЯ
; ТАБЛИЦЫ P
```

В ПРИМЕРЕ 2 ИЗ СУПЕРВИЗОРА СТРОКА 1 ОБЕСПЕЧИВАЕТ ИНИЦИАЛИЗАЦИЮ ОПИСАТЕЛЯ ТАБЛИЦЫ P ДЛЯ ИНТЕРФЕЙСА RH780. ОПЕРАТОРОМ МОЖЕТ БЫТЬ ВЫБРАНО ДО 3 УСТРОЙСТВ. МАКРОКОМАНДА ADS\_DECIMAL В СТРОКЕ 2 БУДЕТ ВЫЗЫВАТЬ ЗАПРОС ВВОДА ОПЕРАТОРОМ НОМЕРА TR. СУПЕРВИЗОР ПРИНИМАЕТ ЧИСЛО В ДИАПАЗОНЕ 1-15.

МАКРОКОМАНДА ADS\_STORE В СТРОКЕ 3 ПРИВОДИТ К ТОМУ, ЧТО СУПЕРВИЗОР ЗАПИСЫВАЕТ ВЕЛИЧИНУ, ПОЛУЧЕННУЮ ОТ ОПЕРАТОРА (НОМЕР TR) В ЯЧЕЙКЕ, ОПРЕДЕЛЕННОЙ СИМВОЛИЧЕСКИМ ИМЕНЕМ СМЕ-

13.00078-01 33 01

ЦЕННЯ RH730PB\_TR. БИТОВОЕ СМЕЩЕНИЕ ОТ НАЧАЛА ЭТОЙ ЯЧЕЙКИ РАВНО 0, А РАЗМЕР ПОЛЯ СОСТАВЛЯЕТ 3 БИТОВ. СТРОКА 4 ОБЕСПЕЧИВАЕТ ЗАПИСЬ ЭТОЙ ЖЕ ВЕЛИЧИНЫ TR В БИТАХ 13-16 АДРЕСА УСТРОЙСТВА, А СТРОКА 5 ЗАПИСЫВАЕТ ВЕЛИЧИНУ TR В БИТАХ 2-5 ВЕКТОРА УСТРОЙСТВА.

СТРОКИ 6-8 ОБЕСПЕЧИВАЮТ ЗАПРОС И ЗАПИСЬ НОМЕРА BR ТОЧНО ТАКИМ ЖЕ ОБРАЗОМ. СТРОКИ 9 И 10 ИСПОЛЬЗУЮТ ЛИТЕРАЛ 6 ДЛЯ УСТАНОВКИ БИТОВ 29 И 30 АДРЕСА УСТРОЙСТВА. СТРОКИ 11 И 12 ВЫПОЛНЯЮТ УСТАНОВКУ БИТА 8 В ВЕКТОРЕ, А МАКРОКОМАНДА ADS\_END ЗАКАНЧИВАЕТ ОПИСАТЕЛЬ P-ТАБЛИЦЫ.

#### 7.7.8. ADS\_GPHARD\_X - ПОЛУЧЕНИЕ БАЗОВОГО АДРЕСА ТАБЛИЦЫ P

ФОРМАТ МАКРОКОМАНДЫ:

ADS\_GPHARD DEVNUM, ADRLOC

ГДЕ DEVNUM - ЛОГИЧЕСКИЙ НОМЕР УСТРОЙСТВА;

ADRLOC - АДРЕС ДЛИННОГО СЛОВА ДЛЯ ПРИЕМА БАЗОВОГО АДРЕСА ТАБЛИЦЫ P.

ПРИ ИСПОЛЬЗОВАНИИ МАКРОКОМАНДЫ ВЫДАЮТСЯ СЛЕДУЮЩИЕ КОДЫ СОСТОЯНИЯ:

DSR\_NORMAL - НОРМАЛЬНОЕ ВЫПОЛНЕНИЕ

DSR\_ERROR - СПИСОК АРГУМЕНТОВ НЕ СОДЕРЖИТ ТОЧНО ДВА ЭЛЕМЕНТА ИЛИ УКАЗАНЫ ЛОГИЧЕСКИЙ НОМЕР УСТРОЙСТВА СЛИШКОМ ВЕЛИК.

МАКРОКОМАНДА ADS\_GPHARD ВЫЗЫВАЕТ ПОДПРОГРАММУ СУПЕРВИЗОРА ДЛЯ ПОЛУЧЕНИЯ АДРЕСА ВХОДА В ТАБЛИЦУ P, СВЯЗАННУЮ С ЛОГИЧЕСКИМ НОМЕРОМ ДАННОГО УСТРОЙСТВА.

13.00078-01 33 01'

ДАННАЯ МАКРОКОМАНДА ИСПОЛЬЗУЕТСЯ ПРИ НЕОБХОДИМОСТИ  
ОПРОСА СОДЕРЖИМОГО Р-ТАБЛИЦЫ.

РАССМОТРИМ ПРИМЕР ИСПОЛЬЗОВАНИЯ МАКРОКОМАНДЫ  
#DS\_GPHARD\_X В ПОДПРОГРАММЕ ИНИЦИАЛИЗАЦИИ:

ПРИМЕР.

```
#DS_BGNINIT
#DS_BPASSD 10#
#DS_ENDPASS_G
10# CLRL LOG_UNIT#L ; ЛОГИЧЕСКОЕ УСТРОЙСТВО 0
    CLRW CSR#W ; ОЧИСТКА БУФЕРА CSR
#DS_GPHARD_S- ; ПОЛУЧЕНИЕ АДРЕСА
    ; Р-ТАБЛИЦЫ
    DEVNUM=LOG_UNIT#L,- ; ЛОГИЧЕСКОЕ УСТРОЙСТВО
    ADLOG=HDW_PTR#LE#A ; БУФЕР ДЛЯ ПРИЕМА БАЗОВОГО
    ; АДРЕСА Р-ТАБЛИЦЫ
#DS_BERROR
MOVL HDW_PTR#LE#A,R2 ; АДРЕС Р-ТАБЛИЦЫ ЗАСЛАТЬ В
    ; R2
MOVL HDW_PTR#LE#A,DZ#CSR ; АДРЕС CSR ЗАСЛАТЬ В
    ; ЯЧЕЙКУ DZ#CSR
```

### 7.8. МАКРОКОМАНДЫ ДИАЛОГА МЕЖДУ ПРОГРАММОЙ И ОПЕРАТОРОМ

В РЯДЕ СЛУЧАЕВ ДЛЯ ВЫПОЛНЕНИЯ ТЕСТОВ ТРЕБУЕТСЯ ПОЛУЧЕ-  
НИЕ ИНФОРМАЦИИ ОТ ОПЕРАТОРА. ТАКИЕ ТЕСТЫ ДОЛЖНЫ ПОМЕЩАТЬСЯ  
В ТЕСТОВУЮ СЕКЦИЮ РУЧНОГО ВМЕШАТЕЛЬСТВА (MANUAL).

ДЛЯ ОРГАНИЗАЦИИ ДИАЛОГА МЕЖДУ ПРОГРАММОЙ И ОПЕРАТОРОМ  
В ДИАГНОСТИЧЕСКОЙ БИБЛИОТЕКЕ ИМЕЮТСЯ СЛЕДУЮЩИЕ МАКРОКОМАН-  
ДЫ:

```
#DS_ASKSTR_X, #DS_ASKDATA_X, #DS_ASKVLD_X, #DS_ASKLGCL_X,  
#DS_ASKADR_X, #DS_PARSE_X, #DS_CLI.
```

С ПОМОЩЬЮ ПЕРВЫХ ПЯТИ МАКРОКОМАНД МОЖНО ЗАПРОСИТЬ У  
ОПЕРАТОРА СООТВЕТСТВЕННО СТРОКУ КОДА КОИ-8, ЧИСЛОВУЮ ВЕЛИ-  
ЧИНУ, ЧИСЛОВУЮ ВЕЛИЧИНУ В КОДЕ КОИ-8, ЛОГИЧЕСКИЙ ОТВЕТ (ДА,  
НЕТ) И АДРЕС.

МАКРОКОМАНДА #DS\_PARSE ИСПОЛЬЗУЕТСЯ ДЛЯ АНАЛИЗА СТРОКИ  
КОМАНДЫ С ПОМОЩЬЮ СИНТАКСИЧЕСКОГО ДЕРЕВА. В КАЖДОМ УЗЛЕ

13.00074-01 33 01

ЭТОГО ДЕРЕВА, СОЗДАННОМ С ПОМОЩЬЮ МАКРОКОМАНД #DS\_CLI, УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ ИНСТРУКЦИИ CASE, КОТОРАЯ В СВОЮ ОЧЕРЕДЬ ПЕРЕДАЕТ УПРАВЛЕНИЕ ПОДПРОГРАММЕ, ВЫПОЛНЯЮЩЕЙ СООТВЕТСТВУЮЩУЮ КОМАНДУ ОПЕРАТОРА. НА РИС.9 (СМ. ПРИЛОЖЕНИЕ 3) ИЗОБРАЖЕНА БЛОК-СХЕМА ЧАСТИ ПРОГРАММЫ, ПРЕДСТАВЛЯЮЩЕЙ ТИПИЧНЫЙ ДИАЛОГ С ОПЕРАТОРОМ.

ЕСЛИ ОПЕРАТОР ВВОДИТ НЕДОПУСТИМУЮ КОМАНДУ, ТО У ОПЕРАТОРА ЗАПРАШИВАЕТСЯ ВВОД ДРУГОЙ КОМАНДЫ. ЕСЛИ ОШИБКИ НЕТ, ТО КОМАНДА ПОЛНОСТЬЮ ОБРАБАТЫВАЕТСЯ, А ЗАТЕМ ВЫПОЛНЯЕТСЯ.

#### 7.8.1. #DS\_ASKSTR\_X - ЗАПРОС У ОПЕРАТОРА СТРОКИ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_ASKSTR\_X MSGADR, BUFADR, MAXLEN, VALTAB, DEFADR

ГДЕ MSGADR - АДРЕС СТРОКИ КОДА КОИ-8 (ДИРЕКТИВА .ASCII), ИСПОЛЬЗУЕМОЙ В КАЧЕСТВЕ ЗАПРОСА ОПЕРАТОРУ;

BUFADR - АДРЕС БУФЕРА ДЛЯ ПОЛУЧЕНИЯ ОТ ОПЕРАТОРА СТРОКИ (ОТ ОПЕРАТОРА ПРИХОДИТ СТРОКА С НАЧАЛЬНЫМ БАЙТОМ ДЛИНЫ);

MAXLEN - МАКСИМАЛЬНАЯ ДЛИНА СТРОКИ ОТВЕТА (БЕЗ ВКЛЮЧЕНИЯ БАЙТА ДЛИНЫ). ПО УМОЛЧАНИЮ - 128

VALTAB - АДРЕС СПИСКА УКАЗАТЕЛЕЙ ДОПУСТИМЫХ СТРОК ОТВЕТА, В ПЕРВОМ ДЛИННОМ СЛОВЕ - ЧИСЛО УКАЗАТЕЛЕЙ. ПО УМОЛЧАНИЮ 0, ЧТО ОЗНАЧАЕТ ОТСУТСТВИЕ ТАБЛИЦЫ ДЛЯ ПРОВЕРКИ ОТВЕТОВ ОПЕРАТОРА;

DEFADR - АДРЕС СТРОКИ (С НАЧАЛЬНЫМ БАЙТОМ ДЛИНЫ), КОТОРАЯ ИСПОЛЬЗУЕТСЯ В СЛУЧАЕ, ЕСЛИ ОПЕРАТОР ВЫДАЕТ



13.00078-01 33 01

НУЛЕВОЙ ОТВЕТ. ПО УМОЛЧАНИЮ - 0, ЧТО ОЗНАЧАЕТ ОТСУТСТВИЕ СТРОКИ ПО УМОЛЧАНИЮ.

МАКРОКОМАНДА #DS\_ASKSTR\_X ВОЗВРАЩАЕТ КОДЫ СОСТОЯНИЯ:

DS#\_NORMAL - НОРМАЛЬНОЕ ОКОНЧАНИЕ РАБОТЫ,

DS#\_PROGERR - НЕПРАВИЛЬНОЕ ЧИСЛО УКАЗАННЫХ АРГУМЕНТОВ,

DS#\_TRUNCATE - СТРОКА, СОЗДАННАЯ ОПЕРАТОРОМ, УСЕЧЕНА,

Т.К. ПЕРЕПОЛНЯЕТСЯ ОТВЕДЕННЫЙ БУФЕР.

МАКРОКОМАНДА #DS\_ASKSTR\_X ВЫЗЫВАЕТ СЛУЖЕБНУЮ ПОДПРОГРАММУ СУПЕРВИЗОРА, КОТОРАЯ ЗАПРАШИВАЕТ У ОПЕРАТОРА СТРОКУ ОТВЕТА. ЕСЛИ СТРОКА, ЗАДАННАЯ ОПЕРАТОРОМ, ДОПУСТИМА, ТО ОНА ПОМЕЩАЕТСЯ В ОТВЕДЕННЫЙ ЕЙ БУФЕР.

МАКРОКОМАНДА #DS\_ASKSTR\_X НЕ ВЫПОЛНЯЕТСЯ, ЕСЛИ НЕ УСТАНОВЛЕН ФЛАГ ОПЕРАТОРА.

ЕСЛИ УСТАНОВЛЕН ФЛАГ СООБЩЕНИЯ (PROMPT), ТО ВМЕСТЕ С ЗАПРОСОМ НА ТЕРМИНАЛЕ БУДУТ ПЕЧАТАТЬСЯ ДОПУСТИМЫЕ ЗНАЧЕНИЯ И ЗНАЧЕНИЯ ПО УМОЛЧАНИЮ.

### 7.8.2. #DS\_ASKDATA\_X - ЗАПРОС У ОПЕРАТОРА ЧИСЛОВОЙ ВЕЛИЧИНЫ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_ASKDATA\_X MSGADR, DATADR[, RADIX][, MASK][, LCLIM][, HILIM][, DEFALT][, RESERV][, EXWORD]

ГДЕ MSGADR - АДРЕС СТРОКИ КОДА КОИ-8 (ДИРЕКТИВА .ASCII), ИСПОЛЬЗУЕМОЙ В КАЧЕСТВЕ ЗАПРОСА ОПЕРАТОРУ;

DATADR - АДРЕС ДЛИННОГО СЛОВА, В КОТОРОЕ ПОСТУПАЕТ ОТВЕТ ОТ ОПЕРАТОРА;

RADIX - ОСНОВАНИЕ (СИСТЕМА СЧИСЛЕНИЯ), В КОТОРОМ ДОЛ-

19.00078-01 33 01

Может быть выдан ответ оператором: PARAM\_BIN (двоичное),  
PARAM\_OCT (восьмеричное), PARAM\_HEX (шестнадцатеричное)  
или PARAM\_DEC (десятичное). По умолчанию - PARAM\_DEC;

MASK - битовая маска, указывающая позицию и размер  
поля, в которое засылается числовой ответ;

LOLIM - минимальный допустимый числовой ответ. По  
умолчанию -2\*\*31;

HILIM - максимальный допустимый числовой ответ. По  
умолчанию +2\*\*31-1;

DEFAULT - значение, используемое в случае, если опера-  
тор вводит ноль. По умолчанию - 0;

RESERV - не используется;

EXWORD - маска исключения. ПАРАМЕТР PARAM\_NODEF ОЗНА-  
ЧАЕТ ОТСУТСТВИЕ УМОЛЧАНИЯ. PARAM\_ATDEF, PARAM\_ATLO И  
PARAM\_ATHI УКАЗЫВАЮТ, ЧТО АРГУМЕНТЫ DEFAULT, LOLIM И  
HILIM УКАЗЫВАЮТ АДРЕСА (А НЕ ЛИТЕРАЛЬНЫЕ ЗНАЧЕНИЯ), ПО  
КОТОРЫМ НАХОДИТСЯ СООТВЕТСТВУЮЩИЕ ВЕЛИЧИНЫ.

МАКРОКОМАНДА #DS\_ASKDATA\_X ВОЗВРАЩАЕТ СЛЕДУЮЩИЕ КОДЫ  
СОСТОЯНИЯ:

DSF\_NORMAL - НОРМАЛЬНОЕ ОКОНЧАНИЕ,

DSF\_PROGERR - ОПЕРАТОР ЗАДАЛ НЕПРАВИЛЬНОЕ ЗНАЧЕНИЕ,

DSF\_TRUNCATE - ВЕЛИЧИНА, ЗАДАННАЯ ОПЕРАТОРОМ, СЛИШКОМ  
ВЕЛИКА И НЕ МОЖЕТ РАЗМЕСТИТЬСЯ В УКАЗАННОМ БУФЕРЕ.

МАКРОКОМАНДА #DS\_ASKDATA\_X ВЫЗЫВАЕТ СЛУЖЕБНУЮ ПОДПРОГ-  
РАММУ СУПЕРВИЗОРА, КОТОРАЯ ЗАПРАШИВАЕТ У ОПЕРАТОРА (НА ТЕР-  
МИНАЛЕ) ВВОД ЧИСЛОВОЙ ВЕЛИЧИНЫ. ПОДПРОГРАММА ПРОВЕРЯЕТ

13.00078-01 33 01

ЛЕЖИТ ЛИ ПОЛУЧЕННАЯ ОТ ОПЕРАТОРА ВЕЛИЧИНА В ПРЕДЕЛАХ ДОПУСТИМОГО ДИАПАЗОНА. ЕСЛИ ДА, ТО ПОЛУЧЕННОЕ ОТ ОПЕРАТОРА ЧИСЛО ЗАПИСЫВАЕТСЯ В ПОЛЕ, УКАЗАННОЕ В МАКРОКОМАНДЕ АРГУМЕНТАМИ DATADR И MASK.

МАКРОКОМАНДА #DS\_ASKDATA\_X НЕ ВЫПОЛНЯЕТСЯ, ЕСЛИ ОЧИЩЕН ФЛАГ ОПЕРАТОРА.

ЕСЛИ УСТАНОВЛЕН ФЛАГ СООБЩЕНИЯ (PROMPT), ТО ВМЕСТЕ С ЗАПРОСОМ НА ТЕРМИНАЛЕ БУДУТ ПЕЧАТАТЬСЯ ГРАНИЧНЫЕ ЗНАЧЕНИЯ И УСТАНАВЛИВАЕМЫЕ ПО УМОЛЧАНИЮ ВЕЛИЧИНЫ.

ЕСЛИ ЧИСЛОВОЙ ОТВЕТ ОПЕРАТОРА ПРЕВЫШАЕТ РАЗМЕР, УКАЗАННЫЙ В МАСКЕ, ПРОИСХОДИТ УСЕЧЕНИЕ ЛЕВЫХ БИТОВ.

АРГУМЕНТЫ МАКРОКОМАНДЫ RADIX И EXWORD ОПРЕДЕЛЯЮТСЯ МАКРОКОМАНДОЙ #DS\_PARDEF.

МАКРОКОМАНДА #DS\_ASKDATA\_X ВОЗВРАЩАЕТ В РЕГИСТРЕ R1 НЕУСЕЧЕННУЮ ДВОИЧНУЮ ВЕЛИЧИНУ.

7.8.3. #DS\_ASKVLD\_X - ЗАПРОС У ОПЕРАТОРА ЧИСЛОВОГО ЗНАЧЕНИЯ В КОДЕ КОИ-И

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_ASKVLD\_X MSGADR, DATADR[,RADIX][,POS][,SIZE][,LOLIM][,HILIM][,DEFAULT][,RESERV][,EXWORD]

ГДЕ MSGADR - АДРЕС СТРОКИ КОДА КОИ-8 С НАЧАЛЬНЫМ БАЙТОМ ДЛИНЫ. ЭТА СТРОКА ИСПОЛЬЗУЕТСЯ ДЛЯ ЗАПРОСА ОПЕРАТОРУ;

DATADR - АДРЕС ДЛИННОГО СЛОВА, КОТОРОЕ БУДЕТ ПРИНИМАТЬ ОТВЕТ ОПЕРАТОРА;

RADIX - ОСНОВАНИЕ (СИСТЕМА СЧИСЛЕНИЯ), В КОТОРОМ ОПЕРАТОР ДОЛЖЕН ВЫДАТЬ ОТВЕТ: PAR#\_BIN(ДВОИЧНОЕ);

13.00078-01 33 01

PARM\_OCT (ВОСЬМЕРИЧНОЕ), PARM\_HEX (ШЕСТИНАДЦАТЕРИЧНОЕ)  
ИЛИ PARM\_DEC (ДЕСЯТИЧНОЕ). ПО УМОЛЧАНИЮ - PARM\_DEC;

POS - МЛАДШИЙ БИТ ПОЛЯ, В КОТОРОЕ ПОСТУПАЕТ ОТВЕТ ОПЕРАТОРА;

SIZE - ЧИСЛО БИТОВ В ПОЛЕ;

LOLIM - МИНИМАЛЬНЫЙ ДОПУСТИМЫЙ ЧИСЛОВОЙ ОТВЕТ. ПО УМОЛЧАНИЮ -2\*\*31;

HILIM - МАКСИМАЛЬНЫЙ ДОПУСТИМЫЙ ЧИСЛОВОЙ ОТВЕТ. ПО УМОЛЧАНИЮ 2\*\*31-1;

DEFAULT - ЗНАЧЕНИЕ, ИСПОЛЬЗУЕМОЕ В СЛУЧАЕ, ЕСЛИ ОПЕРАТОР ВВОДИТ НУЛЬ. ПО УМОЛЧАНИЮ - 0;

RESERV - НЕ ИСПОЛЬЗУЕТСЯ;

EXWORD - МАСКА ИСКЛЮЧЕНИЯ. ПАРАМЕТР PARM\_NODEF ОЗНАЧАЕТ ОТСУТСТВИЕ УМОЛЧАНИЯ. PARM\_ATDEF, PARM\_ATLO И PARM\_ATHI УКАЗЫВАЕТ, ЧТО АРГУМЕНТЫ DEFAULT, LOLIM И HILIM УКАЗЫВАЮТ АДРЕСА (А НЕ ЛИТЕРАЛЬНЫЕ ЗНАЧЕНИЯ), ПО КОТОРЫМ НАХОДЯТСЯ СООТВЕТСТВУЮЩИЕ ВЕЛИЧИНЫ.

МАКРОКОМАНДА #DS\_ASKVLD\_X ВОЗВРАЩАЕТ СЛЕДУЮЩИЕ КОДЫ СОСТОЯНИЯ:

DSF\_NORMAL - УСПЕШНОЕ ОКОНЧАНИЕ,

DSF\_PROGERR - УКАЗАНО НЕПРАВИЛЬНОЕ ЧИСЛО АРГУМЕНТОВ,

#DS\_TRUNCATE - ВЕЛИЧИНА, ЗАДАННАЯ ОПЕРАТОРОМ, СЛИШКОМ ВЕЛИКА И НЕ ПОМЕЩАЕТСЯ В БУФЕР.

МАКРОКОМАНДА #DS\_ASKVLD\_X ВЫЗЫВАЕТ СЛУЖЕБНУЮ ПОДПРОГРАММУ СУПЕРВИЗОРА, КОТОРАЯ ЗАПРАШИВАЕТ У ОПЕРАТОРА ЧИСЛОВУЮ ВЕЛИЧИНУ. ПОДПРОГРАММА ПРИНИМАЕТ ОТ ОПЕРАТОРА ЧИСЛОВУЮ

13.00078-01 33 01

СТРОКУ В КОДЕ КОИ-8, ПРЕОБРАЗУЕТ В ДВОИЧНЫЙ ФОРМАТ И ПРИ НЕОБХОДИМОСТИ УСЕКАЕТ СТРОКУ. ЗАТЕМ ПРОВЕРЯЕТСЯ, НАХОДИТСЯ ЛИ ВВЕДЕННАЯ ВЕЛИЧИНА В ПРЕДЕЛАХ ДОПУСТИМОГО ДИАПАЗОНА. ЕСЛИ ДА, ТО ЭТА ВЕЛИЧИНА ЗАПИСЫВАЕТСЯ В ПОЛЕ, УКАЗАННОЕ В МАКРОКОМАНДЕ.

МАКРОКОМАНДА #DS\_ASKVLD\_X НЕ БУДЕТ ВЫПОЛНЯТЬСЯ, ЕСЛИ ОЧИЩЕН ФЛАГ ОПЕРАТОРА.

ЕСЛИ УСТАНОВЛЕН ФЛАГ СООБЩЕНИЯ (PROMPT), ТО ВМЕСТЕ С ЗАПРОСОМ ОПЕРАТОРУ НА ТЕРМИНАЛЕ БУДУТ ПЕЧАТАТЬСЯ ДИАПАЗОНЫ И ВЕЛИЧИНЫ ПО УМОЛЧАНИЮ.

ПАРАМЕТРЫ АРГУМЕНТОВ RADIX И EXWORD ОПРЕДЕЛЯЮТСЯ МАКРОКОМАНДОЙ #DS\_PARDEF.

#### 7.3.4. #DS\_ASKLGCL\_X - ЗАПРОС У ОПЕРАТОРА ЛОГИЧЕСКОГО ОТВЕТА

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_ASKLGCL\_X MSGADR, DATADR[, POS][, ADR1][, ADR2][, DEFAULT]

ГДЕ MSGADR - АДРЕС СТРОКИ КОДА КОИ-8 (ДИРЕКТИВА .ASCIC), ИСПОЛЬЗУЕМОЙ В КАЧЕСТВЕ ЗАПРОСА ОПЕРАТОРУ;

DATADR - АДРЕС БАЙТА, В КОТОРЫЙ ПОСТУПАЕТ ОТВЕТ ОПЕРАТОРА;

POS - НОМЕР БИТА В DATADR, В КОТОРЫЙ ПОПАДАЕТ ОТВЕТ. ПО УМОЛЧАНИЮ 0;

ADR1 - АДРЕС ПЕРЕХОДА ПРИ ПОЛОЖИТЕЛЬНОМ ОТВЕТЕ. ПО УМОЛЧАНИЮ 0, ЧТО ОЗНАЧАЕТ ОТСУТСТВИЕ ПЕРЕХОДА;

ADR2 - АДРЕС ПЕРЕХОДА ПРИ ОТРИЦАТЕЛЬНОМ ОТВЕТЕ. ПО УМОЛЧАНИЮ 0, ЧТО ОЗНАЧАЕТ ОТСУТСТВИЕ ПЕРЕХОДА;



19.00078-01 33 01

RADIX - ОСНОВАНИЕ (СИСТЕМА СЧИСЛЕНИЯ), В КОТОРОМ ДОЛЖЕН БЫТЬ ВЫДАН ОТВЕТ ОПЕРАТОРОМ: PARM\_BIN (ДВОИЧНОЕ), PARM\_OCT (ВОСЬМЕРИЧНОЕ), PARM\_DEC (ДЕСЯТИЧНОЕ), PARM\_HEX (ШЕСТНАДЦАТЕРИЧНОЕ). ПО УМОЛЧАНИЮ PARM\_HEX;

LOLIM - МИНИМАЛЬНЫЙ ДОПУСТИМЫЙ ЧИСЛОВОЙ ОТВЕТ. ПО УМОЛЧАНИЮ - 0;

HILIM - МАКСИМАЛЬНЫЙ ДОПУСТИМЫЙ ЧИСЛОВОЙ ОТВЕТ. ПО УМОЛЧАНИЮ +2\*\*31-1;

DEFAULT - ЗНАЧЕНИЕ, ИСПОЛЬЗУЕМОЕ В СЛУЧАЕ, ЕСЛИ ОПЕРАТОР ВВОДИТ НУЛЬ. ПО УМОЛЧАНИЮ DEFAULT=0;

EXWORD - МАСКА ИСКЛЮЧЕНИЯ. ПАРАМЕТР PARM\_NODEF ОЗНАЧАЕТ ОТСУТСТВИЕ УМОЛЧАНИЯ. PARM\_ATDEF, PARM\_ATLO И PARM\_ATHI ОЗНАЧАЮТ, ЧТО АРГУМЕНТЫ DEFAULT, LOLIM И HILIM УКАЗЫВАЮТ АДРЕСА, ПО КОТОРЫМ НАХОДЯТСЯ СООТВЕТСТВУЮЩИЕ ВЕЛИЧИНЫ (ВМЕСТО ЛИТЕРАЛЬНЫХ ЗНАЧЕНИЙ).

МАКРОКОМАНДА ВОЗВРАЩАЕТ СЛЕДУЮЩИЕ КОДЫ СОСТОЯНИЯ:

DSF\_NORMAL - НОРМАЛЬНОЕ ОКОНЧАНИЕ,

DSF\_PROGERR - ЗАДАНО НЕПРАВИЛЬНОЕ ЧИСЛО АРГУМЕНТОВ.

МАКРОКОМАНДА PDS\_ASKADR\_X ВЫЗЫВАЕТ СЛУЖЕБНУЮ ПОДПРОГРАММУ СУПЕРВИЗОРА, КОТОРАЯ ВЫДАЕТ НА ТЕРМИНАЛ СООБЩЕНИЕ, ЗАПРАШИВАЮЩЕЕ У ОПЕРАТОРА АДРЕС. ПОДПРОГРАММА ПРОВЕРЯЕТ ПОЛУЧЕННЫЙ ОТ ОПЕРАТОРА АДРЕС И ЕСЛИ ОН НАХОДИТСЯ В ДОПУСТИМЫХ ПРЕДЕЛАХ, ЗАПИСЫВАЕТ ПО АДРЕСУ, УКАЗАННОМУ АРГУМЕНТОМ DATAADR.

МАКРОКОМАНДА PDS\_ASKADR\_X НЕ БУДЕТ ВЫПОЛНЯТЬСЯ, ЕСЛИ ОЧИЩЕН ФЛАГ ОПЕРАТОРА. ЕСЛИ УСТАНОВЛЕН ФЛАГ СООБЩЕНИЯ (PROMPT), ТО ВМЕСТЕ С ЗАПРОСОМ НА ТЕРМИНАЛЕ БУДУТ ПЕЧАТАТЬ-

13.00078-01 33 07

СЯ ГРАНИЧНЫЕ ЗНАЧЕНИЯ И ЗНАЧЕНИЯ ПО УМОЛЧАНИЮ.

АРГУМЕНТЫ МАКРОКОМАНДЫ RADIX И EXWORD ОПРЕДЕЛЯЮТСЯ МАКРОКОМАНДОЙ PDS\_PARSE.

#### 7.8.5. PDS\_PARSE\_X - АНАЛИЗ КОМАНДЫ

ФОРМАТ МАКРОКОМАНДЫ:

PDS\_PARSE\_X BUFADR, TREE, ACTION

ГДЕ BUFADR - АДРЕС ЧЕТВЕРНОГО СЛОВА ОПИСАТЕЛЯ БУФЕРА,

В КОТОРОМ НАХОДИТСЯ АНАЛИЗИРУЕМАЯ СТРОКА КОДА КОИ-8;

TREE - АДРЕС ДРЕВОВИДНОЙ СТРУКТУРЫ, КОТОРАЯ ДОЛЖНА БЫТЬ ИСПОЛЬЗОВАНА ДЛЯ АНАЛИЗА;

ACTION - АДРЕС ПОДПРОГРАММЫ ДЕЙСТВИЯ.

МАКРОКОМАНДА PDS\_PARSE\_X ВОЗВРАЩАЕТ СЛЕДУЮЩИЕ КОДЫ СОСТОЯНИЯ:

DSA\_NORMAL - УСПЕШНОЕ ОКОНЧАНИЕ,

DSA\_ERROR - В ДЕРЕВЕ АНАЛИЗА ВСТРЕТИЛАСЬ ОШИБОЧНАЯ КОМАНДА,

DSA\_OVERFLOW - ПЕРЕПОЛНЕНИЕ БУФЕРА.

МАКРОКОМАНДА PDS\_PARSE\_X ВЫЗЫВАЕТ СЛУЖЕБНУЮ ПОДПРОГРАММУ СУПЕРВИЗОРА, КОТОРАЯ ПРОИЗВОДИТ АНАЛИЗ СТРОКИ КОДА КОИ-8 В ЗАДАННОМ БУФЕРЕ. ДЛЯ АНАЛИЗА ИСПОЛЬЗУЕТСЯ ДРЕВОВИДНАЯ СТРУКТУРА, СОСТОЯЩАЯ ИЗ ПОСЛЕДОВАТЕЛЬНОСТИ МАКРОКОМАНД PDS\_CLI. УКАЗАННАЯ СТРУКТУРА ЗАДАЕТСЯ СОСТАВИТЕЛЕМ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ.

ПРИ ОБНАРУЖЕНИИ СОВПАДЕНИЯ В ДЕРЕВЕ ПОДПРОГРАММА ВЫЗЫВАЕТ ПОДПРОГРАММУ ДЕЙСТВИЯ (ACTION). В СЛУЧАЕ НЕСОВПАДЕНИЯ ПРОИСХОДИТ ПЕРЕХОД К ДРУГОЙ ТОЧКЕ В ДЕРЕВЕ АНАЛИЗА КОМАНД.



13.00078-01 33 01

7.3.7. #DS\_CLI - ДЕРЕВО ИНТЕРПРЕТАТОРА  
КОМАНДНОЙ СТРОКИ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_CLI CHAR[,ACTION][,MISS][,ASCII]

ГДЕ CHAR - СИМВОЛ СРАВНЕНИЯ ИЛИ СИМВОЛИЧЕСКОЕ ИМЯ,  
ОПРЕДЕЛЯЮЩЕЕ ФУНКЦИЮ, ЗАДАВАЕМУЮ МАКРОКОМАНДОЙ #DS\_CLI  
(СИМВОЛИЧЕСКИЕ ИМЕНА И СООТВЕТСТВУЮЩИЕ ИМ ФУНКЦИИ ОПИ-  
САНЫ В НАСТОЯЩЕМ ПУНКТЕ);

ACTION - КОД ДЛЯ ПЕРЕДАЧИ ПОДПРОГРАММЕ ДЕЙСТВИЯ;

MISS - АДРЕС (МЕТКА) ПЕРЕХОДА В ДЕРЕВЕ ПРИ НЕСОВПАДЕ-  
НИИ;

ASCII - СТРОКА КОДА КОИ-8 ДЛЯ СРАВНЕНИЯ.

МАКРОКОМАНДА #DS\_CLI ИСПОЛЬЗУЕТСЯ ДЛЯ ПОСТРОЕНИЯ  
ДЕРЕВЬЕВ АНАЛИЗА КОМАНД. КАЖДАЯ МАКРОКОМАНДА ФОРМИРУЕТ В  
ДЕРЕВЕ ОДИН УЗЕЛ.

АНАЛИЗАТОР (ПОДПРОГРАММА СУПЕРВИЗОРА, ВЫЗЫВАЕМАЯ МАК-  
РОКОМАНДОЙ #DS\_PARSE) ПРОДВИГАЕТСЯ ВНИЗ ПО ДЕРЕВУ ДО ТЕХ  
ПОР ПОКА НЕ ОБНАРУЖИТСЯ НЕСОВПАДЕНИЕ ИЛИ ДИРЕКТИВА ПЕРЕХО-  
ДА.

В КАЧЕСТВЕ АРГУМЕНТА CHAR МАКРОКОМАНДЫ #DS\_CLI МОГУТ  
ЗАДАВАТЬСЯ СЛЕДУЮЩИЕ СИМВОЛИЧЕСКИЕ ИМЕНА:

CLIFK\_ERROR - ВЫХОД ИЗ ПОДПРОГРАММЫ АНАЛИЗА КОМАНД  
ВСЛЕДСТВИЕ ОБНАРУЖЕННОЙ ОШИБКИ (БИТ 0  
РЕГИСТРА R0 РАВЕН НУЛЮ);

CLIFK\_EXIT - ВЫХОД ИЗ ПОДПРОГРАММЫ АНАЛИЗА КОМАНД ПРИ  
ОКОНЧАНИИ АНАЛИЗА, ЕСЛИ НЕ ОБНАРУЖЕНА ОШИБКА  
(БИТ 0 РЕГИСТРА R0 РАВЕН ЕДИНИЦЕ);

13.00078-01 33 01

- CLIPK\_BR - БЕЗУСЛОВНЫЙ ПЕРЕХОД В ПРЕДЕЛАХ ДЕРЕВА ПО АРГУМЕНТУ MISS;
- CLIPK\_BIF - ПЕРЕХОД К СЛЕДУЮЩЕМУ УЗЛУ, ЕСЛИ БИТ 0 РЕГИСТРА RD РАВЕН 0. В ПРОТИВНОМ СЛУЧАЕ ПЕРЕХОД ПО АДРЕСУ, УКАЗАННОМУ АРГУМЕНТОМ MISS;
- CLIPK\_SPACE - ПРОПУСК ПРОБЕЛОВ И ТАБУЛЯЦИИ И ВЪЗОВ ПОДПРОГРАММЫ ДЕЙСТВИЯ (ACTION) ПРИ ИХ ОБНАРУЖЕНИИ (РЕГИСТР R8 УКАЗЫВАЕТ СЛЕДУЮЩИИ СИМВОЛ, ОТЛИЧНЫЙ ОТ ПРОБЕЛА);
- CLIPK\_NUM - ПРОПУСК ЦИФРОВЫХ ТОЛЕИ И ВЪЗОВ ПОДПРОГРАММЫ ДЕЙСТВИЯ С ЦИФРОВЫМИ ДАННЫМИ (ЦИФРОВЫЕ ДАННЫЕ ПОМЕЩАЮТСЯ В РЕГИСТР R10). ИСПОЛЬЗУЕТСЯ ОСНОВАНИЕ (СИСТЕМА СЧИСЛЕНИЯ) ПО УМОЛЧАНИЮ. ЕСЛИ ЦИФРОВЫХ ДАННЫХ НЕТ, ТО ПЕРЕХОД ПО АДРЕСУ, УКАЗАННОМУ АРГУМЕНТОМ MISS;
- CLIPK\_ALPHA - ПРОПУСК БУКВЕННЫХ ПОЛЕЙ;
- CLIPK\_ALNUM - ПРОПУСК БУКВЕННО-ЦИФРОВЫХ ПОЛЕЙ;
- CLIPK\_OCT - АНАЛОГИЧНО CLIPK\_NUM, НО ИСПОЛЬЗУЕТСЯ ОСНОВАНИЕ ВОСЕМЬ;
- CLIPK\_DEC - АНАЛОГИЧНО CLIPK\_NUM, НО ИСПОЛЬЗУЕТСЯ ОСНОВАНИЕ ДЕСЯТЬ;
- CLIPK\_HEX - АНАЛОГИЧНО CLIPK\_NUM, НО ИСПОЛЬЗУЕТСЯ ОСНОВАНИЕ ШЕСТНАДЦАТЬ;
- CLIPK\_STRING - ДЛЯ СРАВНЕНИЯ ИСПОЛЬЗУЕТСЯ АРГУМЕНТ ASCII. НЕОБХОДИМО СОВПАДЕНИЕ ТОЛЬКО С ПЕРВЫМ СИМВОЛОМ СТРОКИ.

ЗНАЧЕНИЯ ОПИСАННЫХ СИМВОЛИЧЕСКИХ ИМЕН ЗАДАЕТСЯ С ПОМОЩЬЮ МАКРОКОМАНДЫ #DS\_CLIDEF.:

13.00078-01 33 01

ПРИ ВХОДЕ В ПОДПРОГРАММУ ДЕЙСТВИЯ (ACTION) РЕГИСТРЫ  
СОДЕРЖАТ СЛЕДУЮЩУЮ ИНФОРМАЦИЮ:

R0 - КОД АРГУМЕНТА ACTION ИЗ ДЕРЕВА,

R7 - УКАЗАТЕЛЬ ДЕРЕВА АНАЛИЗА,

R8 - УКАЗАТЕЛЬ ВХОДНОЙ СТРОКИ,

R9 - ДЛИНА ОСТАТКА ВХОДНОЙ СТРОКИ,

R10 - БУФЕР ЦИФРОВЫХ ДАННЫХ.

7.6.8. ПРИМЕР ИСПОЛЬЗОВАНИЯ МАКРОКОМАНД  
ДИАЛОГА МЕЖДУ ПРОГРАММОЙ И ОПЕРАТОРОМ

ПРИМЕР.

```
; ГОЛОВНОЙ МОДУЛЬ
NULL=0
SUB1=1
SUB2=2
STOP=3
INIT=4
CHAR_BUF:: .BLKB 132 ; БУФЕР ВВОДА-ВЫВОДА НА 132 БАЙТА
CMD_BUF:: .QUAD ; ЧЕТВЕРНОЕ СЛОВО
T_BLK:: .ADDRESS 0 ; БУФЕР ДЛЯ ПОДПРОГРАММЫ ДЕЙСТВИЯ
.PAGE
; ++
; ДЕРЕВО ИНТЕРПРЕТАТОРА КОМАНДНОЙ СТРОКИ. ОНО ПОЗВОЛЯЕТ
; ИНТЕРПРЕТИРОВАТЬ СЛЕДУЮЩИЕ КОМАНДЫ ОПЕРАТОРА:
; SUB1
; SUB2
; STOP
; ДВУЗНАЧЕННАЯ ИЛИ НЕДОПУСТИМАЯ КОМАНДА ВЫЗЫВАЕТ СООБЩЕНИЕ
; ОБ ЭТОМ ОПЕРАТОРУ
; --
FIRTREE::
10H: #DS_CLI CLI#K_BR, INIT, 20H ; ОЧИСТКА БЛОКА ОБОЗНАЧЕ-
; НИЙ
20H: #DS_CLI CHAR=<^A^S^>,- ; ПЕРВЫЙ СИМВОЛ S?
; ACTION=NULL,- ; ИНФОРМАЦИИ НЕДОСТАТОЧНО
; MISS=110H ; ЕСЛИ ПЕРВЫЙ СИМВОЛ НЕ S,
; ПЕРЕХОД ПО МЕТКЕ 110H
30H: #DS_CLI CHAR=CLI#K_STRING,- ; СЛЕДУЮЩИЕ 3 СИМВОЛА TOP?
; ACTION=STOP,- ; ДА
; MISS=50H,- ; НЕТ
; ASCII=<^TOP^>
40H: #DS_CLI CHAR=CLI#K_EXIT ; ОПЕРАТОР ВЗЕЛ КОМАНДУ
; ; STOP
50H: #DS_CLI CHAR=<^A^U^>,- ; СЛЕДУЮЩИЙ СИМВОЛ U?
; ACTION=NULL,- ; ИНФОРМАЦИИ ПО-ПРЕЖНЕМУ
; ; МАЛО
; MISS=11DH ; ОШИБКА
```

13.00078-01 33 01

```

604: ADS_CLI CHAR=<A^B^>,- ; СЛЕДУЮЩИЙ СИМВОЛ В?
      ACTION=NULL,- ; ИНФОРМАЦИИ ЕЩЕ МАЛО
      MISS=1104 ; ОШИБКА
704: ADS_CLI CHAR=<A^1^>,- ; СЛЕДУЮЩИЙ СИМВОЛ 1?
      ACTION=SUB1,- ; ДА
      MISS=904 ; НЕТ
304: ADS_CLI CHAR=CLIKK_EXIT ; ОПЕРАТОР ВВЕЛ КОМАНДУ
      ; SUB1
904: ADS_CLI CHAR=<A^2^>,- ; СЛЕДУЮЩИЙ СИМВОЛ 2?
      ACTION=SUB2,- ; ДА
      MISS=1104 ; НЕТ
1004: ADS_CLI CHAR=CLIKK_EXIT ; ОПЕРАТОР ВВЕЛ КОМАНДУ
      ; SUB2
1104: ADS_CLI CHAR=CLIKK_ERROR ; ОПЕРАТОР ВВЕЛ
      ; НЕПРАВИЛЬНУЮ КОМАНДУ

```

.SBTTL ОТДЕЛ ПОДПРОГРАММ

```

ACT_ENTRY:: CASEL RU, #0, #4
104: .WORD ACT_NULL -104 ; ЕСЛИ NULL
      .WORD ACT_SUB1 -104 ; ЕСЛИ SUB1
      .WORD ACT_SUB2 -104 ; ЕСЛИ SUB2
      .WORD ACT_STOP -104 ; ЕСЛИ STOP
      .WORD ACT_INIT -104 ;

```

```

204: ADS_ERRSYS_S ; СЕРЬЕЗНАЯ СИСТЕМНАЯ
      ; ОШИБКА
      ADS_ABORT PROGRAM ; ОТМЕНА ВЫПОЛНЕНОЙ
      ; ПРОГРАММЫ

```

```

ACT_INIT: CLRL T_BL
          RSB
ACT_SUB1: MOVAL SUB1_ADR, T_BL ; ЗАСЫЛКА АДРЕСА
          ; ПОДПРОГРАММЫ ДЕЙСТВИЯ
          RSB
ACT_SUB2: MOVAL SUB2_ADR, T_BL ; ЗАСЫЛКА АДРЕСА
          ; ПОДПРОГРАММЫ ДЕЙСТВИЯ
ACT_STOP: MOVAL STOP_ADR, T_BL ;
ACT_NULL: RSB

```

```

GETLINE:: .ASCII /ВВЕДИТЕ,ПОЖАЛУЙСТА,КОМАНДУ/ ; ДИРЕКТИВЫ
          ; FAQ ЗДЕСЬ
          ; ПРИМЕНЯТЬ
          ; НЕЛЬЗЯ

```

```

AMBICMD:: .ASCII /ДВУСМЫСЛЕННАЯ КОМАНДА/
INVCMDB:: .ASCII /НЕДЕЙСТВИТЕЛЬНАЯ КОМАНДА/

```

-----  
; ТЕСТОВЫЙ МОДУЛЬ

```

      ADS_BGNTEST
LABEL1::
104: ADS_ASKSTR_S- ; ЗАПРОС КОМАНДЫ
      ; ОПЕРАТОРА
      MSGADR=GETLINE,- ; СТРОКА ЗАПРОСА
      BUFADR=CHAR_BUF ; БУФЕР ДЛЯ ПРИЕМА КОМАНДЫ
      ; ОПЕРАТОРА
      ADS_BNCOMLETE 104 ; ЕСЛИ ПОПЫТКА
      ; НЕУСПЕШНА...
      MOVZBL CHAR_BUF, CMDBUF ; ДЛИНА СТРОКИ

```

19.00078-01 33 01

```

MOVAL CHAR_BUF+1,CMDBUF+4 ; ФОРМИРОВАНИЕ
                                ; ЧЕТЫРЕХСЛОВНОГО
                                ; ОПИСАТЕЛЯ
#DS_PARSE_S BUFADR=CMDBUF,- ; АНАЛИЗ КОМАНДЫ
                                TREE=FIRTREE,-
                                ACTION=ACT_ENTRY
#DS_COMPLETE 20# ; АНАЛИЗ КОМАНДЫ ЗАВЕРШЕН
#DS_PRINTF_S FORMAT=INVCMO ; ПЕЧАТЬ СООБЩЕНИЯ
                                ; "НЕДЕЙСТВИТЕЛЬНАЯ
                                ; КОМАНДА"

BRB 10#
20# : TSTL T_BL ; БЛОК T_BL ПУСТ?
BRB 30# ; НЕТ. БЫЛА ВВЕДЕНА
                                ; ДЕЙСТВИТЕЛЬНАЯ КОМАНДА
#DS_PRINTF_S FORMAT=AMVICMD ; ДА? БЫЛА ВРЕДЕНА
                                ; ДВУМЫСЛЕННАЯ КОМАНДА
BRW 10# ; НУЖНО ПОЛУЧИТЬ ДРУГУЮ
                                ; КОМАНДУ ОТ ОПЕРАТОРА
30# : JSB @T_BL ; ВЫПОЛНЕНИЕ КОМАНДЫ
.
.
SUB1_ADR:
.
.
RSB
SUB2_ADR:
.
.
RSB
STOP_ADR:
#DS_ENDTEST ; ВЫХОД

```

В ПРИВЕДЕННОМ ПРИМЕРЕ МАКРОКОМАНДА #DS\_ASKSTR\_S ВЫЗЫВАЕТ ПОДПРОГРАММУ СУПЕРВИЗОРА, КОТОРАЯ ВЫДАЕТ НА ТЕРМИНАЛ ОПЕРАТОРА СООБЩЕНИЕ (ЗАПРОС), ЗАПИСАННОЕ ПО АДРЕСУ SETLINE. ОТВЕТ ОПЕРАТОРА ЗАСЫЛАЕТСЯ В БУФЕР CHAR\_BUF. ЕСЛИ УКАЗАННАЯ ПОДПРОГРАММА СУПЕРВИЗОРА НЕ СМОЖЕТ ВЫПОЛНИТЬ ЗАДАЧУ УСПЕШНО, ТО МАКРОКОМАНДА #DS\_ASKSTR\_S ВЫПОЛНЯЕТСЯ СНОВА. ЕСЛИ МЕ ПОДПРОГРАММА ВЫПОЛНЕНА УСПЕШНО, ТО УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ МАКРОКОМАНДЕ #DS\_PARSE\_S.

МАКРОКОМАНДА #DS\_PARSE\_S ВЫЗЫВАЕТ ПОДПРОГРАММУ СУПЕРВИЗОРА, КОТОРАЯ ПРОИЗВОДИТ АНАЛИЗ КОМАНДЫ, ПОЛУЧЕННОЙ ОТ ОПЕРАТОРА И НАХОДЯЩЕЙСЯ В БУФЕРЕ CHAR\_BUF.

АНАЛИЗ ПРОИЗВОДИТСЯ В СООТВЕТСТВИИ С ДЕРЕВОМ ИНТЕРПРЕ-

13.00078-01 33 01

ТАТОРА КОМАНДНОЙ СТРОКИ. ЭТО ДЕРЕВО СОЗДАНО С ПОМОЩЬЮ МАКРОКОМАНД #DS\_CLI И НАХОДИТСЯ ПО АДРЕСУ FIRSTREE.

НА РИС.10 (СМ. ПРИЛОЖЕНИЕ 3) ПОКАЗАНА КОНФИГУРАЦИЯ ДЕРЕВА. ПО АДРЕСУ 20Я ПРОВЕРЯЕТСЯ ПЕРВЫЙ СИМВОЛ. ЕСЛИ ОН НЕ S, ТО ЭТО ОЗНАЧАЕТ, ЧТО ОПЕРАТОР ВВЕЛ НЕПРАВИЛЬНУЮ КОМАНДУ И УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ МЕТКЕ 110Я, ОПРЕДЕЛЕННОЙ АРГУМЕНТОМ MISS.

ЕСЛИ ПЕРВЫЙ СИМВОЛ S, ТО УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ К МЕТКЕ ACT\_NULL И ДАЛЕЕ К 30Я, ГДЕ ТРИ СЛЕДУЮЩИХ СИМВОЛА СРАВНИВАЮТСЯ С СИМВОЛАМИ TOP. T, TO ИЛИ TOP ЯВЛЯЮТСЯ СОВПАДЕНИЕМ, А ЛИБАЯ ДРУГАЯ КОМБИНАЦИЯ - НЕТ. ПРИ СОВПАДЕНИИ УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ К 40Я, В КОТОРОЙ НАХОДИТСЯ МАКРОКОМАНДА #DS\_CLI С АРГУМЕНТОМ SLINK\_EXIT, КОТОРАЯ ПРИВОДИТ К ВЫХОДУ ИЗ ДЕРЕВА.

ЕСЛИ ЖЕ ИМЕЕТ МЕСТО НЕСОВПАДЕНИЕ, ТО УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ ПО АДРЕСУ 50Я, ГДЕ ВЫПОЛНЯЕТСЯ ПРОВЕРКА НА СИМВОЛ U И Т.Д.

РАССМАТРИВАЕМОЕ ДЕРЕВО ИМЕЕТ 6 ВЫХОДОВ:

40Я - ПРИ STOP,

80Я - ПРИ SUB1,

100Я - ПРИ SUB2,

110Я - ПРИ НЕПРАВИЛЬНОЙ КОМАНДЕ,

20Я, 30Я И 60Я - ПРИ ДВУСМЫСЛЕННЫХ КОМАНДАХ.

ЕСЛИ ОПЕРАТОР ВВОДИТ С КЛАВИАТУРЫ STOP, SUB1 ИЛИ SUB2, ТО УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ К МЕТКЕ ACT\_ENTRY. ИНСТРУКЦИЯ CASEL ВЫБИРАЕТ ПОДПРОГРАММУ ACT\_STOP ДЛЯ КОМАНДЫ STOP, ACT\_SUB1-ДЛЯ SUB1 И Т.Д. КАЖДАЯ ИЗ ЭТИХ ПОДПРОГРАММ ЗАСЫЛАЕТ РАЗНЫЕ АДРЕСА В БЛОК T\_3L ПОСЛЕ ЕГО ОЧИСТКИ ПОДПРОГРАММОЙ ACT\_INIT. ИНСТРУКЦИЯ JSB ОТ\_3L ПОСЛЕ МАКРОКОМАНДЫ #DS\_PARSE\_S ПЕРЕДАЕТ ЗАТЕМ УПРАВЛЕНИЕ ОДНОЙ ИЗ ПОДПРОГРАММ

(SUB1\_ADR, SUB2\_ADR или STOP\_ADR).

ЕСЛИ ОПЕРАТОР ВВЕЛ ДВУСМЫСЛЕННУЮ КОМАНДУ (S, SU или SUB), БЛОК T\_VL ОСТАЕТСЯ ПУСТЫМ. ЭТО ОБНАРУЖИВАЕТ ИНСТРУКЦИЯ ПО АДРЕСУ 20H В ТЕСТОВОЙ ПОДПРОГРАММЕ И МАКРОКОМАНДА #DS\_PRINTF\_S ОПОВЕЩАЕТ ОПЕРАТОРА О ДВУСМЫСЛЕННОСТИ КОМАНДЫ, ПОСЛЕ ЧЕГО УПРАВЛЕНИЕ ВОЗВРАЩАЕТСЯ ПО АДРЕСУ 10H ДЛЯ ВЫДАЧИ ЗАПРОСА ОПЕРАТОРУ С ЦЕЛЮ ПОЛУЧЕНИЯ ДРУГОЙ КОМАНДЫ.

### 7.9. МАКРОКОМАНДЫ СООБЩЕНИИ ОБ ОШИБКАХ

В ДИАГНОСТИЧЕСКОЙ СИСТЕМЕ МСПД ИСПОЛЬЗУЕТСЯ ТРИ ВИДА ИНФОРМАЦИИ ОБ ОШИБКАХ:

1) КАК СУПЕРВИЗОР, ТАК И ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ ФОРМИРУЮТ ЗАГОЛОВОК, В КОТОРЫЙ ВКЛЮЧАЕТСЯ ИМЯ ПРОГРАММЫ, ВЕРСИЯ, ИЗМЕНЕНИЕ, НОМЕРА ТЕСТА, ПОДТЕСТА И ОШИБКИ, ТИП ОШИБКИ, ПРОВЕРЯЕМОЕ УСТРОЙСТВО И КРАТКОЕ СООБЩЕНИЕ;

2) ДИАГНОСТИЧЕСКАЯ ПРОГРАММА ФОРМИРУЕТ ОСНОВНУЮ ИНФОРМАЦИЮ, ВКЛЮЧАЮЩУЮ ОПИСАНИЕ ОШИБКИ, СОДЕРЖИМОЕ РЕГИСТРОВ УСТРОЙСТВА, А ТАКЖЕ ОЖИДАЕМЫЕ И ПОЛУЧЕННЫЕ ДАННЫЕ;

3) ДИАГНОСТИЧЕСКАЯ ПРОГРАММА ФОРМИРУЕТ РАСШИРЕННУЮ ИНФОРМАЦИЮ, КОТОРАЯ ЗАВИСИТ ОТ КОНКРЕТНОЙ ПРОГРАММЫ И ВКЛЮЧАЕТ ТАКЖЕ СТАТИСТИКУ (НАПРИМЕР, ЧИСЛО ПЕРЕДАННЫХ БАЙТОВ).

ПЕРЕЧИСЛЕННЫЕ ВИДЫ ИНФОРМАЦИИ ОБ ОШИБКАХ СООТВЕТСТВУЮТ ТРЕМ ФЛАГАМ ЗАПРЕТА СООБЩЕНИЙ ОБ ОШИБКАХ (IE1, IE2 И IE3) В ЯЗЫКЕ КОМАНД СУПЕРВИЗОРА.

ПРИ УСТАНОВКЕ ФЛАГА ЗВУКОВОГО СИГНАЛА (BELL) СУПЕРВИЗОР ОБЕСПЕЧИВАЕТ ВЫДАЧУ ЗВУКОВОГО СИГНАЛА ПРИ КАЖДОЙ ОБНАРУЖЕННОЙ ПРОГРАММОЙ ОШИБКЕ.

ДЛЯ ПОДПРОГРАММ ПЕЧАТИ МОЖЕТ ИСПОЛЬЗОВАТЬСЯ ТАКЖЕ МАКРОКОМАНДА #DS\_CVTRIG\_X ДЛЯ ПРЕОБРАЗОВАНИЯ ИНФОРМАЦИИ, НАХО-

13.00078-01 33 01

ДЯЩЕИЯ В РЕГИСТРЕ, КОТОРАЯ ПОЗВОЛЯЕТ ПРОГРАММИСТУ ОПРЕДЕ-  
ЛЯТЬ МНЕМОНИЧЕСКИЕ ОБОЗНАЧЕНИЯ БИТОВ РЕГИСТРА С ОШИБКОЙ.

#### 7.9.1. МАКРОКОМАНДЫ ПЕЧАТИ ЗАГОЛОВКА СООБЩЕНИЯ

В ГРУППУ ВХОДЯТ МАКРОКОМАНДЫ #DS\_ERRSYS\_X,  
#DS\_ERRDEV\_X, #DS\_ERRHARD\_X, #DS\_ERRSOFT\_X.

МАКРОКОМАНДА #DS\_ERRSYS\_X СЛУЖИТ ДЛЯ ПЕЧАТИ ЗАГОЛОВКА  
СООБЩЕНИЯ ПРИ НЕУСТРАНИМОЙ СИСТЕМНОЙ ОШИБКЕ, #DS\_ERRDEV\_X -  
ПРИ НЕУСТРАНИМОЙ ОШИБКЕ УСТРОЙСТВА, #DS\_ERRHARD\_X - ПРИ  
ОБНАРУЖЕНИИ ОШИБКИ ОБОРУДОВАНИЯ, #DS\_ERRSOFT\_X - ПРИ ОШИБКЕ  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И УСТРАНИМОЙ ОШИБКЕ.

ВСЕ УКАЗАННЫЕ МАКРОКОМАНДЫ ИМЕЮТ ОДИНАКОВЫЙ ФОРМАТ И  
ВЫЗЫВАЮТ ПОДПРОГРАММЫ СУПЕРВИЗОРА, КАЖДАЯ ИЗ КОТОРЫХ ВЫДАЕТ  
ОПЕРАТОРУ ТРИ СТРОКИ СООБЩЕНИЯ ОБ ОШИБКЕ. В СООБЩЕНИИ УКА-  
ЗЫВАЕТСЯ НОМЕР ВЕРСИИ ПРОГРАММЫ, НОМЕР ПРОХОДА, НОМЕР ТЕСТА  
И ПОДТЕСТА, ДАТА И ВРЕМЯ. КРОМЕ ТОГО, ПЕЧАТАЕТСЯ КРАТКОЕ  
СООБЩЕНИЕ, ОПРЕДЕЛЯЕМОЕ ПРОГРАММИСТОМ.

ПРИ НЕОБХОДИМОСТИ РАСПЕЧАТКИ ДОПОЛНИТЕЛЬНОЙ ИНФОРМАЦИИ  
ПРОГРАММИСТ ДОЛЖЕН УКАЗАТЬ АДРЕС ПОДПРОГРАММЫ ПЕЧАТИ АРГУ-  
МЕНТОМ МАКРОКОМАНДЫ PRLINK. ПОДПРОГРАММА ПЕЧАТИ ДОЛЖНА ЗАК-  
ЛЮЧАТЬСЯ МЕЖДУ МАКРОКОМАНДАМИ #DS\_BGNMESSAGE И  
#DS\_ENDMESSAGE.

МАКРОКОМАНДЫ ПЕЧАТИ ОСНОВНОЙ И РАСШИРЕННОЙ ИНФОРМАЦИИ  
ДОЛЖНЫ НАХОДИТЬСЯ ВНУТРИ УПОМЯНУТОЙ ПОДПРОГРАММЫ ПЕЧАТИ.

ОТМЕТИМ, ЧТО ПРИ УСТАНОВЛЕННОМ ФЛАГЕ IET РАСПЕЧАТКА  
ЗАГОЛОВКА СООБЩЕНИЯ ОБ ОШИБКЕ НЕ ПРОИЗВОДИТСЯ.



13.00073-01 33 01'

7.9.1.1. #DS\_ERRSYS\_X - РАСПЕЧАТКА ЗАГЛАВНОЙ  
ИНФОРМАЦИИ ПРИ НЕУСТРАНИМОЙ СИСТЕМНОЙ  
ОШИБКЕ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_ERRSYS\_X [NUM][,UNIT][,MSGADR][,PRLINK][,P1-P6]

ГДЕ NUM - УНИКАЛЬНЫЙ НОМЕР ОШИБКИ В ПРЕДЕЛАХ ТЕКУЩЕГО  
ПОДТЕСТА. ПО УМОЛЧАНИЮ АВТОМАТИЧЕСКИ УСТАНОВЛИВАЕТСЯ  
СЛЕДУЮЩИЙ ПО ПОРЯДКУ НОМЕР ОШИБКИ;

UNIT - ЛОГИЧЕСКИЙ НОМЕР ПРОВЕРЯЕМОГО УСТРОЙСТВА;

MSGADR - АДРЕС СТРОКИ СИМВОЛОВ С НАЧАЛЬНЫМ БАЙТОМ ДЛИ-  
НЫ (ДИРЕКТИВА МАКРОАССЕМБЛЕРА .ASCII). В СТРОКЕ СОДЕР-  
ЖИТСЯ ИНФОРМАЦИЯ, КОТОРАЯ ДОЛЖНА БЫТЬ ВКЛЮЧЕНА В ЗАГО-  
ЛОВОК СООБЩЕНИЯ ОБ ОШИБКЕ. УКАЗАННАЯ ИНФОРМАЦИЯ ДОЛЖНА  
СОДЕРЖАТЬ КРАТКОЕ ОПИСАНИЕ ОШИБКИ;

PRLINK - АДРЕС ПОДПРОГРАММЫ ПЕЧАТИ ДОПОЛНИТЕЛЬНОЙ  
ИНФОРМАЦИИ ОБ ОБНАРУЖЕННОЙ ОШИБКЕ. ДАННАЯ ПОДПРОГРАММА  
ДОЛЖНА РАЗМЕЩАТЬСЯ МЕЖДУ МАКРОКОМАНДАМИ #DS\_BGNMESSAGE  
И #DS\_ENDMESSAGE. ВЫПОЛНЕНИЕ ЭТОЙ ПОДПРОГРАММЫ НЕ  
ЗАВИСИТ ОТ СОСТОЯНИЯ ФЛАГА ОСТАНОВА ПО ОШИБКЕ;

P1-P6 - ОТ ОДНОГО ДО ШЕСТИ ВСПОМОГАТЕЛЬНЫХ ПАРАМЕТРОВ,  
КОТОРЫЕ МОГУТ ПЕРЕДАВАТЬСЯ ПОДПРОГРАММЕ ПЕЧАТИ СООБЩЕ-  
НИЯ ОБ ОШИБКЕ.

МАКРОКОМАНДУ #DS\_ERRSYS\_X СЛЕДУЕТ ИСПОЛЬЗОВАТЬ ПРИ  
ОБНАРУЖЕНИИ ФАТАЛЬНОЙ СИСТЕМНОЙ ОШИБКИ (НАПРИМЕР, ЕСЛИ  
ОБНАРУЖЕН СБОИ ПРИ ВЫЗОВЕ КАНАЛА).

РАССМОТРИМ ПРИМЕР ИСПОЛЬЗОВАНИЯ МАКРОКОМАНДЫ

ADS\_ERRSYS\_X В ПОДПРОГРАММЕ ИНИЦИАЛИЗАЦИИ.

ПРИМЕР.

```
.SBTTL INIT
ADS_BGNINIT
.
.
.
ADS_CHANNEL_S #0, #CHCP_INITA ; ИНИЦИАЛИЗАЦИЯ АДАПТЕРА
                                ; UBA
ADS_BNEROR  TUS ; ЕСТЬ ОШИБКА?
ADS_ERRSYS_S UNIT=0,- ; ДА, НЕУСТРАНИМАЯ СИС-
                        ; ТЕМНАЯ ОШИБКА
                        MSGADR=UBA_INIT_ERR,- ; СООБЩЕНИЕ
                        PRLINK=DUMP_UBA ; ПОДПРОГРАММА ПЕЧАТИ
                        ; СООБЩЕНИИ ОБ ОШИБКЕ
ADS_ABORT PROGRAM ; ОТМЕНА ВЫПОЛНЕНИЯ
                        ; ПРОГРАММЫ
104: ; ПРОДОЛЖЕНИЕ ПРОГРАММЫ
```

7.9.1.2. ADS\_ERRDEV\_X - ПАСПЕЧАТКА ЗАГЛАВНОЙ  
ИНФОРМАЦИИ ПРИ НЕУСТРАНИМОЙ ОШИБКЕ  
УСТРОЙСТВА

ФОРМАТ МАКРОКОМАНДЫ:

```
ADS_ERRDEV_X LNUMJL,UNITJL,MSGADRJL,PRLINKJL,P1-P6J
```

ГДЕ NUM - УНИКАЛЬНЫЙ НОМЕР ОШИБКИ В ПРЕДЕЛАХ ТЕКУЩЕГО  
ПОДТЕСТА. ПО УМОЛЧАНИЮ АВТОМАТИЧЕСКИ УСТАНОВЛИВАЕТСЯ  
СЛЕДУЮЩИЙ ПО ПОРЯДКУ НОМЕР ОШИБКИ;

UNIT - ЛОГИЧЕСКИЙ НОМЕР ПРОВЕРЯЕМОГО УСТРОЙСТВА;

MSGADR - АДРЕС СТРОКИ СИМВОЛОВ С НАЧАЛЬНЫМ БАЙТОМ ДЛИ-  
НЫ (ДИРЕКТИВА МАКРОАССЕМБЛЕРА .ASCII). В ЭТОЙ СТРОКЕ  
ДОЛЖНО СОДЕРЖАТЬСЯ КРАТКОЕ ОПИСАНИЕ ОШИБКИ, КОТОРОЕ  
ВКЛЮЧАЕТСЯ В ЗАГОЛОК СООБЩЕНИЯ ОБ ОШИБКЕ;

PRLINK - АДРЕС ПОДПРОГРАММЫ ПЕЧАТИ ДОПОЛНИТЕЛЬНОЙ  
ИНФОРМАЦИИ ОБ ОБНАРУЖЕННОЙ ОШИБКЕ. ВЫПОЛНЕНИЕ ПОДПРОГ-  
РАММЫ НЕ ЗАВИСИТ ОТ СОСТОЯНИЯ ФЛАГА ОСТАНОВА ПО ОШИБ-

13.00078-01 33 01\*

КЕ. ПОДПРОГРАММА ДОЛЖНА РАЗМЕЩАТЬСЯ МЕЖДУ МАКРОКОМАНДАМИ #DS\_BGNMESSAGE И #DS\_ENDMESSAGE;

P1-P6 - ОТ ОДНОГО ДО ШЕСТИ ДОПОЛНИТЕЛЬНЫХ ПАРАМЕТРОВ, КОТОРЫЕ МОГУТ ПЕРЕДАВАТЬСЯ ПОДПРОГРАММЕ ПЕЧАТИ СООБЩЕНИЯ ОБ ОШИБКЕ.

МАКРОКОМАНДУ #DS\_ERRDEV\_X СЛЕДУЕТ ИСПОЛЬЗОВАТЬ ПОСЛЕ ОБНАРУЖЕНИЯ ФАТАЛЬНОЙ ОШИБКИ УСТРОЙСТВА, КОТОРАЯ НЕ ПОЗВОЛЯЕТ ПРОИЗВОДИТЬ ДАЛЬНЕЙШУЮ ПРОВЕРКУ УСТРОЙСТВА.

### 7.9.1.3. #DS\_ERRHARD\_X - РАСПЕЧАТКА ЗАГЛАВНОЙ ИНФОРМАЦИИ ОБ АППАРАТНОЙ ОШИБКЕ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_ERRHARD\_X [NUM][,UNIT][,MSGADR][,PRLINK][,P1-P6]

ГДЕ NUM - УНИКАЛЬНЫЙ НОМЕР ОШИБКИ В ПРЕДЕЛАХ ТЕКУЩЕГО ПОДТЕСТА. ПО УМОЛЧАНИЮ АВТОМАТИЧЕСКИ УСТАНАВЛИВАЕТСЯ СЛЕДУЮЩИЙ ПО ПОРЯДКУ НОМЕР ОШИБКИ;

UNIT - ЛОГИЧЕСКИЙ НОМЕР ПРОВЕРЯЕМОГО УСТРОЙСТВА;

MSGADR - АДРЕС СТРОКИ СИМВОЛОВ С НАЧАЛЬНЫМ БАЙТОМ ДЛИНЫ (ДИРЕКТИВА МАКРОАССЕМБЛЕРА .ASCII). В ЭТОЙ СТРОКЕ ДОЛЖНО СОДЕРЖАТЬСЯ КРАТКОЕ ОПИСАНИЕ ОШИБКИ, КОТОРОЕ ВКЛЮЧАЕТСЯ В ЗАГОЛОВОК СООБЩЕНИЯ ОБ ОШИБКЕ;

PRLINK - АДРЕС ПОДПРОГРАММЫ ПЕЧАТИ ДОПОЛНИТЕЛЬНОЙ ИНФОРМАЦИИ ОБ ОБНАРУЖЕННОЙ ОШИБКЕ. ДАННАЯ ПОДПРОГРАММА ДОЛЖНА РАЗМЕЩАТЬСЯ МЕЖДУ МАКРОКОМАНДАМИ #DS\_BGNMESSAGE И #DS\_ENDMESSAGE. ПОДПРОГРАММА ВЫПОЛНЯЕТСЯ НЕЗАВИСИМО ОТ СОСТОЯНИЯ ФЛАГА ОСТАНОВА ПО ОШИБКЕ;

13.00078-01 33 01

P7-P6 - ОТ ОДНОГО ДО ШЕСТИ ДОПОЛНИТЕЛЬНЫХ ПАРАМЕТРОВ, КОТОРЫЕ МОГУТ ПЕРЕДАВАТЬСЯ ТОДПРОГРАММЕ ПЕЧАТИ СООБЩЕНИЯ ОБ ОШИБКЕ.

МАКРОКОМАНДУ ADS\_ERRHARD\_X СЛЕДУЕТ ИСПОЛЬЗОВАТЬ ПОСЛЕ ОБНАРУЖЕНИЯ ОШИБКИ ОБОРУДОВАНИЯ.

РАССМОТРИМ ПРИМЕР ИСПОЛЬЗОВАНИЯ МАКРОКОМАНДЫ ADS\_ERRHARD\_X.

ПРИМЕР.

```
100: MOVW #^X5068, @DZCSR      ; ЗАПИСЬ В РЕГИСТР CSR
      MOVW @DZCSR, CSRW       ; ЧТЕНИЕ СОДЕРЖИМОГО CSR
      XORWS #^X5068, CSRW, XORW ; СРАВНЕНИЕ ЗАПИСАННЫХ И
                                ; ПРОЧИТАННЫХ ДАННЫХ
      BEQL 204                ; УСПЕШНОЕ ВЫПОЛНЕНИЕ
      ADS_ERRHARD_S-         ; ПЕЧАТЬ СООБЩЕНИЯ ОБ
                                ; ОШИБКЕ
                                ; УСТРОЙСТВУ
                                ; АДРЕС СООБЩЕНИЯ
      UNIT=LOG_UNIT,-
      MSGADR=REGERR
204: ADS_CKLOOP.....
```

7.9.1.4. ADS\_ERRSOFT\_X - РАСПЕЧАТКА ЗАГЛАВНОЙ ИНФОРМАЦИИ ОБ УСТРАНИМОЙ ОШИБКЕ ПРОГРАММНЫХ СРЕДСТВ

ФОРМАТ МАКРОКОМАНДЫ:

ADS\_ERRSOFT\_X [NUM]L,UNIT]L,MSGADR]L,PRLINK]L,P7-P6]

ГДЕ NUM - УНИКАЛЬНЫЙ НОМЕР ОШИБКИ В ПРЕДЕЛАХ ТЕКУЩЕГО ПОДТЕСТА. ПО УМОЛЧАНИЮ АВТОМАТИЧЕСКИ УСТАНАВЛИВАЕТСЯ СЛЕДУЮЩИЙ ПО ПОРЯДКУ НОМЕР ОШИБКИ;

UNIT - ЛОГИЧЕСКИЙ НОМЕР ПРОВЕРЯЕМОГО УСТРОЙСТВА;

MSGADR - АДРЕС СТРОКИ СИМВОЛОВ С НАЧАЛЬНЫМ БАЙТОМ ДЛИНЫ (ДИРЕКТИВА МАКРОАССЕМБЛЕРА .ASCII). В ЭТОЙ СТРОКЕ ДОЛЖНО СОДЕРЖАТЬСЯ КРАТКОЕ ОПИСАНИЕ ОШИБКИ, КОТОРОЕ ВКЛЮЧАЕТСЯ В ЗАГОЛОВОК СООБЩЕНИЯ ОБ ОШИБКЕ;

13.00078-01 33 01

PRLINK - АДРЕС ПОДПРОГРАММЫ ПЕЧАТИ ДОПОЛНИТЕЛЬНОЙ ИНФОРМАЦИИ ОБ ОБНАРУЖЕННОЙ ОШИБКЕ. ДАННАЯ ПОДПРОГРАММА ДОЛЖНА РАЗМЕЩАТЬСЯ МЕЖДУ МАКРОКОМАНДАМИ #DS\_BGNMESSAGE И #DS\_ENDMESSAGE. ПРОГРАММА ВЫПОЛНЯЕТСЯ НЕЗАВИСИМО ОТ СОСТОЯНИЯ ФЛАГА ОСТАНОВА ПО ОШИБКЕ;

P1-P6 - ОТ ОДНОГО ДО ШЕСТИ ДОПОЛНИТЕЛЬНЫХ ПАРАМЕТРОВ, КОТОРЫЕ МОГУТ ПЕРЕДАВАТЬСЯ ПОДПРОГРАММЕ ПЕЧАТИ СООБЩЕНИЯ ОБ ОШИБКЕ.

МАКРОКОМАНДУ #DS\_ERRSOFT\_X СЛЕДУЕТ ИСПОЛЬЗОВАТЬ В ТЕХ СЛУЧАЯХ, КОГДА ТЕСТ СНАЧАЛА ОБНАРУЖИВАЕТ СБОИ, А ЗАТЕМ ПРИ ПОВТОРНОЙ ПОПЫТКЕ ОН ВЫПОЛНЯЕТСЯ ПРАВИЛЬНО.

#### 7.9.2. МАКРОКОМАНДЫ ПЕЧАТИ ИНФОРМАЦИИ

ДЛЯ ПЕЧАТИ ОСНОВНОЙ И РАСШИРЕННОЙ ИНФОРМАЦИИ ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ ИСПОЛЬЗУЮТ МАКРОКОМАНДЫ #DS\_PRINTB\_X, #DS\_PRINTX\_X, #DS\_PRINTS\_X И #DS\_PRINTF\_X.

КАЖДАЯ ИЗ ЭТИХ МАКРОКОМАНД ВЫЗЫВАЕТ ПОДПРОГРАММУ РАСШИРЕННОЙ ПЕЧАТИ СУПЕРВИЗОРА ДЛЯ ПРОВЕРКИ ФЛАГОВ ЗАПРЕТА СООБЩЕНИЙ ОБ ОШИБКАХ ДО НАЧАЛА ПЕЧАТИ СООБЩЕНИЯ.

МАКРОКОМАНДА #DS\_PRINTB\_X ПРИВОДИТ К ПРОВЕРКЕ ФЛАГОВ IE1 И IE2, #DS\_PRINTX\_X - IE1, IE2 И IE3, #DS\_PRINTS\_X - IE5. ПРИ ИСПОЛЬЗОВАНИИ МАКРОКОМАНДЫ #DS\_PRINTF\_X ФЛАГИ НЕ ПРОВЕРЯЮТСЯ.

ПОДПРОГРАММА РАСШИРЕННОЙ ПЕЧАТИ ИСПОЛЬЗУЕТ ФОРМАТИРОВАННЫЙ ВЫВОД В КОДЕ КОИ-8 С ПРИМЕЧАНИЕМ ДИРЕКТИВ FAO, КОМПИЛИРУЕТ СТРОКУ, А ЗАТЕМ РАСПЕЧАТЫВАЕТ ЕЕ.

МАКРОКОМАНДЫ #DS\_PRINTB\_X И #DS\_PRINTX\_X НЕОБХОДИМО ИСПОЛЬЗОВАТЬ В ПОДПРОГРАММЕ ПЕЧАТИ (АРГУМЕНТ PRLINK В МАК-

19.00078-01 33 01

ПОКОМАНДЕ #DS\_ERR...\_X), А НЕ В ТЕСТОВЫХ ПОДПРОГРАММАХ.

РАССМОТРИМ ПРИМЕР ИСПОЛЬЗОВАНИЯ МАКРОКОМАНД ПЕЧАТИ СООБЩЕНИЙ.

ПРИМЕР.

```
; ОТДЕЛ ДАННЫХ ПРОГРАММЫ:
MSG5::      .ASCII \СООБЩЕНИЕ 5\
MSG6::      .ASCII \СООБЩЕНИЕ 6\
; ПОДПРОГРАММА ПЕЧАТИ, ВЫЗЫВАЮЩАЯ МАКРОКОМАНДУ
; #DS_PRINTB_X
PRINT_ROUTINE::
#DS_BGNMESSAGE REGMASK=<R2,R3,R4>
MOVL ERRP_P1(AP), R2      ; ЗАПОМИНАНИЕ АДРЕСА
                          ; ОСНОВНОГО СООБЩЕНИЯ
#DS_PRINTB_S FORMAT=(R2) ; ПЕЧАТЬ ОСНОВНОГО
                          ; СООБЩЕНИЯ
.
.
.
#DS_ENDMESSAGE
; ТЕСТОВАЯ ПОДПРОГРАММА, ВЫЗЫВАЮЩАЯ ПОДПРОГРАММУ ПЕЧАТИ
#DS_BGN SUB
.
.
.
#DS_BNERROR 20S          ; ИМЕЕТСЯ ОШИБКА?
#DS_ERRHARD_S-          ; ДА
UNIT=LOG_UNIT,-        ; ПРОВЕРЯЕМОЕ УСТРОЙСТВО
MSGADR=MSG5,-          ; ТЕКСТ ЗАГОЛОВКА
                          ; СООБЩЕНИЯ
PRLINK=PRINT_ROUTINE,- ; ПОДПРОГРАММА ПЕЧАТИ
                          ; ОСНОВНОЙ ИНФОРМАЦИИ
                          ; ОБ ОШИБКЕ
P1=MSG6                 ; СООБЩЕНИЕ ОСНОВНОЙ
                          ; ПОДПРОГРАММЫ ПЕЧАТИ
```

В ПРИМЕРЕ #DS\_ERRHARD\_S ВЫЗЫВАЕТ ПОДПРОГРАММУ СУПЕРВИЗОРА ДЛЯ ФОРМИРОВАНИЯ ЗАГОЛОВКА СООБЩЕНИЯ ИЗ ТРЕХ СТРОК. АРГУМЕНТ MSGADR УКАЗЫВАЕТ СТРОКУ КОДА КОИ-8 (АДРЕС MSG5), КОТОРАЯ СОСТАВЛЯЕТ ЧАСТЬ ЗАГОЛОВКА СООБЩЕНИЯ. АРГУМЕНТ PRLINK УКАЗЫВАЕТ ПОДПРОГРАММУ ПЕЧАТИ PRINT\_ROUTINE, УПРАВЛЕНИЕ КОТОРОЙ ПЕРЕДАЕТСЯ ОТ ПОДПРОГРАММЫ СУПЕРВИЗОРА. ПОДПРОГРАММА ПЕЧАТИ ЗАПОМИНАЕТ В РЕГИСТРЕ R2 АРГУМЕНТ P1 МАКРОКОМАНДЫ #DS\_ERRHARD\_S. ЗАТЕМ МАКРОКОМАНДА #DS\_PRINTB\_S В ПОДПРОГРАММЕ ПЕЧАТИ ВЫЗЫВАЕТ РАСПЕЧАТКУ СООБЩЕНИЯ MSG6.

ЧАСТЬ ТРЕБУЕТСЯ ПЕЧАТАТЬ СООБЩЕНИЯ, ЧАСТЬ ИНФОРМАЦИИ В

13.00078-01 33 01

КОТОРЫХ МЕНЯЕТСЯ (НАПРИМЕР, СОДЕРЖИМОЕ РЕГИСТРА ИЛИ МНЕМОНИКА БИТА С ОШИБКОЙ). В ТАКИХ СЛУЧАЯХ ДЛЯ ПОЛУЧЕНИЯ ПОЛНОГО СООБЩЕНИЯ НУЖНО ИСПОЛЬЗОВАТЬ УПРАВЛЯЮЩУЮ СТРОКУ, СОДЕРЖАЩУЮ ДИРЕКТИВЫ FAO. СПИСОК ДИРЕКТИВ FAO ПРИВЕДЕН В ТАБЛ.1.

#### 7.9.2.1. #DS\_PRINTB\_X - ПЕЧАТЬ ОСНОВНОЙ ИНФОРМАЦИИ ОБ ОШИБКЕ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_PRINTB\_X FORMAT[,PO],P1]...

ГДЕ FORMAT - АДРЕС УПРАВЛЯЮЩЕЙ СТРОКИ, СОСТОЯЩЕЙ ИЗ ФИКСИРОВАННОГО ТЕКСТА СТРОКИ ВЫВОДА И ДИРЕКТИВ ПРЕОБРАЗОВАНИЯ;

PO, P1... - ПАРАМЕТРЫ ДИРЕКТИВ УПРАВЛЯЮЩЕЙ СТРОКИ, СОДЕРЖАЩИЕСЯ В ДЛИННЫХ СЛОВАХ. ЧИСЛО ПАРАМЕТРОВ - НЕ БОЛЕЕ 16.

МАКРОКОМАНДУ #DS\_PRINTB\_X СЛЕДУЕТ ИСПОЛЬЗОВАТЬ В ПОДПРОГРАММЕ ПЕЧАТИ (А НЕ В ТЕСТОВЫХ ПОДПРОГРАММАХ) ДЛЯ ПЕЧАТИ ОСНОВНОЙ ИНФОРМАЦИИ ОБ ОШИБКЕ (НАПРИМЕР, ПОЯСНИТЕЛЬНОЕ СООБЩЕНИЕ, ОЖИДАЕМЫЕ ДАННЫЕ, ПОЛУЧЕННЫЕ ДАННЫЕ, РЕЗУЛЬТАТ СЛОЖЕНИЯ ПО МОДУЛЮ 2 ОЖИДАЕМЫХ И ПОЛУЧЕННЫХ ДАННЫХ). ОБЫЧНО ОДНА МАКРОКОМАНДА #DS\_PRINTB\_X ИСПОЛЬЗУЕТСЯ ДЛЯ РАСПЕЧАТКИ ОДНОЙ СТРОКИ, ПОЭТОМУ ДЛЯ ПЕЧАТИ ВСЕЙ ОСНОВНОЙ ИНФОРМАЦИИ МОЖЕТ ПОТРЕБОВАТЬСЯ НЕСКОЛЬКО ЭТИХ МАКРОКОМАНД.

ПРИ УСТАНОВКЕ ФЛАГОВ IE1 ИЛИ IE2 РАСПЕЧАТКА ИНФОРМАЦИИ МАКРОКОМАНДОЙ #DS\_PRINTB\_X ПОДАВЛЯЕТСЯ.

МАКРОКОМАНДУ #DS\_PRINTB\_X СЛЕДУЕТ ИСПОЛЬЗОВАТЬ В ПОДПРОГРАММАХ ПЕЧАТИ, ВЫЗЫВАЕМЫХ ПОСРЕДСТВОМ ОДНОЙ ИЗ МАКРОКОМАНД #DS\_ERR...\_X.

13.00078-01 33 01

### 7.9.2.2. #DS\_PRINTX\_X - ПЕЧАТЬ РАСШИРЕННОЙ ИНФОРМАЦИИ ОБ ОШИБКЕ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_PRINTX\_X FORMAT[,P0][,P1]...I

ГДЕ FORMAT - АДРЕС УПРАВЛЯЮЩЕЙ СТРОКИ, СОСТОЯЩЕЙ ИЗ  
ФИКСИРОВАННОГО ТЕКСТА СТРОКИ ВЫВОДА И ДИРЕКТИВ ПРЕОБ-  
РАЗОВАНИЙ;

P1, P2... - ПАРАМЕТРЫ ДИРЕКТИВ УПРАВЛЯЮЩЕЙ СТРОКИ,  
СОДЕРЖАЩИЕСЯ В ДЛИННЫХ СЛОВАХ. ЧИСЛО ПАРАМЕТРОВ - НЕ  
БОЛЕЕ 16.

МАКРОКОМАНДУ #DS\_PRINTX\_X СЛЕДУЕТ ИСПОЛЬЗОВАТЬ В ПОДП-  
РОГРАММЕ ПЕЧАТИ (А НЕ В ТЕСТОВЫХ ПОДПРОГРАММАХ) ДЛЯ ПЕЧАТИ  
ДОПОЛНИТЕЛЬНОЙ ИНФОРМАЦИИ ОБ ОШИБКЕ (НАПРИМЕР, АДРЕСА ОШИ-  
БОК, СОДЕРЖИМОЕ РЕГИСТРОВ УСТРОЙСТВА И КАНАЛА, ДОПОЛНИТЕЛЬ-  
НЫЕ ПОЯСНЕНИЯ).

ПРИ УСТАНОВКЕ ФЛАГОВ IE1, IE2 ИЛИ IE3 РАСПЕЧАТКА  
ИНФОРМАЦИИ МАКРОКОМАНДОЙ #DS\_PRINTX\_X ПОДАВЛЯЕТСЯ.

МАКРОКОМАНДУ #DS\_PRINTX\_X СЛЕДУЕТ ИСПОЛЬЗОВАТЬ В ПОДП-  
РОГРАММАХ РАСПЕЧАТКИ, ВЫЗЫВАЕМЫХ ПОСРЕДСТВОМ ОДНОЙ ИЗ МАК-  
РОКОМАНД #DS\_ERR...\_X.

### 7.9.2.3. #DS\_PRINTF\_X - ТРИНУДИТЕЛЬНАЯ ПЕЧАТЬ СООБЩЕНИЯ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_PRINTF\_X FORMAT[,P0][,P1]...I

ГДЕ FORMAT - АДРЕС УПРАВЛЯЮЩЕЙ СТРОКИ, СОСТОЯЩЕЙ ИЗ  
ФИКСИРОВАННОГО ТЕКСТА СТРОКИ ВЫВОДА И ДИРЕКТИВ ПРЕОБ-



13.00078-01 33 01

РАЗОВАНИЯ:

P1, P2... - ПАРАМЕТРЫ ДИРЕКТИВ УПРАВЛЯЮЩЕЙ СТРОКИ, СОДЕРЖАЩИЕСЯ В ДЛИННЫХ СЛОВАХ. ЧИСЛО ПАРАМЕТРОВ - НЕ БОЛЕЕ 16.

МАКРОКОМАНДУ #DS\_PRINTF\_X СЛЕДУЕТ ИСПОЛЬЗОВАТЬ ДЛЯ РАСПЕЧАТКИ СООБЩЕНИЙ, КОТОРЫЕ ПРЕДУПРЕЖДАЮТ ОПЕРАТОРА О ЗНАЧИТЕЛЬНЫХ СОБЫТИЯХ И СООБЩАЮТ ЕМУ О НЕОБХОДИМОСТИ ДЕЙСТВИИ С ЕГО СТОРОНЫ. ДАННАЯ МАКРОКОМАНДА ПРОИЗВОДИТ РАСПЕЧАТКУ СООБЩЕНИЙ НЕЗАВИСИМО ОТ СОСТОЯНИЯ ФЛАГОВ ЗАПРЕЩЕНИЯ СООБЩЕНИИ ОБ ОШИБКАХ.

МАКРОКОМАНДА #DS\_PRINTF\_X ИСПОЛЬЗУЕТСЯ В ПОДПРОГРАММАХ РАСПЕЧАТКИ, ВЫЗЫВАЕМЫХ ПОСРЕДСТВОМ ОДНОЙ ИЗ МАКРОКОМАНД #DS\_ERR...\_X.

#### 7.9.2.4. #DS\_PRINTS\_X - ПЕЧАТЬ ИТОГОВОГО ОТЧЕТА

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_PRINTS\_X FORMAT[,P0][,P1]...

ГДЕ FORMAT - АДРЕС УПРАВЛЯЮЩЕЙ СТРОКИ, СОСТОЯЩЕЙ ИЗ ФИКСИРОВАННОГО ТЕКСТА СТРОКИ ВЫВОДА И ДИРЕКТИВ ПРЕОБРАЗОВАНИЯ:

P1, P2... - ПАРАМЕТРЫ ДИРЕКТИВ УПРАВЛЯЮЩИХ СТРОК, СОДЕРЖАЩИЕСЯ В ДЛИННЫХ СЛОВАХ. ЧИСЛО ПАРАМЕТРОВ - НЕ БОЛЕЕ 16.

МАКРОКОМАНДУ #DS\_PRINTS\_X СЛЕДУЕТ ИСПОЛЬЗОВАТЬ ТОЛЬКО В ПОДПРОГРАММЕ ЗАКЛЮЧИТЕЛЬНОГО (ИТОГОВОГО) ОТЧЕТА ДЛЯ РАСПЕЧАТКИ ЗАКЛЮЧИТЕЛЬНОГО СООБЩЕНИЯ. ПРИ УСТАНОВКЕ УПРАВЛЯЮЩЕГО ФЛАГА IES ПЕЧАТЬ СООБЩЕНИЙ ПОДАВЛЯЕТСЯ.

13.00078-01 33 01

### 7.9.3. ДИРЕКТИВЫ ПРЕОБРАЗОВАНИЯ FAO

ДИРЕКТИВЫ FAO ИСПОЛЬЗУЮТСЯ ДЛЯ ВНЕСЕНИЯ В СТРОКУ ПЕЧАТИ (СВХОДНУЮ СТРОКУ) ДВОИЧНЫХ ВЕЛИЧИН, ПРЕДСТАВЛЕННЫХ В ИХ ДЕСЯТИЧНОМ, ДЕСЯТИЧНО-ШЕСТИНАДЦАТЕРИЧНОМ ИЛИ ВОСЬМЕРИЧНОМ ЭКВИВАLENTE В КОДЕ КОИ-8. ДИРЕКТИВЫ FAO ПОЗВОЛЯЮТ ТАКЖЕ ВНОСИТЬ ЦЕПОЧКИ СИМВОЛОВ В СТРОКУ ПЕЧАТИ.

ПРИ ОБРАБОТКЕ СТРОКИ, СОДЕРЖАЩЕЙ ДИРЕКТИВУ ПРЕОБРАЗОВАНИЯ FAO, КАЖДАЯ ДИРЕКТИВА ЗАМЕЩАЕТСЯ СИМВОЛАМИ В ЗАВИСИМОСТИ ОТ СОДЕРЖИМОГО СООТВЕТСТВУЮЩЕГО ЕЙ (ДИРЕКТИВЕ) ПАРАМЕТРА. ПАРАМЕТРЫ ДИРЕКТИВЫ УКАЗЫВАЮТСЯ В ДЛИННЫХ СЛОВАХ. СИМВОЛЫ УПРАВЛЯЮЩЕЙ СТРОКИ, НЕ ЯВЛЯЮЩИЕСЯ ЧАСТЬЮ ДИРЕКТИВЫ FAO, ОСТАЮТСЯ БЕЗ ИЗМЕНЕНИЯ.

ДИРЕКТИВА FAO ИМЕЕТ СЛЕДУЮЩИИ ФОРМАТ:

!DD

ГДЕ ! (ВОСКЛИЦАТЕЛЬНЫЙ ЗНАК) УКАЗЫВАЕТ, ЧТО СЛЕДУЮЩИИ ЗА НИМ СИМВОЛ ИЛИ СИМВОЛЫ ДОЛЖНЫ ИНТЕРПРЕТИРОВАТЬСЯ КАК ДИРЕКТИВА FAO;

DD - КОД ИЗ ОДНОГО ИЛИ ДВУХ СИМВОЛОВ, УКАЗЫВАЮЩИИ ДЕЙСТВИЕ, КОТОРОЕ ДОЛЖНО ВЫПОЛНЯТЬСЯ ПО ДИРЕКТИВЕ FAO. ДИРЕКТИВЫ ДОЛЖНЫ УКАЗЫВАТЬСЯ С ИСПОЛЬЗОВАНИЕМ БУКВ ВЕРХНЕГО РЕГИСТРА. ДОПОЛНИТЕЛЬНО ДИРЕКТИВЫ FAO МОГУТ ЗАДАВАТЬ ЧИСЛО ПОВТОРЕНИИ И (ИЛИ) ДЛИНУ ПОЛЯ, ГЕНЕРИРУЕМОГО ПО ДИРЕКТИВЕ FAO.

ПРИ УКАЗАНИИ ЧИСЛА ПОВТОРЕНИИ ДИРЕКТИВА FAO ЗАПИСЫВАЕТСЯ СЛЕДУЮЩИМ ОБРАЗОМ:

!N(DD)

19.00078-01 33 01

ГДЕ N — ДЕСЯТИЧНОЕ ЧИСЛО, УКАЗЫВАЮЩЕЕ СКОЛЬКО РАЗ  
ПОВТОРЯЕТСЯ ДАННАЯ ДИРЕКТИВА.

ДЛИНА ВЫХОДНОГО ПОЛЯ ЗАДАЕТСЯ ТАК:

!MDD

ГДЕ M — ДЕСЯТИЧНОЕ ЧИСЛО, УКАЗЫВАЮЩЕЕ РАЗМЕР ПОЛЯ,  
ГЕНЕРИРУЕМОГО ПО ДИРЕКТИВЕ FAD.

ЕСЛИ ДИРЕКТИВА FAD ДОЛЖНА СОДЕРЖАТЬ ОДНОВРЕМЕННО КОЛИ-  
ЧЕСТВО ПОВТОРЕНИЙ И ДЛИНУ ПОЛЯ, ТО ИСПОЛЬЗУЕТСЯ ФОРМАТ:

!N(MDD)

В ТАБЛ.1 ПРИВЕДЕНЫ ДИРЕКТИВЫ FAD И ПЕРЕЧИСЛЕННЫ ПАРА-  
МЕТРЫ, ТРЕБУЮЩИЕСЯ КАЖДОЙ ДИРЕКТИВЕ.

ТАБЛИЦА 1

ДИРЕКТИВЫ ПРЕОБРАЗОВАНИЯ FAD

ДИРЕКТИВА	!	ФУНКЦИЯ	!	ПАРАМЕТРЫ
ПОДСТАНОВКА В СТРОКУ СИМВОЛОВ				
!AC	!	ВНЕСЕНИЕ СТРОКИ КОДА КОИ-8 С НА- ЧАЛЬНЫМ БАЙТОМ ДЛИНЫ	!	АДРЕС СТРОКИ; ПЕРВЫЙ БАЙТ ДОЛЖЕН СОДЕР- ЖАТЬ ДЛИНУ СТРОКИ
!AD	!	ВНЕСЕНИЕ СТРОКИ КОДА КОИ-8	!	1) ДЛИНА СТРОКИ; 2) АДРЕС СТРОКИ
!AS	!	ВНЕСЕНИЕ СТРОКИ КОДА КОИ-8	!	АДРЕС ЧЕТЫРЕХСЛОВ- НОГО ОПИСАТЕЛЯ, УКА- ЗЫВАЮЩЕГО СТРОКУ
ЧИСЛЕННОЕ ПРЕОБРАЗОВАНИЕ (С ВЕДУЩИМИ НУЛЯМИ)				
!OB	!	ПРЕОБРАЗОВАНИЕ БАЙТА В ВОСЬМЕРИ- ЧНУЮ ФОРМУ	!	
!OW	!	ПРЕОБРАЗОВАНИЕ СЛОВА В ВОСЬМЕРИ-	!	

ПРОДОЛЖЕНИЕ ТАБЛ. 1

ДИРЕКТИВА	!	ФУНКЦИЯ	!	ПАРАМЕТРЫ
	!	ЧНУЮ ФОРМУ	!	
!OL	!	ПРЕОБРАЗОВАНИЕ ДЛИННОГО СЛОВА В ВОСЬМЕРИЧНУЮ ФОР- МУ	!	ВЕЛИЧИНА, КОТОРАЯ ДОЛЖНА БЫТЬ ПРЕОБРА- ЗОВАНА В ПРЕДСТАВЛЕ- НИЕ КОДА КОИ-8. ПРИ
!XB	!	ПРЕОБРАЗОВАНИЕ БАЙТА В ШЕСТНАД- ЦАТЕРИЧНУЮ ФОРМУ	!	ПРЕОБРАЗОВАНИИ БАЙТА ИЛИ СЛОВА ИСПОЛЬЗУ- ЕТСЯ ТОЛЬКО МЛАДШИИ
!XW	!	ПРЕОБРАЗОВАНИЕ СЛОВА В ШЕСТНАД- ЦАТЕРИЧНУЮ ФОРМУ	!	БАИТ ИЛИ СЛОВО ДЛИН- НОГО СЛОВА (ПАРАМЕТ- РА)
!XL	!	ПРЕОБРАЗОВАНИЕ ДЛИННОГО СЛОВА В ШЕСТНАДЦАТЕРИЧНУЮ ФОРМУ	!	
!ZB	!	ПРЕОБРАЗОВАНИЕ ДЕСЯТИЧНОГО БАЙТА БЕЗ ЗНАКА	!	
!ZW	!	ПРЕОБРАЗОВАНИЕ ДЕСЯТИЧНОГО СЛОВА БЕЗ ЗНАКА	!	
!ZL	!	ПРЕОБРАЗОВАНИЕ ДЕСЯТИЧНОГО ДЛИН- НОГО СЛОВА БЕЗ ЗНАКА	!	

ЧИСЛОВЫЕ ПРЕОБРАЗОВАНИЕ (С ПРОБЕЛАМИ ВМЕСТО ВЕДУЩИХ НУЛЕЙ)

!UB	!	ПРЕОБРАЗОВАНИЕ ДЕСЯТИЧНОГО БАЙТА БЕЗ ЗНАКА	!	
!UW	!	ПРЕОБРАЗОВАНИЕ ДЕСЯТИЧНОГО СЛОВА БЕЗ ЗНАКА	!	
!UL	!	ПРЕОБРАЗОВАНИЕ ДЕСЯТИЧНОГО ДЛИН- НОГО СЛОВА БЕЗ ЗНАКА	!	ВЕЛИЧИНА, КОТОРАЯ ДОЛЖНА БЫТЬ ПРЕОБРА- ЗОВАНА В ПРЕДСТАВЛЕ- НИЕ КОДА КОИ-8. ПРИ
!SB	!	ПРЕОБРАЗОВАНИЕ ДЕСЯТИЧНОГО БАЙТА СО ЗНАКОМ	!	ПРЕОБРАЗОВАНИИ БАЙТА ИЛИ СЛОВА ИСПОЛЬЗУЕ- ТСЯ ТОЛЬКО МЛАДШИИ
!SW	!	ПРЕОБРАЗОВАНИЕ ДЕСЯТИЧНОГО СЛОВА	!	БАИТ ИЛИ СЛОВО ДЛИН- НОГО СЛОВА (ПАРАМЕТ- РА)

13.00078-01 33 01

ПРОДОЛЖЕНИЕ ТАБЛ. 1

ДИРЕКТИВА	ФУНКЦИЯ	ПАРАМЕТРЫ
!SL	СО ЗНАКОМ ПРЕОБРАЗОВАНИЕ ДЛИННОГО ДЕСЯТИЧ- НОГО СЛОВА СО ЗНАКОМ	
ФОРМАТИРОВАНИЕ ВЫХОДНОЙ СТРОКИ		
! /	ПЕРЕХОД НА НОВУЮ СТРОКУ	
! _	ВВЕДЕНИЕ ТАБУЛЯ- ЦИИ	ПАРАМЕТРЫ ОТСУТСТВУЮТ
! !	ВВЕДЕНИЕ ВОСКЛИ- ЦАТЕЛЬНОГО ЗНАКА	
! %T	ВНЕСЕНИЕ СИСТЕМ- НОГО ВРЕМЕНИ	АДРЕС ЧЕТЫРЬНОГО СЛОВА, СОДЕРЖИМОЕ КОТОРОГО ДОЛЖНО БЫТЬ ПРЕОБРАЗОВАНО В КОД КОИ-8. ЕСЛИ УКАЗАН НУЛЬ, ТО ИСПОЛЬЗУЕТ- СЯ ТЕКУЩЕЕ СИСТЕМНОЕ ВРЕМЯ
! %D	ВНЕСЕНИЕ СИСТЕМ- НОЙ ДАТЫ И ВРЕМЕ- НИ	
! -	ПОВТОРНО ИСПОЛЬ- ЗОВАТЬ ПОСЛЕДНИЙ ПАРАМЕТР В СПИСКЕ	НЕТ
! +	ПРОПУСТИТЬ СЛЕДУ- ЮЩИЙ ПАРАМЕТР В СПИСКЕ	НЕТ
! N * C	ПОВТОРИТЬ СИМВОЛ C В ВЫХОДНОЙ СТРОКЕ N РАЗ	НЕТ

7.10. МАКРОКОМАНДЫ ОПРЕДЕЛЕНИЯ СИМВОЛИЧЕСКИХ  
ИМЕН

ЭТИ МАКРОКОМАНДЫ ОПРЕДЕЛЯЮТ СИМВОЛИЧЕСКИЕ ИМЕНА, КОТО-  
РЫЕ ЯВЛЯЮТСЯ ОБЩИМИ ДЛЯ ДИАГНОСТИЧЕСКОЙ СИСТЕМЫ СМ-1700.  
ОДНА ЧАСТЬ СИМВОЛИЧЕСКИХ ИМЕН, ОПРЕДЕЛЯЕМЫХ МАКРОКОМАНДАМИ,

19.00078-01 33 01

являются именами ключевых слов для других макрокоманд. Эти имена нужны для интерфейса между супервизором и программой. В составе диагностической макробиблиотеки имеются также макрокоманды, определяющие символические имена битов, сдвижений таблицы и возвращаемых подпрограммами кодов состояний. Макрокоманды определения символических имен используются в основном в отделе описании головного модуля. Если наборы символических имен относятся к конкретному модулю, то они должны быть определены в этом модуле.

#### 7.10.1. RDS\_BITDEF - ОПРЕДЕЛЕНИЕ МНЕМОНИЧЕСКИХ ОБОЗНАЧЕНИИ БИТОВОЙ МАСКИ

ФОРМАТ МАКРОКОМАНДЫ:

```
RDS_BITDEF LGBLJ
```

ГДЕ GBL - GLOBAL (ГЛОБАЛЬНЫЕ ИМЕНА) ИЛИ LOCAL

(ЛОКАЛЬНЫЕ ИМЕНА). ПО УМОЛЧАНИЮ ГЕНЕРИРУЮТСЯ ЛОКАЛЬНЫЕ ИМЕНА.

МАКРОКОМАНДА ГЕНЕРИРУЕТ 32 СИМВОЛИЧЕСКИХ ИМЕНИ: BIT0, BIT1, ..., BIT31. КАЖДОМУ ИЗ ЭТАХ ИМЕН ПРИСВАИВАЕТСЯ МАСКА, В КОТОРОЙ ОДИН БИТ УСТАНОВЛЕН, А ОСТАЛЬНЫЕ РАВНЫ НУЛЮ. НОМЕР УСТАНОВЛЕННОГО БИТА В МАСКЕ СОВПАДАЕТ С ЧИСЛОМ, УКАЗАННЫМ ПОСЛЕ СИМВОЛОВ BIT.

СИМВОЛИЧЕСКИЕ ИМЕНА BIT0 - BIT31 ПОЗВОЛЯЮТ ПРОГРАММИСТУ УКАЗЫВАТЬ ОПРЕДЕЛЕННЫМ БИТ ЯЧЕЙКИ БЕЗ НЕОБХОДИМОСТИ НАПИСАНИЯ В ПРОГРАММЕ СООТВЕТСТВУЮЩЕЙ МАСКИ.

19.00078-01 33 01'

7.10.2. #DS\_DSADEF - ОПРЕДЕЛЕНИЕ СИМВОЛИЧЕСКИХ  
ИМЕН ФЛАГОВ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_DSADEF [G|L]

ГДЕ G|L - GLOBAL (ГЛОБАЛЬНЫЕ ИМЕНА) ИЛИ LOCAL

(ЛОКАЛЬНЫЕ ИМЕНА). ПО УМОЛЧАНИЮ ГЕНЕРИРУЮТСЯ ЛОКАЛЬНЫЕ  
ИМЕНА.

МАКРОКОМАНДА ГЕНЕРИРУЕТ СЛЕДУЮЩИЕ СИМВОЛИЧЕСКИЕ ИМЕНА,  
ОПРЕДЕЛЯЮЩИЕ ОБЛАСТИ ФЛАГОВ ИНТЕРПРЕТАТОРА КОМАНДНОЙ СТРОКИ  
И КОМАНД:

DSARGL_FLAGS	ДЛИННОЕ СЛОВО, СОДЕРЖАЩЕЕ УКАЗАННЫЕ НИЖЕ БИТЫ ФЛАГОВ;
ОЛАГИ	
DSARM_HALTD	ОСТАНОВ ПРИ ОБНАРУЖЕНИИ ОШИБКИ;
DSARM_HALTII	ОСТАНОВ ПРИ ЛОКАЛИЗАЦИИ ОШИБКИ;
DSARM_LOOP	ЗАЦИКЛИВАНИЕ ПО ФЛАГУ ОШИБКИ;
DSARM_BELL	ЗВУКОВОЙ СИГНАЛ ПРИ ОШИБКЕ;
DSARM_IES1	ЗАПРЕТ ВСЕХ ОТЧЕТОВ ОБ ОШИБКАХ;
DSARM_IES2	ЗАПРЕТ ОСНОВНЫХ ОТЧЕТОВ ОБ ОШИБКАХ;
DSARM_IES3	ЗАПРЕТ РАСШИРЕННЫХ ОТЧЕТОВ ОБ ОШИБКАХ;
DSARM_IES4	ЗАПРЕТ ИТоговых ОТЧЕТОВ;
DSARM_QUICK	БЫСТРАЯ ПРОВЕРКА;
DSARM_TRACE	ОТСЛЕЖИВАНИЕ ТЕСТОВ;
DSARM_OPER	ПРИСУТСТВИЕ ОПЕРАТОРА;
DSARM_NORPT	ЗАПРЕТ ЛЮБОГО ВЫВОДА НА ТЕРМИНАЛ;

13.00078-01 33 01

DSAPM\_PASSO                    ПРОХОД 0.

ОПРЕДЕЛЕНИЯ ОБЛАСТИ КОМАНД

DSAPGL\_PASSES                  ЧИСЛО ПРОХОДОВ;  
DSAPGL\_UNITS                  ЧИСЛО ПРОВЕРЯЕМЫХ УСТРОЙСТВ;  
DSAPGL\_SECTNO                 НОМЕР СЕКЦИИ;  
DSAPGL\_CRUTYP                 КОД ТИПА ЦЕНТРАЛЬНОГО  
                                 ПРОЦЕССОРА.

7.10.3. #DS\_DSDEF - ОПРЕДЕЛЕНИЕ КОДОВ СОСТОЯНИЯ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_DSDEF [GBL]

ГДЕ GBL - GLOBAL (ГЛОБАЛЬНЫЕ ИМЕНА) ИЛИ LOCAL

(ЛОКАЛЬНЫЕ ИМЕНА). ПО УМОЛЧАНИЮ ГЕНЕРИРУЮТСЯ ЛОКАЛЬНЫЕ  
ИМЕНА.

МАКРОКОМАНДА ГЕНЕРИРУЕТ СИМВОЛИЧЕСКИЕ ИМЕНА, КОТОРЫЕ  
УКАЗЫВАЮТ ВЫДАВАЕМЫЕ СОСТОЯНИЯ. СЛУЖЕБНЫЕ МАКРОКОМАНДЫ  
СУПЕРВИЗОРА ВЫЗЫВАЮТ СЛУЖЕБНЫЕ ПОДПРОГРАММЫ СУПЕРВИЗОРА,  
КОТОРЫЕ ВОЗВРАЩАЮТ В РЕГИСТРЕ R0 ОДИН ИЗ КОДОВ, ПРЕДСТАВ-  
ЛЯЕМЫХ ЭТИМИ СИМВОЛИЧЕСКИМИ ИМЕНАМИ, ДЛЯ УКАЗАНИЯ ВЫДАВАЕ-  
МОГО СОСТОЯНИЯ.

МАКРОКОМАНДА ФОРМИРУЕТ СЛЕДУЮЩИЕ СИМВОЛИЧЕСКИЕ ИМЕНА:

DSE\_WARNING                    ПРЕДУПРЕЖДЕНИЕ;  
DSE\_NORMAL                    НОРМАЛЬНОЕ ВЫПОЛНЕНИЕ;  
DSE\_ERROR                      ОШИБКА;  
DSE\_SEVERE                     СЕРЬЕЗНАЯ ОШИБКА;  
DSE\_OVERFLOW                  ПЕРЕТОЛНЕНИЕ;  
DSE\_NULLSTR                    ПУСТАЯ СТРОКА;



13.00078-01 33 01

DSM_PROGERR	ПРОГРАММНАЯ ОШИБКА;
DSM_TRUNCATE	УСЕЧЕНИЕ ДАННЫХ;
DSM_NOTDOON	ФУНКЦИЯ ИЛИ ОПЕРАЦИЯ НЕ ВЫПОЛНЕНА;
DSM_IVVECT	НЕПРАВИЛЬНЫЙ ВЕКТОР;
DSM_IVADDR	НЕПРАВИЛЬНЫЙ АДРЕС;
DSM_VASFUL	ВИРТУАЛЬНОЕ АДРЕСНОЕ ПРОСТРАНСТВО ЗАПОЛНЕНО;
DSM_INSMEM	НЕДОСТАТОЧНО ПАМЯТИ;
DSM_MM0FF	ВЫКЛЮЧЕНО УПРАВЛЕНИЕ ПАМЯТЬЮ;
DSM_IHWE	ИСХОДНАЯ ОШИБКА ОБОРУДОВАНИЯ;
DSM_FHWE	КОНЕЧНАЯ ОШИБКА ОБОРУДОВАНИЯ;
DSM_LOGIC	ОШИБКА ИНТЕРФЕЙСА;
DSM_ILLRAGCNT	НЕДОПУСТИМОЕ КОЛИЧЕСТВО СТРАНИЦ;
DSM_FRAGBUF	ПРИ ОСВОБОЖДЕНИИ БУФЕР ОКАЗАЛСЯ ФРАГМЕНТИРОВАННЫМ;
DSM_MCHK	МАШИННЫЙ СБОЙ;
DSM_KRNLSLK	НЕДОПУСТИМЫЙ СТЕК ЯДРА;
DSM_POWER	СБОЙ ПИТАНИЯ;
DSM_TRANSL	НЕПРАВИЛЬНАЯ ТРАНСЛЯЦИЯ;
DSM_CHNE	ОШИБКА ИЗМЕНЕНИЯ РЕЖИМА;
DSM_NOTIMP	ФУНКЦИЯ НЕ РЕАЛИЗОВАНА;
DSM_IPLZHI	УРОВЕНЬ ПРИОРИТЕТА ПЕРЕРЫВАНИЯ (IPL) СЛИШКОМ ВЫСОК;
DSM_ICERR	ОШИБКА ДАТЧИКА ИНТЕРВАЛОВ ВРЕМЕНИ;
DSM_ICBUSY	ДАТЧИК ИНТЕРВАЛОВ ВРЕМЕНИ ЗАНЯТ;
DSM_ARITH	АРИФМЕТИЧЕСКИЙ ЗАХВАТ.

1Э.00078-01 33 01

7.10.4. #DS\_DSSDEF - ОПРЕДЕЛЕНИЕ ТОЧЕК ВХОДА  
СЛУЖЕБНЫХ СРЕДСТВ СУПЕРВИЗОРА

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_DSSDEF [GBL]

ГДЕ GBL - GLOBAL (ГЛОБАЛЬНЫЕ ИМЕНА) ИЛИ LOCAL  
(ЛОКАЛЬНЫЕ ИМЕНА). ПО УМОЛЧАНИЮ ГЕНЕРИРУЮТСЯ ЛОКАЛЬНЫЕ  
ИМЕНА.

МАКРОКОМАНДА ГЕНЕРИРУЕТ СИМВОЛИЧЕСКИЕ ИМЕНА, КОТОРЫЕ  
ОПРЕДЕЛЯЮТ ТОЧКИ ВХОДА СЛУЖЕБНЫХ СРЕДСТВ СУПЕРВИЗОРА ДЛЯ  
ДИАГНОСТИЧЕСКИХ ПРОГРАММ. ПРИ РАСШИРЕНИИ СЛУЖЕБНЫХ МАКРОКО-  
МАНД СУПЕРВИЗОРА ФОРМИРУЕТСЯ ВЫЗОВ ДЛЯ ОДНОЙ ИЗ ЭТИХ ТОЧЕК.

7.10.5. #DS\_ENVDEF - ОПРЕДЕЛЕНИЕ КОДОВ СРЕДЫ

ФОРМАТ МАКРОКОМАНДЫ

#DS\_ENVDEF

АРГУМЕНТОВ В МАКРОКОМАНДЕ ЧЕТ.

МАКРОКОМАНДА ОПРЕДЕЛЯЕТ СИМВОЛИЧЕСКИЕ ИМЕНА, КОТОРЫЕ  
ИСПОЛЬЗУЮТСЯ ДЛЯ АРГУМЕНТА ENV МАКРОКОМАНДЫ #DS\_BGNMOD.  
ОПРЕДЕЛЯЮТСЯ СЛЕДУЮЩИЕ СИМВОЛИЧЕСКИЕ ИМЕНА:

SER\_FUNCTIONAL;

SER\_REPAIR;

SER\_FUNCTIONAL;

SER\_REPAIR.

13.00078-01 33 01

7.10.6. #DS\_ERRDEF - ОПРЕДЕЛЕНИЕ СМЕЩЕНИЙ  
СПИСКА АРГУМЕНТОВ ВЫЗОВА ОШИБКИ

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_ERRDEF

АРГУМЕНТОВ В МАКРОКОМАНДЕ НЕТ.

МАКРОКОМАНДА ОПРЕДЕЛЯЕТ СИМВОЛИЧЕСКИЕ АРГУМЕНТЫ ДЛЯ  
МАКРОКОМАНДЫ #DS\_ERRDEF\_X> #DS\_ERRNADR\_X, #DS\_ERRSOFT\_X И  
#DS\_ERRSYS\_X.

ОПРЕДЕЛЯЮТСЯ СЛЕДУЮЩИЕ СИМВОЛИЧЕСКИЕ ИМЕНА:

ERR#_NUM	НОМЕР ОШИБКИ;
ERR#_UNIT	ЛОГИЧЕСКИЙ НОМЕР УСТРОЙСТВА;
ERR#_MSGADR	АДРЕС ЗАГОЛОВКА СООБЩЕНИЯ;
ERR#_POINTER	ПОДПРОГРАММА ПЕЧАТИ РАСШИРЕННОГО СООБЩЕНИЯ;
ERR#_P1	ДОПОЛНИТЕЛЬНЫЕ ДАННЫЕ (P1);
ERR#_P2	ДОПОЛНИТЕЛЬНЫЕ ДАННЫЕ (P2);
ERR#_P3	ДОПОЛНИТЕЛЬНЫЕ ДАННЫЕ (P3);
ERR#_P4	ДОПОЛНИТЕЛЬНЫЕ ДАННЫЕ (P4);
ERR#_P5	ДОПОЛНИТЕЛЬНЫЕ ДАННЫЕ (P5);
ERR#_P6	ДОПОЛНИТЕЛЬНЫЕ ДАННЫЕ (P6).

7.10.7. #DS\_HRDEF - ОПРЕДЕЛЕНИЕ СИМВОЛИЧЕСКИХ  
ИМЕН СМЕЩЕНИЙ ЭЛЕМЕНТОВ ТАБЛИЦЫ P

ФОРМАТ МАКРОКОМАНДЫ:

#DS\_HRDEF

АРГУМЕНТОВ У МАКРОКОМАНДЫ НЕТ.

МАКРОКОМАНДА ОПРЕДЕЛЯЕТ СИМВОЛИЧЕСКИЕ ИМЕНА СМЕЩЕНИЙ  
АППАРАТНОЙ ТАБЛИЦЫ P. ЭТО ПОЗВОЛЯЕТ ПРОГРАММИСТУ ОБРАЩАТЬСЯ

К ЭЛЕМЕНТАМ ТАБЛИЦЫ P С ПОМОЩЬЮ СИМВОЛИЧЕСКИХ ИМЕН.

МАКРОКОМАНДА #DS\_HPRODEF ОПРЕДЕЛЯЕТ СИМВОЛИЧЕСКИЕ ИМЕНА СМЕЩЕНИИ, НЕ ЗАВИСЯЩИЕ ОТ УСТРОЙСТВА.

ОПРЕДЕЛЯЮТСЯ СЛЕДУЮЩИЕ СИМВОЛИЧЕСКИЕ ИМЕНА:

HPAQ_DEVICE	ОПИСАТЕЛЬ ИМЕНИ УСТРОЙСТВА (ЧЕТВЕРНОЕ СЛОВО);
HPAW_SIZE	ОБЩИЙ РАЗМЕР ТАБЛИЦЫ P УСТРОЙСТВА;
HPAB_FLAGS	БАЙТ ФЛАГОВ;
HPAB_DRIVE	НОМЕР УСТРОЙСТВА;
HPAT_DEVICE	ИМЯ УСТРОЙСТВА В КОДЕ КОИ-8 С НАЧАЛЬНЫМ СИМВОЛОМ _ И МАКСИМАЛЬНОЙ ДЛИНОЙ - 11 СИМВОЛОВ;
HPAA_DEVICE	АДРЕС ПРОВЕРЯЕМОГО УСТРОЙСТВА;
HPAA_DVA	АДРЕС, ИСПОЛЬЗУЕМЫЙ ДЛЯ ПРЯМОЙ АДРЕСАЦИИ ПРОВЕРЯЕМОГО УСТРОИ- СТВА ЧЕРЕЗ ДАННОЕ УСТРОЙСТВО;
HPAA_LINK	АДРЕС ТАБЛИЦЫ P ДЛЯ УСТРОИС- ТВА, СВЯЗЫВАЮЩЕГО ДАННОЕ УСТРОЙСТВО С ЦЕНТРАЛЬНЫМ ПРОЦЕССОРОМ;
HPAT_VECTOR	ПЕРВИЧНЫЙ ВЕКТОР ПРЕРЫВАНИЯ ДЛЯ УСТРОЙСТВА;
HPAT_TYPE	ТИП ОБОРУДОВАНИЯ В КОДЕ КОИ-8 С НАЧАЛЬНЫМ БАЙТОМ ДЛИНЫ, МАКСИМАЛЬНАЯ ДЛИНА - 11 СИМВОЛОВ;
HPAA_DEPENDENT	НАЧАЛО НЕСТАНДАРТНОЙ ЧАСТИ ТАБЛИЦЫ.

7.10.8. MDS\_PARDEF - ОПРЕДЕЛЕНИЕ ПАРАМЕТРОВ

ФОРМАТ МАКРОКОМАНДЫ:

MDS\_PARDEF [GBL]

ГДЕ GBL - GLOBAL (ГЛОБАЛЬНЫЕ ИМЕНА) ИЛИ LOCAL

(ЛОКАЛЬНЫЕ ИМЕНА). ПО УМОЛЧАНИЮ ГЕНЕРИРУЮТСЯ ЛОКАЛЬНЫЕ ИМЕНА.

МАКРОКОМАНДА MDS\_PARDEF ФОРМИРУЕТ СИМВОЛИЧЕСКИЕ ИМЕНА ОСНОВАНИИ (СИСТЕМ СЧИСЛЕНИЯ) И АРГУМЕНТОВ МАСКИ ИСКЛЮЧЕНИЯ И ИХ ЗНАЧЕНИЯ. УКАЗАННЫЕ СИМВОЛИЧЕСКИЕ ИМЕНА ИСПОЛЬЗУЮТСЯ МАКРОКОМАНДАМИ MDS\_ASK...\_X.

ГЕНЕРИРУЮТСЯ СЛЕДУЮЩИЕ СИМВОЛИЧЕСКИЕ ИМЕНА:

PAR#\_BIN - ДВОИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ;

PAR#\_OCT - ВОСЬМЕРИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ;

PAR#\_DEC - ДЕСЯТИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ;

PAR#\_HEX - ШЕСТНАДЦАТЕРИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ;

PAR#\_NO - УСТАНОВКА ПО УМОЛЧАНИЮ "НЕТ";

PAR#\_YES - УСТАНОВКА ПО УМОЛЧАНИЮ "ДА";

PAR#M\_NODEF;

PAR#M\_ATLO;

PAR#M\_ATHI;

PAR#M\_ATDEF;

PAR#V\_NODEF;

PAR#V\_ATLO;

PAR#V\_ATHI;

PAR#V\_ATDEF.

## 7.11. МАКРОКОМАНДЫ УПРАВЛЕНИЯ ПАМЯТЬЮ

УПРАВЛЕНИЕ ПАМЯТЬЮ ДЛЯ ДИАГНОСТИЧЕСКОЙ СИСТЕМЫ ПРОИЗВОДИТСЯ ЛИБО СУПЕРВИЗОРОМ, ЛИБО ОПЕРАЦИОННОЙ СИСТЕМОЙ МОС ВП. СРЕДСТВА УПРАВЛЕНИЯ ПАМЯТЬЮ ОБЕСПЕЧИВАЮТ СОВМЕСТИМОСТЬ ДЛЯ РАЗЛИЧНЫХ УСЛОВИЙ РАБОТЫ И ТИПОВ ПРОЦЕССОРОВ. ПРИ АВТОНОМНОЙ РАБОТЕ СРЕДСТВА УПРАВЛЕНИЯ ПАМЯТЬЮ ОБЫЧНО ВЫКЛЮЧЕНЫ; А ПРИ РАБОТЕ С СИСТЕМОЙ МОС ВП УПРАВЛЕНИЕ ПАМЯТЬЮ ПРИНАДЛЕЖИТ ЭТОЙ ОПЕРАЦИОННОЙ СИСТЕМЕ И ВСЕГДА ВКЛЮЧЕНО.

СУПЕРВИЗОР ОБЕСПЕЧИВАЕТ СЛУЖЕБНЫЕ СРЕДСТВА, КОТОРЫЕ ВЫПОЛНЯЮТ РАСПРЕДЕЛЕНИЕ КАК ВИРТУАЛЬНОЙ, ТАК И ФИЗИЧЕСКОЙ ПАМЯТИ. ФИЗИЧЕСКОЕ РАСПРЕДЕЛЕНИЕ ПАМЯТИ ОГРАНИЧИВАЕТСЯ ДИАГНОСТИЧЕСКИМИ ПРОГРАММАМИ УРОВНЯ 3.

ДЛЯ УПРАВЛЕНИЯ ПАМЯТЬЮ В ДИАГНОСТИЧЕСКОЙ БИБЛИОТЕКЕ ИМЕЮТСЯ МАКРОКОМАНДЫ:

ADS\_MMON\_X - ДЛЯ ВКЛЮЧЕНИЯ УПРАВЛЕНИЯ ПАМЯТЬЮ;

ADS\_MMDF\_X - ДЛЯ ВЫКЛЮЧЕНИЯ УПРАВЛЕНИЯ ПАМЯТЬЮ;

ADS\_GETBUF\_X - ДЛЯ ПОЛУЧЕНИЯ БУФЕРНОГО ПРОСТРАНСТВА;

ADS\_RELBUF\_X - ДЛЯ ОСВОБОЖДЕНИЯ БУФЕРНОГО ПРОСТРАНСТВА.

### 7.11.1. ADS\_MMON\_X - ВКЛЮЧЕНИЕ УПРАВЛЕНИЯ ПАМЯТЬЮ

ФОРМАТ МАКРОКОМАНДЫ:

ADS\_MMON\_X

АРГУМЕНТОВ У МАКРОКОМАНДЫ НЕТ.

МАКРОКОМАНДА ВЫЗЫВАЕТ ПОДПРОГРАММУ СУПЕРВИЗОРА, КОТОРАЯ ВКЛЮЧАЕТ УПРАВЛЕНИЕ ПАМЯТЬЮ. ОНА ДЕЙСТВИТЕЛЬНА ТОЛЬКО ДЛЯ ДИАГНОСТИЧЕСКИХ ПРОГРАММ УРОВНЯ 3.

13.00078-01 33 01

7.11.2. `ADS_MMOFF_X` - ВЫКЛЮЧЕНИЕ УПРАВЛЕНИЯ  
ПАМЯТЬЮ

ФОРМАТ МАКРОКОМАНДЫ:

`ADS_MMOFF_X`

АРГУМЕНТОВ У МАКРОКОМАНДЫ НЕТ.

МАКРОКОМАНДА ВЫЗЫВАЕТ ПОДПРОГРАММУ СУПЕРВИЗОРА, КОТО-  
РАЯ ВЫКЛЮЧАЕТ УПРАВЛЕНИЕ ПАМЯТЬЮ. ОНА ДЕЙСТВИТЕЛЬНА ТОЛЬКО  
ДЛЯ ДИАГНОСТИЧЕСКИХ ПРОГРАММ УРОВНЯ 3.

7.11.3. `ADS_GETBUF_X` - ПОЛУЧЕНИЕ ОБЛАСТИ  
ВИРТУАЛЬНОЙ ПАМЯТИ

ФОРМАТ МАКРОКОМАНДЫ:

`ADS_GETBUF PAGECNT, RETADR, RHYADR, REGION`

ГДЕ PAGECNT - ТРЕБУЕМОЕ ЧИСЛО СТРАНИЦ ПАМЯТИ;

RETA DR - АДРЕС МАССИВА ИЗ ДВУХ ДЛИННЫХ СЛОВ ДЛЯ ПРИЕМА  
ВИРТУАЛЬНЫХ ГРАНИЦ БУФЕРА;

RHYADR - АДРЕС МАССИВА ИЗ ДВУХ ДЛИННЫХ СЛОВ ДЛЯ ПРИЕМА  
НАЧАЛЬНЫХ И КОНЕЧНЫХ ФИЗИЧЕСКИХ АДРЕСОВ;

REGION - ОБЛАСТЬ ВИРТУАЛЬНОЙ ПАМЯТИ, В КОТОРОЙ ДОЛЖЕН  
БЫТЬ ВЫДЕЛЕН БУФЕР: 0 - ПРОСТРАНСТВО P0 (УСТАНОВКА ПО  
УМОЛЧАНИЮ), 1 - ПРОСТРАНСТВО P1, 2 - СИСТЕМНОЕ ПРОСТ-  
РАНСТВО.

ВОЗВРАЩАЕМЫЕ КОДЫ СОСТОЯНИЯ:

`SSA_NORMAL` - УСПЕШНОЕ ЗАВЕРШЕНИЕ;

`SSA_ILLPAGECNT` - ЧИСЛО СТРАНИЦ МЕНЬШЕ ЕДИНИЦЫ;

`SSA_VASFULL` - ВИРТУАЛЬНОЕ АДРЕСНОЕ ПРОСТРАНСТВО

13.00078-01 33 01

ЗАПОЛНЕНО.

МАКРОКОМАНДА ВЫЗЫВАЕТ ПОДПРОГРАММУ СУПЕРВИЗОРА ДЛЯ ПОЛУЧЕНИЯ БУФЕРНЫХ ОБЛАСТЕЙ ПАМЯТИ, КОТОРЫЕ РАЗМЕЩАЮТСЯ ПОСЛЕ ОБЛАСТИ ПАМЯТИ, ВЫДЕЛЕННОЙ ДЛЯ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ. В АВТОНОМНОМ РЕЖИМЕ РАСПРЕДЕЛЕНИЕ ФИЗИЧЕСКОЙ ПАМЯТИ ЯВЛЯЕТСЯ НЕПРЕРЫВНЫМ ПОСЛЕДОВАТЕЛЬНЫМ.

ДЛЯ ПОЛУЧЕНИЯ ПАМЯТИ, НЕОБХОДИМОЙ ДЛЯ КОНКРЕТНЫХ ТЕСТОВ, СЛЕДУЕТ ИСПОЛЬЗОВАТЬ МАКРОКОМАНДУ #DS\_GETBUF\_X СОВМЕСТНО С МАКРОКОМАНДОЙ #DS\_RELBUF\_X.

ЕСЛИ ВКЛЮЧЕНО УПРАВЛЕНИЕ ПАМЯТЬЮ, ТО БУФЕРНЫЕ ОБЛАСТИ МОГУТ БЫТЬ ВЫДЕЛЕНЫ ТОЛЬКО В ПРОСТРАНСТВЕ PU.

#### 7.11.4. #DS\_RELBUF\_X - ОСВОБОЖДЕНИЕ БУФЕРА

ФОРМАТ МАКРОКОМАНДЫ:

```
#DS_RELBUF_X PAGESNT L,RETADR] L,REGION]
```

ГДЕ PAGESNT - ЧИСЛО ОСВОБОЖДАЕМЫХ СТРАНИЦ;

RETADR - АДРЕС МАССИВА ИЗ ДВУХ ДЛИННЫХ СЛОВ ДЛЯ ПРИЕМА ГРАНИЦ ОСВОБОЖДАЕМОГО БУФЕРНОГО ПРОСТРАНСТВА;

REGION - ОБЛАСТЬ (ВИРТУАЛЬНОЕ АДРЕСНОЕ ПРОСТРАНСТВО), ИЗ КОТОРОЙ ДОЛЖЕН БЫТЬ ОСВОБОЖДЕН БУФЕР. 0 - ПРОСТРАНСТВО PU (УСТАНОВКА ПО УМОЛЧАНИЮ), 1 - ПРОСТРАНСТВУ P1, 2 - СИСТЕМНОЕ ПРОСТРАНСТВО.

ВОЗВРАЩАЕМЫЕ КОДЫ СОСТОЯНИЯ:

SSR\_NORMAL - УСПЕШНОЕ ЗАВЕРШЕНИЕ РАБОТЫ;

DSR\_FRAGBUF - БУФЕР НЕ ЯВЛЯЕТСЯ НЕПРЕРЫВНЫМ ПОСЛЕДОВАТЕЛЬНЫМ;

SSR\_ILLPAGESNT - КОЛИЧЕСТВО СТРАНИЦ МЕНЬШЕ 1;



13.00078-01 33 01

SSA\_PASOBNITO - СТРАНИЦА ПРИНАДЛЕЖИТ БОЛЕЕ

ПРИВИЛЕГИРОВАННОМУ РЕЖИМУ ДОСТУПА.

МАКРОКОМАНДА PDS\_RELBUF\_X ВЫЗЫВАЕТ ПОДПРОГРАММУ СУПЕР-  
ВИЗОРА ДЛЯ ОСВОБОЖДЕНИЯ БУФЕРНОЙ ОБЛАСТИ ПАМЯТИ.

4 71

## В. ОБРАЩЕНИЕ К ПРОГРАММЕ

ДИАГНОСТИЧЕСКАЯ ПРОГРАММА СОСТОИТ ИЗ ОТДЕЛЬНЫХ ПОДП-  
РОГРАММ, ЗАПУСК КОТОРЫХ В НУЖНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ ОСУ-  
ЩЕСТВЛЯЕТСЯ СУПЕРВИЗОРОМ.

ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ УРОВНЯ 3 ВЫПОЛНЯЮТСЯ В АВТО-  
НОМНОМ РЕЖИМЕ. ПРОГРАММЫ УРОВНЯ 2R ВЫПОЛНЯЮТСЯ ПОД УПРАВЛЕ-  
НИЕМ ОПЕРАЦИОННОЙ СИСТЕМЫ МОС ЗП, А ПРОГРАММЫ УРОВНЯ 2  
МОГУТ ВЫПОЛНЯТЬСЯ КАК В АВТОНОМНОМ РЕЖИМЕ, ТАК И ПОД УПРАВ-  
ЛЕНИЕМ ОПЕРАЦИОННОЙ СИСТЕМЫ МОС ЗП.

ПРОЦЕДУРА ЗАПУСКА СУПЕРВИЗОРА ЗАВИСИТ ОТ ТОГО, В КАКОМ  
РЕЖИМЕ ДОЛЖНА РАБОТАТЬ ДИАГНОСТИЧЕСКАЯ ПРОГРАММА: АВТОНОМ-  
НОМ ИЛИ ПОД УПРАВЛЕНИЕМ МОС ЗП.

### В.1. ЗАПУСК ДИАГНОСТИЧЕСКОГО СУПЕРВИЗОРА В АВТОНОМНОМ РЕЖИМЕ

В АВТОНОМНОМ РЕЖИМЕ ВОЗМОЖНЫ ДВА СПОСОБА ЗАПУСКА ДИАГ-  
НОСТИЧЕСКОГО СУПЕРВИЗОРА И ЗАГРУЗКИ ПРОГРАММ: С ДИСКА И С  
КОНСОЛЬНОГО УСТРОЙСТВА ЗВОДА.

ДЛЯ ЗАПУСКА СУПЕРВИЗОРА С ДИСКА СЛЕДУЕТ ВЫПОЛНИТЬ КОН-  
СОЛЬНУЮ КОМАНДУ:

>>>В ХУН

19.00078-01 33 01

ГДЕ ХУ - МНЕМОНИЧЕСКОЕ ОБОЗНАЧЕНИЕ УСТРОЙСТВА, НА КОТОРОМ  
УСТАНОВЛЕН ДИАГНОСТИЧЕСКИЙ ДИСК;

N - НОМЕР УСТРОЙСТВА.

ДЛЯ ЗАПУСКА СУПЕРВИЗОРА С КОНСОЛЬНОГО УСТРОЙСТВА ВВОДА  
НУЖНО ВЫПОЛНИТЬ КОМАНДЫ:

```
>>>L/P/S:FEUD DDU:ENSAА.EXE
```

```
>>>S 1000J
```

ГДЕ U - НОМЕР КОНСОЛЬНОГО УСТРОЙСТВА ВВОДА.

## 8.2. ЗАПУСК ДИАГНОСТИЧЕСКОГО СУПЕРВИЗОРА ПОД УПРАВЛЕНИЕМ МОС ВП

ПЕРЕД ВЫПОЛНЕНИЕМ ПРОГРАММЫ ПОД УПРАВЛЕНИЕМ ОПЕРАЦИОН-  
НОЙ СИСТЕМЫ СЛЕДУЕТ УБЕДИТЬСЯ, ЧТО ПРЕДВАРИТЕЛЬНО СМОНТИРО-  
ВАННЫЙ ДИСК С КАТАЛОГОМ SYSMAINT, СОДЕРЖАЩИИ ДИАГНОСТИЧЕС-  
КУЮ ПРОГРАММУ, НАХОДИТСЯ В РАБОЧЕМ СОСТОЯНИИ И СМОНТИРОВАН.

ЗАГРУЗКА И ЗАПУСК ДИАГНОСТИЧЕСКОГО СУПЕРВИЗОРА ПРОИС-  
ХОДИТ С ПОМОЩЬЮ КОМАНДЫ:

```
BRUN <УСТРОЙСТВО>:[КАТАЛОГ] ENSAA
```

ГДЕ <УСТРОЙСТВО> - МНЕМОНИЧЕСКОЕ ОБОЗНАЧЕНИЕ  
УСТРОЙСТВА;

КАТАЛОГ - НАЗВАНИЕ КАТАЛОГА, В КОТОРОМ ЗАПИСАН ФАЙЛ  
ENSAА.

ДЛЯ ВЫПОЛНЕНИЯ ДИАГНОСТИЧЕСКИХ ПРОГРАММ НЕОБХОДИМО  
ПРЕДВАРИТЕЛЬНО С ПОМОЩЬЮ ПРОГРАММЫ AUTHORIZЕ УСТАНОВИТЬ  
ПРИВИЛЕГИИ И КВОТЫ В ФАЙЛЕ ИНФОРМАЦИИ О ПОЛЬЗОВАТЕЛЯХ.  
ИНФОРМАЦИЯ О ТОМ, КАКИЕ ПРИВИЛЕГИИ И КВОТЫ ДОЛЖНЫ БЫТЬ

УСТАНОВЛЕННЫ, ДАНА В [1].

### 8.3. ЗАПУСК ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ

ЗАПУСК ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ ОСУЩЕСТВЛЯЕТСЯ, КАК ПРАВИЛО, ЗА ДВА ШАГА. НА ПЕРВОМ ШАГЕ ВЫПОЛНЯЕТСЯ ПОСЛЕДОВАТЕЛЬНОСТЬ КОМАНД, ОПИСЫВАЮЩИХ СОЕДИНЕНИЕ ПРОВЕРЯЕМЫХ УСТРОЙСТВ (КОМАНДЫ ATTACH) И ВЫБОР ПРОВЕРЯЕМЫХ УСТРОЙСТВ (КОМАНДЫ SELECT). НА ВТОРОМ ШАГЕ ВЫПОЛНЯЕТСЯ ПОСЛЕДОВАТЕЛЬНОСТЬ КОМАНД LOAD И START ИЛИ RUN, ОСУЩЕСТВЛЯЮЩИХ ЗАГРУЗКУ И ЗАПУСК ДИАГНОСТИЧЕСКИХ ПРОГРАММ.

ЗАПУСК ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ МОЖНО ВЫПОЛНИТЬ ДВУМЯ СПОСОБАМИ: С ПОМОЩЬЮ ПОСЛЕДОВАТЕЛЬНОСТИ КОМАНД ДИАГНОСТИЧЕСКОГО СУПЕРВИЗОРА ИЛИ С ПОМОЩЬЮ КОМАНДНЫХ ФАЙЛОВ. ИСПОЛЬЗОВАНИЕ КОМАНДНЫХ ФАЙЛОВ ПОЗВОЛЯЕТ ЗАПУСКАТЬ ДИАГНОСТИЧЕСКУЮ ПРОГРАММУ С ПОМОЩЬЮ ОДНОЙ ИЛИ НЕСКОЛЬКИХ ПРОЦЕДУР. ОБЫЧНО ИСПОЛЬЗУЕТСЯ ДВА КОСВЕННЫХ КОМАНДНЫХ ФАЙЛА. ОДИН ИЗ НИХ ОПИСЫВАЕТ СОЕДИНЕНИЕ И ВЫБОР ПРОВЕРЯЕМЫХ УСТРОЙСТВ, А ДРУГОЙ СОДЕРЖИТ ПОСЛЕДОВАТЕЛЬНОСТЬ КОМАНД ЗАПУСКА ПРОГРАММЫ. КОРРЕКЦИЯ КОМАНДНЫХ ФАЙЛОВ МОЖЕТ ПРОИЗВОДИТЬСЯ ТЕКСТОВЫМ РЕДАКТОРОМ.

ПО УМОЛЧАНИЮ КОМАНДЫ LOAD И RUN ЗАПУСКАЮТ ДИАГНОСТИЧЕСКИЕ ПРОГРАММЫ С УСТРОЙСТВА, УСТАНОВЛЕННОГО ПО КОМАНДЕ SET LOAD ИЛИ С УСТРОЙСТВА, С КОТОРОГО ЗАГРУЖАЛСЯ ДИАГНОСТИЧЕСКИЙ СУПЕРВИЗОР.

ПОДРОБНОЕ ОПИСАНИЕ СИСТЕМЫ КОМАНД ДИАГНОСТИЧЕСКОГО СУПЕРВИЗОРА ДАНО В [1].

19.00078-01 33 01

## 9. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ. СООБЩЕНИЯ

ВХОДНАЯ ИНФОРМАЦИЯ ДЛЯ РАБОТЫ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ, КОТОРАЯ ПОСТУПАЕТ ОТ СУПЕРВИЗОРА, ЗАДАЕТ:

1) КОНФИГУРАЦИЮ ПРОВЕРЯЕМОГО ОБОРУДОВАНИЯ, КОТОРАЯ ПРЕДПОЛАГАЕТ ЗАДАНИЕ ТИПА И НОМЕРА УСТРОЙСТВА, СПОСОБА ЕГО ВКЛЮЧЕНИЯ В ВЫЧИСЛИТЕЛЬНЫЙ КОМПЛЕКС, А ТАКЖЕ ВЫБОР УСТРОЙСТВ ДЛЯ ТЕКУЩЕЙ ПРОЦЕДУРЫ ДИАГНОСТИРОВАНИЯ;

2) РЕЖИМ ВЫПОЛНЕНИЯ ПРОГРАММЫ, КОТОРЫЙ МОЖЕТ БЫТЬ ОДНОКРАТНЫМ, МНОГОКРАТНЫМ, ПОШАГОВЫМ. ПРОВЕРКА МОЖЕТ БЫТЬ ПОЛНОЙ И УКОРОЧЕННОЙ, ПО ОТДЕЛЬНЫМ ТЕСТАМ И ПОДТЕСТАМ. РЕЖИМ ВЫПОЛНЕНИЯ УСТАНАВЛИВАЕТСЯ С ПОМОЩЬЮ КЛЮЧЕЙ В КОМАНДАХ START, RUN И УПРАВЛЯЮЩИХ ФЛАГОВ;

3) РЕАКЦИЮ НА ОШИБКУ, КОТОРАЯ УСТАНАВЛИВАЕТСЯ С ПОМОЩЬЮ УПРАВЛЯЮЩИХ ФЛАГОВ. ПРОГРАММА МОЖЕТ ОСТАНАВЛИВАТЬСЯ ПО ОШИБКЕ, ЗАЦИКЛИВАТЬСЯ НА УЧАСТКЕ, ЗАФИКСИРОВАВШЕМ ОШИБКУ, ВЫВОДИТЬ РАСШИРЕННЫЕ ИЛИ КРАТКИЕ СООБЩЕНИЯ ОБ ОШИБКАХ;

4) НЕОБХОДИМОСТЬ ВЫВОДА СООБЩЕНИИ О ХОДЕ ВЫПОЛНЕНИЯ ПРОГРАММЫ. УСТАНАВЛИВАЕТСЯ С ПОМОЩЬЮ ФЛАГОВ.

ВЫХОДНЫМИ ДАННЫМИ ПРОГРАММЫ ЯВЛЯЮТСЯ СООБЩЕНИЯ ОБ ОШИБКАХ ИЛИ О ВЫПОЛНЕНИИ ТЕСТОВ БЕЗ ОШИБОК.

СООБЩЕНИЯ ОБ ОШИБКАХ ПОЗВОЛЯЮТ УСТАНОВИТЬ НЕИСПРАВНУЮ ПОДСИСТЕМУ, МОДУЛЬ ИЛИ ФУНКЦИЮ. СООБЩЕНИЯ ОБ ОШИБКАХ ВКЛЮЧАЮТ НОМЕР ТЕСТА И ПОДТЕСТА, ОБНАРУЖИВШЕГО ОШИБКУ И КОД ОШИБКИ. СООТВЕТСТВИЕ МЕЖДУ КОДОМ И ОШИБКОЙ ПРИВЕДЕНО В 11.1.

КРОМЕ СООБЩЕНИИ ОБ ОШИБКАХ ДИАГНОСТИЧЕСКАЯ ПРОГРАММА МОЖЕТ ВЫРАБАТЫВАТЬ ИНФОРМАЦИЮ, НЕОБХОДИМУЮ ДЛЯ ВЫПОЛНЕНИЯ ДРУГИХ ПРОГРАММ.

ВХОДНАЯ И ВЫХОДНАЯ ИНФОРМАЦИЯ ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ ОПРЕДЕЛЯЕТСЯ КОМАНДАМИ ДИАГНОСТИЧЕСКОГО СУПЕРВИЗОРА,

19.00078-01 33 01'

ЗАДАВАЕМЫМИ ОПЕРАТОРОМ ИЛИ ПОСТУПАЮЩИМИ С КОМАНДНОГО ФАЙЛА.  
ОПИСАНИЕ КОМАНД ДИАГНОСТИЧЕСКОГО СУПЕРВИЗОРА ДАНО В [1].

АЛФАВИТНЫЙ УКАЗАТЕЛЬ МАКРОКОМАНД

	СТР.
ADDEF - ОПРЕДЕЛЕНИЕ ПОЛЕЙ В СТРУКТУРЕ.....	97
ADDEFEND - ОКОНЧАНИЕ ОПРЕДЕЛЕНИЯ.....	98
ADDEFINI - НАЧАЛО СТРУКТУРЫ ДАННЫХ.....	97
ADS_ABORT - ОТМЕНА ПРОГРАММЫ ИЛИ ТЕСТА.....	80
ADS_ASKADR_X - ЗАПРОС У ОПЕРАТОРА АДРЕСА.....	100
ADS_ASKDATA_X - ЗАПРОС У ОПЕРАТОРА ЧИСЛОВОЙ ВЕЛИЧИНЫ.....	103
ADS_ASKLGCL_X - ЗАПРОС У ОПЕРАТОРА ЛОГИЧЕСКОГО ОТВЕТА.....	107
ADS_ASKSTR_X - ЗАПРОС У ОПЕРАТОРА СТРОКИ.....	102
ADS_ASKVLD_X - ЗАПРОС У ОПЕРАТОРА ЧИСЛОВОГО ЗНАЧЕНИЯ В КОДЕ КОИ-8.....	105
ADS_BCOMPLETE - ПЕРЕХОД ПРИ УСПЕШНОМ ВЫПОЛНЕНИИ.....	82
ADS_BERROR - ПЕРЕХОД ПРИ ОШИБКЕ.....	84
ADS_BGCLEAN - НАЧАЛО ПОДПРОГРАММЫ ОЧИСТКИ.....	58
ADS_BGNDATA - НАЧАЛО МАССИВА ДАННЫХ ДЛЯ ТЕСТА.....	65
ADS_BGNINIT - НАЧАЛО ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ.....	57
ADS_BGNMESSAGE - НАЧАЛО ПОДПРОГРАММЫ ПЕЧАТИ СООБЩЕНИИ ОБ ОШИБКАХ.....	60
ADS_BGNMOD - НАЧАЛО ПРОГРАММНОГО МОДУЛЯ.....	57
ADS_BGNSTAT - НАЧАЛО СТАТИСТИЧЕСКОЙ ТАБЛИЦЫ.....	55
ADS_BGNSUB - НАЧАЛО ПОДТЕСТА.....	74
ADS_BGNSUMMARY - НАЧАЛО ПОДПРОГРАММЫ ПЕЧАТИ ИТОГОВОГО	

ОТЧЕТА.....	59
ADS_BGNTEST - НАЧАЛО ТЕСТОВОЙ ПОДПРОГРАММЫ.....	63
ADS_BITDEF - ОПРЕДЕЛЕНИЕ МНЕМОНИЧЕСКИХ ОБЗНАЧЕНИЙ БИТОВОЙ МАСКИ.....	132
ADS_BNDCOMPLETE - ПЕРЕХОД ПРИ НЕУСПЕШНОМ ВЫПОЛНЕНИИ.....	83
ADS_BNERROR - ПЕРЕХОД ПРИ ОТСУТСТВИИ ОШИБКИ.....	84
ADS_BNOPER - ПЕРЕХОД ПРИ ОТСУТСТВИИ ОПЕРАТОРА.....	73
ADS_BNPASSD - ПЕРЕХОД ПРИ НЕ ПЕРВОМ ВЫПОЛНЕНИИ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ.....	69
ADS_BNQUICK - ПЕРЕХОД ПРИ ОТСУТСТВИИ РЕЖИМА БЫСТРОЙ ПРОВЕРКИ.....	71
ADS_BDPER - ПЕРЕХОД ПРИ НАЛИЧИИ ОПЕРАТОРА.....	72
ADS_BPASSD - ПЕРЕХОД ПРИ ПЕРВОМ ВЫПОЛНЕНИИ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ.....	69
ADS_BQUICK - ПЕРЕХОД В РЕЖИМЕ БЫСТРОЙ ПРОВЕРКИ.....	71
ADS_BREAK - ВКЛИНИВАНИЕ В ДИАГНОСТИЧЕСКУЮ ПРОГРАММУ.....	81
ADS_CKLOOP - УПРАВЛЕНИЕ ЗАЦИКЛИВАНИЕМ.....	75
ADS_CLI - ДЕРЕВО ИНТЕРПРЕТАТОРА КОМАНДНОЙ СТРОКИ.....	111
ADS_DEVTYP - УКАЗАНИЕ ТИПОВ УСТРОЙСТВ.....	96
ADS_DISPATCH - ИНИЦИАЛИЗАЦИЯ ТАБЛИЦЫ ДИСПЕТЧЕРА.....	54
ADS_DSDEF - ОПРЕДЕЛЕНИЕ СИМВОЛИЧЕСКИХ ИМЕН ФЛАГОВ.....	133
ADS_DSSDEF - ОПРЕДЕЛЕНИЕ КОДОВ СОСТОЯНИИ.....	134
ADS_DSSDEF - ОПРЕДЕЛЕНИЕ ТОЧЕК ВХОДА СЛУЖЕБНЫХ СРЕДСТВ СУПЕРВИЗОРА.....	136
ADS_ENDCLEAN - КОНЕЦ ПОДПРОГРАММЫ ОЧИСТКИ.....	59
ADS_ENDDATA - КОНЕЦ МАССИВА ДАННЫХ ДЛЯ ТЕСТА.....	66
ADS_ENDINIT - КОНЕЦ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ.....	57
ADS_ENDMESSAGE - КОНЕЦ ПОДПРОГРАММЫ ПЕЧАТИ СООБЩЕНИИ ОБ ОШИБКАХ.....	61
ADS_ENDMOD - КОНЕЦ ПРОГРАММНОГО МОДУЛЯ.....	52

ADS_ENDPASS_X - СООБЩЕНИЕ СУПЕРВИЗОРУ ОБ ОКОНЧАНИИ ПРОХОДА ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ.....	70
ADS_ENDSTAT - КОНЕЦ СТАТИСТИЧЕСКОЙ ТАБЛИЦЫ.....	55
ADS_ENDSUB - КОНЕЦ ПОДТЕСТА.....	74
ADS_ENDSUMMARY - КОНЕЦ ПОДПРОГРАММЫ ПЕЧАТИ ИТОГОВОГО ОТЧЕТА.....	60
ADS_ENDTEST - КОНЕЦ ТЕСТОВОЙ ПОДПРОГРАММЫ.....	64
ADS_ENVDEF - ОПРЕДЕЛЕНИЕ КОДОВ СРЕДЫ.....	130
ADS_ERRDEF - ОПРЕДЕЛЕНИЕ СМЕЩЕНИЯ СПИСКА АРГУМЕНТОВ ВЫЗОВА ОШИБКИ.....	137
ADS_ERRDEV_X - ПЕЧАТЬ ЗАГОЛОВКА ПРИ НЕУСТРАНИМОЙ ОШИБКЕ УСТРОЙСТВА.....	120
ADS_ERRHARD_X - ПЕЧАТЬ ЗАГОЛОВКА ОБ АППАРАТНОЙ ОШИБКЕ..	121
ADS_ERRSOFT_X - ПЕЧАТЬ ЗАГОЛОВКА ОБ УСТРАНИМОЙ ОШИБКЕ ПРОГРАММНЫХ СРЕДСТВ.....	122
ADS_ERRSYS_X - ПЕЧАТЬ ЗАГОЛОВКА О НЕУСТРАНИМОЙ СИСТЕМНОЙ ОШИБКЕ.....	119
ADS_ESCAPE - УСЛОВНЫЙ ВЫХОД ИЗ ТЕСТА ИЛИ ПОДТЕСТА.....	77
ADS_EXIT - БЕЗУСЛОВНЫЙ ВЫХОД ИЗ ПРОГРАММНОЙ ФАЗЫ.....	79
ADS_GETBUF_X - ПОЛУЧЕНИЕ ОБЛАСТИ ВИРТУАЛЬНОЙ ПАМЯТИ.....	141
ADS_GPHARD_X - ПОЛУЧЕНИЕ БАЗОВОГО АДРЕСА Р-ТАБЛИЦЫ.....	100
ADS_HEADER - ФОРМИРОВАНИЕ ЗАГОЛОВКА ПРОГРАММНОЙ СЕКЦИИ..	52
ADS_HPRODEF - ОПРЕДЕЛЕНИЕ СИМВОЛИЧЕСКИХ ИМЕН Р-ТАБЛИЦЫ..	138
ADS_INLOOP_X - ПРОВЕРКА ЗАЦИКЛИВАНИЯ.....	70
ADS_MMDFE_X - ВЫКЛЮЧЕНИЕ УПРАВЛЕНИЯ ПАМЯТЬЮ.....	141
ADS_MMON_X - ВКЛЮЧЕНИЕ УПРАВЛЕНИЯ ПАМЯТЬЮ.....	141
ADS_PAGE - ИСКЛЮЧЕНИЕ РАСПЕЧАТКИ ВЫЗОВА МАКРОКОМАНДЫ ADS_SWITL.....	62
ADS_PARDEF - ОПРЕДЕЛЕНИЕ ПАРАМЕТРОВ.....	139
ADS_PARSE_X - АНАЛИЗ КОМАНДЫ.....	110



ADS_PRINTP_X - ПЕЧАТЬ ОСНОВНОЙ ИНФОРМАЦИИ ОБ ОШИБКЕ.....	125
ADS_PRINTF_X - ПРИНУДИТЕЛЬНАЯ ПЕЧАТЬ СООБЩЕНИЯ.....	127
ADS_PRINTS_X - ПЕЧАТЬ ИТОГОВОГО ОТЧЕТА.....	128
ADS_PRINTX_X - ПЕЧАТЬ РАСШИРЕННОЙ ИНФОРМАЦИИ ОБ ОШИБКЕ.	126
ADS_RELBUF_X - ОСВОБОЖДЕНИЕ БУФЕРА.....	142
ADS_SBTTL - ФОРМИРОВАНИЕ ПОДЗАГОЛОВКА ТЕСТА ИЛИ ПОДТЕСТА.....	62
ADS_SECTDEF - ОПРЕДЕЛЕНИЕ ТЕСТОВЫХ СЕКЦИЙ.....	56
ADS_SECTION - ТАБЛИЦА ОПРЕДЕЛЕНИЯ ТЕСТОВЫХ СЕКЦИЙ.....	56
ADS_DESIMAL - ЗАПРОС ДЕСЯТИЧНОЙ ВЕЛИЧИНЫ ДЛЯ P-ТАБЛИЦЫ.....	92
ADS_AEND - ОКОНЧАНИЕ ОБРАБОТКИ P-ТАБЛИЦЫ.....	96
ADS_AFETCH - ИЗВЛЕЧЕНИЕ ПОЛЯ ИЗ P-ТАБЛИЦЫ.....	95
ADS_HEXADESIMAL - ЗАПРОС ШЕСТНАДЦАТЕРИЧНОЙ ВЕЛИЧИНЫ ДЛЯ P-ТАБЛИЦЫ.....	93
ADS_INITIALIZE - НАЧАЛО ОБРАБОТКИ P-ТАБЛИЦЫ.....	91
ADS_LITERAL - ОПРЕДЕЛЕНИЕ ЗНАЧЕНИЯ КОНСТАНТЫ ДЛЯ P-ТАБЛИЦЫ.....	94
ADS_OCTAL - ЗАПРОС ВОСЬМЕРИЧНОЙ ВЕЛИЧИНЫ ДЛЯ P-ТАБЛИЦЫ.....	93
ADS_STORE - ВНЕСЕНИЕ ВЕЛИЧИНЫ В P-ТАБЛИЦУ.....	95
ADS_STRING - ЗАПРОС СТРОКИ СИМВОЛОВ ДЛЯ P-ТАБЛИЦЫ.....	94

### **ПРИМЕР ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ**

ПРИМЕР ИЛЛЮСТРИРУЕТ ОБЩУЮ СТРУКТУРУ И МЕТОДЫ ПОСТРОЕНИЯ ДИАГНОСТИЧЕСКИХ ПРОГРАММ, ИЗЛОЖЕННЫЕ В ДАННОМ РУКОВОДСТВЕ.

ПРИМЕР ПРЕДСТАВЛЯЕТ СОБОЙ РЕЗУЛЬТАТ ТРАНСЛЯЦИИ ГОЛОВНОГО МОДУЛЯ EVAA1 И ТЕСТОВОГО МОДУЛЯ EVAA2 ПРОГРАММЫ ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ.

13.00078-01 33 01

ZZ\_EVAAA1  
TABLE OF CONTENTS

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ

12-AUG-1986 10:43:12

MOS BP MACRO V04-00

PAGE 0

(1)	10	МАКРООПРЕДЕЛЕНИЕ ОПИСАТЕЛЕЙ P-ТАБЛИЦЫ ДЛЯ LP27
(1)	24	МАКРООПРЕДЕЛЕНИЕ ОПИСАТЕЛЕЙ P-ТАБЛИЦЫ ДЛЯ LN01
(1)	39	ОПИСАНИЯ
(1)	73	БЛОК ДАННЫХ ЗАГОЛОВКА ПРОГРАММЫ
(1)	88	ТАБЛИЦА ДИСПЕТЧЕРА
(1)	101	ЛОКАЛЬНЫЕ И ГЛОБАЛЬНЫЕ ПРИСВАИВАНИЯ
(1)	165	СООБЩЕНИЯ ОПЕРАТОРУ НА ТЕРМИНАЛ
(1)	216	СЕКЦИЯ ГЛОБАЛЬНЫХ ДАННЫХ ПРОГРАММЫ И СООБЩЕНИЯ OUT
(1)	446	КОД ИНИЦИАЛИЗАЦИИ
(1)	591	КОД ОЧИСТКИ
(1)	608	КОД ЗАКЛЮЧИТЕЛЬНОГО ОТЧЕТА
(1)	622	ПОДПРОГРАММА ПЕЧАТИ
(1)	641	ПОДПРОГРАММА %QIOW,S ВЫЗОВА СИСТЕМНОГО ОБСЛУЖИВАНИЯ
(1)	688	ПОДПРОГРАММА %QIOW,S, ИСПОЛЬЗУЮЩАЯ СИСТЕМНУЮ ФУНКЦИЮ IO%_WRITERBLK

13.00078-01 33 01

ZZ\_EVAAA1  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ 12-AUG-1986 10:43:12 МОС ВП MACRO V04-00 PAGE 1  
12-AUG-1986 10:42:50 DUA1:[IK]AAA1.MAR;3 (1)

```
0000 2 .TITLE ZZ_EVAAA1 ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ
0000 3 .IDENT /5.3/ ; НОМЕР ВЕРСИИ
0000 4 .LIST MEB
0000 5 .NLIST END
0000 6 .LIBRARY /SYS$LIBRARY:DIAG/ ; ДИАГНОСТИЧЕСКАЯ МАКРОБИБЛИОТЕКА
00000000 7 .PSECT HEADER, PAGE, NOWRT
0000 8 $DS_BGNMOD SEP_FUNCTIONAL ; МАКРОКОМАНДА НАЧАЛА ПРОГРАММНОГО МОДУЛЯ
0000 ; .MACROASSEMBLER LAB333
```

13.00078-01 33 01

ZZ\_EVAAA1  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ 12-AUG-1986 10:43:12 МОС ВП MACRO V04-00  
МАКРООПРЕДЕЛЕНИЕ ОПИСАТЕЛЕЙ Р-ТАБЛИЦЫ ДЛЯ LP27 12-AUG-1986 10:42:50 DUA1: [IKJAAA1.MAR;3

PAGE 2

(1)

```
0000 10 .SBTTL МАКРООПРЕДЕЛЕНИЕ ОПИСАТЕЛЕЙ Р-ТАБЛИЦЫ ДЛЯ LP27
0000 11
0000 12 .MACRO  %DS_LP27_DEF      %GBL
0000 13          %DEFINI LP27,%GBL,HP%А_DEPENDENT
0000 14          %DEF      LP27%К_LEN
0000 15          %DEFEND LP27,%GBL,DEF
0000 16 .ENDM   %DS_LP27_DEF
0000 17 .MACRO  %DS_LP27
0000 18          %DS_LP27_DEF
0000 19          %DS_%INITIALIZE LP27,LP27%К_LEN,1
0000 20          %DS_%LITERAL   HP%М_ALLOC
0000 21          %DS_%STORE     HP%В_FLAGS,0,8
0000 22          %DS_%END
0000 23 .ENDM   %DS_LP27
0000 24 .SBTTL МАКРООПРЕДЕЛЕНИЕ ОПИСАТЕЛЕЙ Р-ТАБЛИЦЫ ДЛЯ LN01
0000 25
0000 26 .MACRO  %DS_LN01_DEF      %GBL
0000 27          %DEFINI LN01,%GBL,HP%А_DEPENDENT
0000 28          %DEF      LN01%К_LEN
0000 29          %DEFEND LN01,%GBL,DEF
0000 30 .ENDM   %DS_LN01_DEF
0000 31 .MACRO  %DS_LN01
0000 32          %DS_LN01_DEF
0000 33          %DS_%INITIALIZE LN01,LN01%К_LEN,1
0000 34          %DS_%LITERAL   HP%М_ALLOC
0000 35          %DS_%STORE     HP%В_FLAGS,0,8
0000 36          %DS_%END
0000 37 .ENDM   %DS_LN01
```

13.00078-01 33 01

ZZ\_EVAAA1  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ  
ОПИСАНИЯ

12-AUG-1986 10:43:12 MDC ВП MACRO V04-00  
12-AUG-1986 10:42:50 DUA1:[IKJAAA1.MAR;3

PAGE 3

(1)

```
0000 39 .SBTTL ОПИСАНИЯ
0000 40
0000 41 ;+++++
0000 42 ;
0000 43 ;ПРИРАВНИВАНИЕ СИМВОЛОВ
0000 44 ;
0000 45 ;+++++
0000 46
0000 47 ;%DCDEF
0000 51
0000 52 ;%LPDEF
0000 56 ;%DEVDEF
0000 60
0000 61 ;%DS_PARDEF      МАКРОКОМАНДА ОПРЕДЕЛЕНИЯ ПАРАМЕТРОВ
0000 62      %DS_PARDEF
0000 63      %DS_HRDEF ;МАКРОКОМАНДА ОПРЕДЕЛЕНИЯ СМЕЩЕНИЯ АППАРАТНОЙ ТАБЛИЦЫ P
0000 64
0000 65 ;%DS_DSADEF МАКРОКОМАНДА ОПРЕДЕЛЕНИЯ ОБЛАСТЕЙ СУПЕРВИЗОРА, КОМАНД АРТ И ПОЧТ. ЯЩ.
0000 66      %DS_DSADEF
0000 67
0000 68 ;%DS_BITDEF МАКРОКОМАНДА ОПРЕДЕЛЕНИЯ МНЕМОНИКИ БИТОВЫХ ЗНАЧЕНИИ
0000 69      %DS_BITDEF
0000 70
0000 71 %DS_DSSDEF      GLOBAL ;МАКРОКОМАНДА ОПРЕДЕЛЕНИЯ ТОЧЕК ВХОДА ОБСЛУЖИВАНИЯ СУПЕРВИЗОРА
```

ZZ\_EVAAA1  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ  
БЛОК ДАННЫХ ЗАГОЛОВКА ПРОГРАММЫ

12-AUG-1986 10:43:12 МОС ВП MACRO V04-00  
12-AUG-1986 10:42:50 DUA1: [IKJAAA1.MAR;3

PAGE 4

(1)

```

0000 73 .SBTTL БЛОК ДАННЫХ ЗАГОЛОВКА ПРОГРАММЫ
0000 74
0000 75 ;+++++
0000 76 ;
0000 77 ; ФУНКЦИОНАЛЬНОЕ ОПИСАНИЕ:
0000 78 ;
0000 79 ; ДИАГНОСТИЧЕСКИЙ СУПЕРВИЗОР ПРОСМАТРИВАЕТ ИНФОРМАЦИЮ ЗАГОЛОВКА,
0000 80 ; НАЧИНАЯ С ВИРТУАЛЬНОГО АДРЕСА 200 <H>. БЛОК ДАННЫХ ЗАГОЛОВКА
0000 81 ; ПРОГРАММЫ СОДЕРЖИТ ПАРАМЕТРЫ, КОТОРЫЕ ПОЗВОЛЯЮТ ДИАГНОСТИЧЕСКОМУ
0000 82 ; СУПЕРВИЗОРУ УПРАВЛЯТЬ СООТВЕТСТВУЮЩЕЙ ПРОГРАММОЙ.
0000 83 ;
0000 84 ;+++++
0000 85
0000 86 xDS_HEADER <EVAAA LINE PRINTER DIAGNOSTIC>,5,3,10
0000 .SAVE
00000000 .PSECT xHEADER,PAGE,NOEXE,NOWRT
00000058/ 0000 LxL_HEADLENGTH: .LONG A_HEADEND- ; ДЛИНА БЛОКА ДАННЫХ ЗАГОЛОВКА
00000006 0004 LxL_ENVIRON: .LONG <ENVx_SUPER@ENVxV_SUPER>+xENV ; ПРОГРАММНАЯ СРЕДА
00000058/ 0008 LxA_NAME: .ADDRESS T_NAME ; АДРЕС ТЕКСТА ИМЕНИ ПРОГРАММЫ

00000005 000C LxL_REV: .LONG 5 ; УРОВЕНЬ ПРОГРАММНОЙ ПРОВЕРКИ
00000003 0010 LxL_UPDATE: .LONG 3 ; ПОРЯДОК ДИАГНОСТИРОВАНИЯ
00000000/ 0014 LxA_LASTAD: .ADDRESS LASTAD ; ПЕРВАЯ СВОБОДНАЯ ЯЧЕЙКА ПОСЛЕ ПРОГРАММЫ
00000000/ 0018 LxL_DTP: .ADDRESS DISPATCH ; УКАЗАТЕЛЬ ТАБЛИЦЫ ДИСПЕТЧЕРА ТЕСТА
00000A2F/ 001C LxA_DEVP: .ADDRESS AL_DEVTYP ; УКАЗАТЕЛЬ СПИСКА ТИПОВ УСТРОЙСТВ
0000000A 0020 LxL_UNIT: .LONG 10 ; ЧИСЛО УС-ТВ, КОТОРЫЕ МОГУТ БЫТЬ ПРОВЕРЕНЫ
00000645/ 0024 LxA_DREG: .ADDRESS DEV_REG ; УКАЗАТЕЛЬ ТАБЛ. СОДЕРЖИМОГО РЕГИСТРОВ УСТ.
0000001C 0028 .LONG 28 ; ДЛИНА P-ТАБЛИЦЫ ДЛЯ СОВМЕСТИМОСТИ С 4.05
00000000/00000000/00000000/00000000/ 002C .LONG 0[4] ; НЕИСПОЛЬЗОВАННЫЕ СМЕЩЕНИЯ
00000000/ 003C LxA_ICP: .ADDRESS INITIALIZE ; УКАЗАТЕЛЬ КОДА ИНИЦИАЛИЗАЦИИ
00000000/ 0040 LxA_CCP: .ADDRESS CLEAN_UP ; УКАЗАТЕЛЬ КОДА ОЧИСТКИ
00000000/ 0044 LxA_REPP: .ADDRESS SUMMARY ; УКАЗАТЕЛЬ КОДА ЗАКЛЮЧИТЕЛЬНОГО ОТЧЕТА
00000000/ 0048 LxA_STATAB: .ADDRESS 0 ; УКАЗАТЕЛЬ СТАТИСТИЧЕСКОЙ ТАБЛИЦЫ
00000000 004C LxL_ERRTYP: .LONG 0 ; ЧИСЛО ТИПОВ xERRSOFT И xERRHARD
00000028/ 0050 LxA_SECNAM: .ADDRESS SECTION ; СПИСОК АДРЕСОВ ИМЕН СЕКЦИЙ
00000000/ 0054 LxA_TSTCNT: .ADDRESS L_TSTCNT ; УКАЗАТЕЛЬ ЧИСЛА ТЕСТОВ
0058 A_HEADEND:
20 45 4E 49 4C 20 41 41 41 56 45 00/ 0058 T_NAME: .ASCII \EVAAA LINE PRINTER DIAGNOSTIC\
47 41 49 44 20 52 45 54 4E 49 52 50 0064
43 49 54 53 4F 4E 0070
1D 0058
00000000 .PSECT _LAST,PAGE
0000 LASTAD:
00000000 .PSECT xTSTCNT,NOEXE,NOWRT,OVR, LONG
0000 L_TSTCNT:
00000000 .RESTORE

```

13.00078-01 33 01

ZZ\_EVAAA1  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ  
ТАБЛИЦА ДИСПЕТЧЕРА

12-AUG-1986 10:43:12 MOC ВП MACRO V04-00  
12-AUG-1986 10:42:50 DUA1:[IK]AAA1.MAR;3

PAGE 5

(1)

```

0000 88 .SBTTL ТАБЛИЦА ДИСПЕТЧЕРА
0000 89
0000 90 ;+++++
0000 91 ;
0000 92 ; ФУНКЦИОНАЛЬНОЕ ОПИСАНИЕ:
0000 93 ; ТАБЛИЦА ДИСПЕТЧЕРА ПРЕДСТАВЛЯЕТ СОБОЙ СОВОКУПНОСТЬ
0000 94 ; АДРЕСОВ, ГЕНЕРИРУЕМЫХ МАКРОКОМАНДОЙ xDS_VGNTEST И
0000 95 ; ИСПОЛЬЗУЕМЫХ СУПЕРВИЗОРОМ
0000 96 ;+++++
0000 97 ;
0000 98 xDS_DISPATCH
0000 .SAVE
00000000 .PSECT DISPATCH, LONG, NOWRT, NOEXE
0000 DISPATCH:
00000000 .PSECT DISPATCH_X, LONG, NOWRT, NOEXE
00000000/00000000/00000000/00000000/ 0000 .LONG 0[6]
00000000/00000000/ 0010
00000000 .RESTORE
00000000 99 .PSECT HEADER

```



ZZ\_EVAAA1  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ  
ЛОКАЛЬНЫЕ И ГЛОБАЛЬНЫЕ ПРИСВАИВАНИЯ

12-AUG-1986 10:43:12 МОС ВП MACRO V04-00  
12-AUG-1986 10:42:50 DUA1:[IK]AAA1.MAR;3

```

0000 101 .SBTTL ЛОКАЛЬНЫЕ И ГЛОБАЛЬНЫЕ ПРИСВАИВАНИЯ
0000 102
00000000 103 .PSECT DATA, PAGE
0000 104 ;+++++
0000 105 ;
0000 106 ; ФУНКЦИОНАЛЬНОЕ ОПИСАНИЕ:
0000 107 ;
0000 108 ; КАЖДЫЙ ТИП УСТРОЙСТВА ПЕЧАТИ ОПРЕДЕЛЯЕТСЯ
0000 109 ; БИТОВОЙ МАСКОЙ. ЭТА МАСКА РАСПОЛОЖЕНА В ЯЧЕЙКЕ LP_TYPE
0000 110 ; И ПРОВЕРЯЕТСЯ В НАЧАЛЕ КАЖДОГО ТЕСТА.
0000 111 ; LA11, LP11 - КОНТРОЛЛЕРЫ ВНЕШНИХ УСТРОЙСТВ,
0000 112 ; КОТОРЫЕ ОПЕРАЦИОННАЯ СИСТЕМА МОЖЕТ НАЗНАЧИТЬ
0000 113 ; ДЛЯ ДИАГНОСТИРОВАНИЯ.
0000 114 ; LN01, LA180, LP05/06/07/14/26/27 - ПЕЧАТАЮЩИЕ
0000 115 ; УСТРОЙСТВА, КОТОРЫЕ НАЗНАЧАЮТСЯ ДЛЯ
0000 116 ; ТЕСТИРОВАНИЯ.
0000 117 ;
0000 118 ;
0000 119 ;+++++
00000001 0000 120 LP_11==BIT0
00000002 0000 121 LA_180==BIT1
00000003 0000 122 GROUP==LP_11!LA_180
00000004 0000 123 LP_04==BIT2
00000008 0000 124 LP_05==BIT3
00000010 0000 125 LP_06==BIT4
00000020 0000 126 LP_07==BIT5
00000080 0000 127 LP_14==BIT7
00000100 0000 128 LP_25==BIT8
00000200 0000 129 LP_26==BIT9
00000400 0000 130 LP_27==BIT10
00000800 0000 131 LN_01==BIT11
0000 132
0000 133 ;+++++
0000 134 ;
0000 135 ; *DS_SECTION МАКРОКОМАНДА, ОПРЕДЕЛЯЮЩАЯ ТЕСТОВЫЕ СЕКЦИИ.
0000 136 ;
0000 137 ;+++++
0000 138
0000 139 *DS_SECTION QUALIFICATION,FUNCTIONAL,MANUAL
54 4C 55 41 46 45 44 00/ 0000 S_DEFAULT: .ASCIC \DEFAULT\
07 0000
49 54 41 43 49 46 49 4C 41 55 51 00/ 0008 S_QUALIFICATION: .ASCIC \QUALIFICATION\
4E 4F 0014
0D 0008
4C 41 4E 4F 49 54 43 4E 55 46 00/ 0016 S_FUNCTIONAL: .ASCIC \FUNCTIONAL\
0A 0016
4C 41 55 4E 41 4D 00/ 0021 S_MANUAL: .ASCIC \MANUAL\
06 0021
00000004 0028 SECTION: .LONG 4
00000000/ 002C .LONG S_DEFAULT
00000008/ 0030 .LONG S_QUALIFICATION
00000016/ 0034 .LONG S_FUNCTIONAL
00000021/ 0038 .LONG S_MANUAL

```

ZZ\_EVAAA1  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ  
ЛОКАЛЬНЫЕ И ГЛОБАЛЬНЫЕ ПРИСВАИВАНИЯ

12-AUG-1986 10:43:12 МОС ВП MACRO V04-00

PAGE 7

12-AUG-1986 10:42:50 DUA1:[IK]AAA1.MAR;3

(1)

```

003C 141 ;+++++
003C 142 ;
003C 143 ; ǝDS_LP27 ВЫЗОВ МАКРОКОМАНДЫ ФОРМИРОВАНИЯ ОПИСАТЕЛЕИ Р-ТАБЛИЦЫ ДЛЯ LP27
003C 144 ;
003C 145 ;+++++
003C 146 T_LP27: ǝDS_LP27
003C .SAVE LOCAL_BLOCK
003C .PSECT ǝABSǝ, ABS
00000032 0610 .=HPǝA_DEPENDENT
0032 .IIF NB, LP27ǝK_LEN, LP27ǝK_LEN:
0000003C .RESTORE
37 32 50 4C 00/ 003C .ASCIC 'LP27'
04 003C
32 0041 .BYTE LP27ǝK_LEN
01 0042 .BYTE 1
0000 0043 .WORD ^A''
80 0045 .BYTE PDǝ_START
86 0046 .BYTE PDǝ_LITERAL
00000001 0047 .LONG HPǝM_ALLOC
88 0048 .BYTE PDǝ_STORE
000A 004C .WORD HPǝB_FLAGS
00 004E .BYTE 0
08 004F .BYTE 8
81 0050 .BYTE PDǝ_END
0051 147 ;+++++
0051 148 ; ǝDS_LN01 ВЫЗОВ МАКРОКОМАНДЫ ФОРМИРОВАНИЯ ОПИСАТЕЛЕИ Р-ТАБЛИЦЫ ДЛЯ LN01
0051 150 ;
0051 151 ;+++++
0051 152 T_LN01: ǝDS_LN01
0051 .SAVE LOCAL_BLOCK
0051 .PSECT ǝABSǝ, ABS
00000032 0610 .=HPǝA_DEPENDENT
0032 .IIF NB, LN01ǝK_LEN, LN01ǝK_LEN:
00000051 .RESTORE
31 30 4E 4C 00/ 0051 .ASCIC 'LN01'
04 0051
32 0056 .BYTE LN01ǝK_LEN
01 0057 .BYTE 1
0000 0058 .WORD ^A''
80 005A .BYTE PDǝ_START
86 005B .BYTE PDǝ_LITERAL
00000001 005C .LONG HPǝM_ALLOC
88 0060 .BYTE PDǝ_STORE
000A 0061 .WORD HPǝB_FLAGS
00 0063 .BYTE 0
08 0064 .BYTE 8
81 0065 .BYTE PDǝ_END

```

13.00078-01 33 01

ZZ\_EVAAA1  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ  
ЛОКАЛЬНЫЕ И ГЛОБАЛЬНЫЕ ПРИСВАИВАНИЯ

12-AUG-1986 10:43:12 MOC ВП MACRO V04-00  
12-AUG-1986 10:42:50 DUA1:[IK]AAA1.MAR;3

PAGE 8

(1)

0066 154 ; ++++++

0066 155 ;

0066 156 ; СТАТИСТИЧЕСКАЯ ТАБЛИЦА

0066 157 ; (ДЛЯ ДАННОЙ ПРОГРАММЫ НЕ ИСПОЛЬЗУЕТСЯ)

0066 158 ;

0066 159 ; ++++++

13.00078-01 33 01

ZZ\_EVAAA1  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ  
ЛОКАЛЬНЫЕ И ГЛОБАЛЬНЫЕ ПРИСВАИВАНИЯ

12-AUG-1986 10:43:12 MOC ВП MACRO V04-00

PAGE 9

12-AUG-1986 10:42:50 DUA1:[IK]AAA1.MAR;3

(1)

0066 161 XDS\_BGNSTAT  
0066 STATISTIC:  
0066 162 XDS\_ENDSTAT

ZZ\_EVAAA1  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ  
ЛОКАЛЬНЫЕ И ГЛОБАЛЬНЫЕ ПРИСВАИВАНИЯ

12-AUG-1986 10:43:12 МОС ВП MACRO V04-00  
12-AUG-1986 10:42:50 DUA1: [IK]AAA1.MAR;3

											0066	164	
											0066	165	.SBTTL СООБЩЕНИЯ ОПЕРАТОРУ НА ТЕРМИНАЛ
											0066	166	
											0066	167	NOTLP_MSG:
4E	20	53	49	20	45	43	49	56	45	44	0066	168	.ASCIC @DEVICE IS NOT A LINE PRINTER@
52	50	20	45	4E	49	4C	20	41	20	54	4F	0072	
							52	45	54	4E	49	007E	
											1C	0066	
											0083	169	UNRECOGLP_MSG:
54	53	45	54	20	54	4F	4E	20	4F	44	0083	170	.ASCIC @DO NOT TEST PRINTER TYPE@
50	59	54	20	52	45	54	4E	49	52	50	20	008F	
											45	009B	
											18	0083	
											009C	171	
											009C	172	DASGN_MSG:
53	41	45	44	20	54	4F	4E	4E	41	43	009C	173	.ASCIC @CANNOT DEASSIGN CHANNEL@
4C	45	4E	4E	41	4B	43	20	4E	47	49	53	00A8	
											17	009C	
											00B4	174	
											00B4	175	FMT_DASGN:
4E	20	4C	45	4E	4E	41	4B	43	2F	21	00B4	176	.ASCIC @!/CHANNEL NUMBER: !XL@
											00C0		
											15	00B4	
											00CA	177	
											00CA	178	ASGNCHAN_M1:
20	4C	45	4E	4E	41	4B	43	20	4F	4E	00CA	179	.ASCIC @NO CHANNEL AVAILABLE@
											00D6		
											14	00CA	
											00DF	180	
											00DF	181	ASGNCHAN_M2:
56	45	44	20	4B	43	55	53	20	4F	4E	00DF	182	.ASCIC @NO SUCH DEVICE@
											45	43	49
											0E	00DF	
											00EE	183	
											00EE	184	FMT_ASGN:
41	4E	20	45	43	49	56	45	44	2F	21	00EE	185	.ASCIC @!/DEVICE NAME !AS@
											53	41	21
											4D	00FA	
											11	00EE	
											0100	186	
											0100	187	ASGNCHAN_M4::
4F	4C	4C	41	20	45	43	49	56	45	44	0066	188	.ASCIC @DEVICE ALLOCATED@
											44	45	54
											41	43	
											10	0100	
											0111	189	
											0111	190	ASGNCHAN_M3:
49	53	53	41	20	54	4F	4E	4E	41	43	0066	191	.ASCIC @CANNOT ASSIGN CHANNEL@
											4C	45	4E
											41	4B	43
											20	4E	47
											15	0111	
											0127	192	
											0127	193	GETCHN_M1:
20	54	45	47	20	54	4F	4E	4E	41	43	0066	194	.ASCIC @CANNOT GET CHANNEL INFORMATION@
4F	46	4E	49	20	4C	45	4E	4E	41	4B	43	0133	
											4E	4F	49
											54	41	4D
											52	013F	
											1E	0127	
											0146	195	
											0146	196	
											0146	197	FMT_LPCHAR:

ZZ\_EVAAA1  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ  
СООБЩЕНИЯ ОПЕРАТОРУ НА ТЕРМИНАЛ

12-AUG-1986 10:43:12 MOC ВП MACRO V04-00  
12-AUG-1986 10:42:50 DUA1:[IK]AAA1.MAR;3

```

43 20 52 45 54 4E 49 52 50 2F 21 00 0146 198 .ASCII @!/PRINTER CHARACTERISTICS!/@
49 54 53 49 52 45 54 43 41 52 41 4B 0152
      2F 21 43 41 21 2F 21 53 43 015E
      20 0146
      0000022F 0167 199 LFCHARMSG: .BLKB 200
      022F 200
      022F 201 LFCHARS:
2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 2C 00 022F 202 .ASCII @,,,,,,,,,,,,@-
      2C 023B
      2C 2C 2C 2C 023C 203 @,,,,@-
      2C 2C 2C 2C 0240 204 @,,,,@-
4D 2C 45 53 41 43 52 45 57 4F 4C 2C 0244 205 @,LOWERCASE,MECHFORM,CR@
      52 43 2C 4D 52 4F 46 4B 43 45 0250
      2A 022F
      025A 206
      025A 207 ;+++ SYSTEM ERROR MESSAGES
      025A 208
      025A 209 QIOWRTMSG:
4F 52 52 45 20 4D 45 54 53 59 53 00 025A 210 .ASCII @SYSTEM ERROR ON WRITE@
      45 54 49 52 57 20 4E 4F 20 52 0266
      15 025A
      0270 211
      0270 212 QIORDMSG1:
4F 52 52 45 20 4D 45 54 53 59 53 00 0270 213 .ASCII @SYSTEM ERROR ON READ@
      44 41 45 52 20 4E 4F 20 52 027C
      14 0270
      0285 214
      0285 215 ;
      0285 216 .SBTTL СЕКЦИЯ ГЛОБАЛЬНЫХ ДАННЫХ ПРОГРАММЫ И СООБЩЕНИЯ OUT
      0285 217
      0285 218 CNTL_CHAR_TABLE::
      00 0285 219 .BYTE 0 ; NUL
      01 0286 220 .BYTE 1 ; SOH
      02 0287 221 .BYTE 2 ; STX
      03 0288 222 .BYTE 3 ; ETX
      04 0289 223 .BYTE 4 ; EOT
      05 028A 224 .BYTE 5 ; IDENTIFY TERM
      06 028B 225 .BYTE 6 ; ACK
      07 028C 226 C_BELL:: .BYTE 7 ; RING THE BELL
      08 028D 227 C_BACKSPACE:: .BYTE 8 ; BACKSPACE
      09 028E 228 C_HTAB:: .BYTE 9 ; HORIZONTAL TAB
      0A 028F 229 C_LINEFEED:: .BYTE 10 ; LINE FEED
      0B 0290 230 C_VTAB:: .BYTE 11 ; VERTICAL TAB
      0C 0291 231 C_FORMFEED:: .BYTE 12 ; FORM FEED
      0D 0292 232 C_CR:: .BYTE 13 ; CARRIAGE RETURN
      0E 0293 233 C_SO:: .BYTE 14 ; SO
      0F 0294 234 .BYTE 15 ; SI
      10 0295 235 .BYTE 16 ; DLE
      11 0296 236 .BYTE 17 ; XON
      12 0297 237 .BYTE 18 ; DC2
      13 0298 238 .BYTE 19 ; XOFF
      14 0299 239 .BYTE 20 ; DC4
      15 029A 240 .BYTE 21 ; NAK
      16 029B 241 .BYTE 22 ; SYN
      17 029C 242 .BYTE 23 ; ETB
      18 029D 243 C_CANCEL:: .BYTE 24 ; CAN
      19 029E 244 .BYTE 25 ; EM

```

ZZ\_EVAAA1  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ 12-AUG-1986 10:43:12 МОС ВП MACRO V04-00  
СЕКЦИЯ ГЛОБАЛЬНЫХ ДАННЫХ ПРОГРАММЫ И СООБЩЕНИЯ UUT 12-AUG-1986 10:42:50 DUA1: [IKJAAA1.MAR;3

	1A	029F	245	C_SUB::	.BYTE	26		; SUB							
	1B	02A0	246	C_ESCAPE::	.BYTE	27		; ESCAPE							
	1C	02A1	247		.BYTE	28		; FS							
	1D	02A2	248		.BYTE	29		; GS							
	1E	02A3	249		.BYTE	30		; RS							
	1F	02A4	250		.BYTE	31		; US							
		02A5	251	CHARACTER_TABLE::											
	20	02A5	252	C_SPACE::	.ASCII	\ \									
29	28	27	26	25	24	23	22	21	02A6	253	.ASCII	\!\"#%&'()*\			
	2A	02AF	254	C_STAR::	.ASCII	\*\									
	2B	02B0	255	C_PLUS::	.ASCII	\+\									
	2E	2D	2C	02B1	256	.ASCII	\,-.\								
	2F	02B4	257	C_FORWARDSLASH::	.ASCII	\/\									
	30	02B5	258	C_ZERO::	.ASCII	\0\									
3C	3B	3A	39	38	37	36	35	34	33	32	31	02B6	259	.ASCII	\123456789:;<=>\
	3E	3D	02C2												
	3F	02C4	260	C_QUE_S_MARK::	.ASCII	\?\									
	40	02C5	261	C_AT_SIGN::	.ASCII	\@\									
	41	02C6	262		.ASCII	\A\									
	42	02C7	263	C_B::	.ASCII	\B\									
	44	43	02C8	264	.ASCII	\CD\									
	45	02CA	265	C_E::	.ASCII	\E\									
4B	4A	49	48	47	46	02CB	266	.ASCII	\FGHIJK\						
	4C	02D1	267	C_L::	.ASCII	\L\									
	4D	02D2	268	C_M::	.ASCII	\M\									
	4E	02D3	269	C_N::	.ASCII	\N\									
	52	51	50	4F	02D4	270	.ASCII	\OPQR\							
	53	02D8	271	C_S::	.ASCII	\S\									
	54	02D9	272	C_T::	.ASCII	\T\									
	55	02DA	273	C_U::	.ASCII	\U\									
	57	56	02DB	274	.ASCII	\VW\									
	58	02DD	275	C_X::	.ASCII	\X\									
	59	02DE	276	C_Y::	.ASCII	\Y\									
	5A	02DF	277	C_Z::	.ASCII	\Z\									
	5B	02E0	278		.ASCII	\[\									
	5C	02E1	279	C_BACKSLASH::	.ASCII	@\@									
	5E	5D	02E2	280	.ASCII	\]^_\									
	5F	02E4	281	C_UNDER::	.ASCII	\_\									
4B	4A	49	48	47	46	45	44	43	42	41	7E	02E5	282	.ASCII	\^ABCDEFGHJKLMNPQRSTUVWXYZ[\
57	56	55	54	53	52	51	50	4F	4E	4D	4C	02F1			
	7B	5A	59	58	02FD										
	7C	0301	283		.BYTE	124									; \
	2D	7D	0302	284	.ASCII	\]-\									
	7F	0304	285	C_DELETE::	.BYTE	127									; DELETE
		0305	286												
		0305	287	ILLEGALMSG::											
45	20	4C	41	47	45	4C	4C	49	0A	0D	00	0305	288	.ASCII	<13><10>@ILLEGAL ENTRY@
		59	52	54	4E	0311									
		0F	0305												
			0315	289											
			0315	290											
			0315	291	;+++	NON PRINTING CODE TEST									
			0315	292											
	30	30	30	20	4C	55	4E	0315	293	M_NUL::	.ASCII	/NUL 000/			
	31	30	30	20	48	4F	53	031C	294	M_SOH::	.ASCII	/SOH 001/			
	32	30	30	20	58	54	53	0323	295	M_STX::	.ASCII	/STX 002/			
	33	30	30	20	58	54	45	032A	296	M_ETX::	.ASCII	/ETX 003/			

13.00078-01 33 01

ZZ\_EVAAA1  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ 12-AUG-1986 10:43:12 МОС ВП MACRO V04-00  
СЕКЦИЯ ГЛОБАЛЬНЫХ ДАННЫХ ПРОГРАММЫ И СООБЩЕНИЯ UUT 12-AUG-1986 10:42:50 DUA1:[IK]AAA1.MAR;3

(1)

34	30	30	20	54	4F	45	0331	297	M_EOT::	.ASCII	/EOT	004/	
35	30	30	20	51	4E	45	0338	298	M_ENQ::	.ASCII	/ENQ	005/	
36	30	30	20	4B	43	41	033F	299	M_ACK::	.ASCII	/ACK	006/	
37	30	30	20	4C	45	42	0346	300	M_BEL::	.ASCII	/BEL	007/	
30	31	30	20	20	53	42	034D	301	M__BS::	.ASCII	/BS	010/	
31	31	30	20	20	54	48	0354	302	M__HT::	.ASCII	/HT	011/	
36	31	30	20	20	4F	53	035B	303	M__SO::	.ASCII	/SO	016/	
37	31	30	20	20	49	53	0362	304	M__SI::	.ASCII	/SI	017/	
30	32	30	20	45	4C	44	0369	305	M_DLE::	.ASCII	/DLE	020/	
31	32	30	20	4E	4F	58	0370	306	M_XON::	.ASCII	/XON	021/	
32	32	30	20	32	43	44	0377	307	M_DC2::	.ASCII	/DC2	022/	
33	32	30	20	46	4F	58	037E	308	M_XOF::	.ASCII	/XOF	023/	
34	32	30	20	34	43	44	0385	309	M_DC4::	.ASCII	/DC4	024/	
35	32	30	20	4B	41	4E	038C	310	M_NAK::	.ASCII	/NAK	025/	
36	32	30	20	4E	59	53	0393	311	M_SYN::	.ASCII	/SYN	026/	
37	32	30	20	42	54	45	039A	312	M_ETB::	.ASCII	/ETB	027/	
30	33	30	20	4E	41	43	03A1	313	M_CAN::	.ASCII	/CAN	030/	
31	33	30	20	20	4D	45	03A8	314	M__EM::	.ASCII	/EM	031/	
32	33	30	20	42	55	53	03AF	315	M_SUB::	.ASCII	/SUB	032/	
33	33	30	20	43	53	45	03B6	316	M_ESC::	.ASCII	/ESC	033/	
34	33	30	20	20	53	46	03BD	317	M__FS::	.ASCII	/FS	034/	
35	33	30	20	20	53	47	03C4	318	M__GS::	.ASCII	/GS	035/	
36	33	30	20	20	53	52	03CB	319	M__RS::	.ASCII	/RS	036/	
37	33	30	20	20	53	55	03D2	320	M__US::	.ASCII	/US	037/	
37	37	31	20	4C	45	44	03D9	321	M_DEL::	.ASCII	/DEL	177/	
							03E0	322					
							03E0	323	M_ADDR_TBL::				
				00000315			03E0	324	.ADDRESS		M_NUL		
				0000031C			03E4	325	.ADDRESS		M_SOH		
				00000323			03E8	326	.ADDRESS		M_STX		
				0000032A			03EC	327	.ADDRESS		M_ETX		
				00000331			03F0	328	.ADDRESS		M_EOT		
				00000338			03F4	329	.ADDRESS		M_ENQ		
				0000033F			03F8	330	.ADDRESS		M_ACK		
				00000346			03FC	331	.ADDRESS		M_BEL		
				0000034D			0400	332	.ADDRESS		M__BS		
				00000354			0404	333	.ADDRESS		M__HT		
				0000035B			0408	334	.ADDRESS		M__SO		
				00000362			040C	335	.ADDRESS		M__SI		
				00000369			0410	336	.ADDRESS		M_DLE		
				00000370			0414	337	.ADDRESS		M_XON		
				00000377			0418	338	.ADDRESS		M_DC2		
				0000037E			041C	339	.ADDRESS		M_XOF		
				00000385			0420	340	.ADDRESS		M_DC4		
				0000038C			0424	341	.ADDRESS		M_NAK		
				00000393			0428	342	.ADDRESS		M_SYN		
				0000039A			042C	343	.ADDRESS		M_ETB		
				000003A1			0430	344	.ADDRESS		M_CAN		
				000003A8			0434	345	.ADDRESS		M__EM		
				000003AF			0438	346	.ADDRESS		M_SUB		
				000003B6			043C	347	.ADDRESS		M_ESC		
				000003BD			0440	348	.ADDRESS		M__FS		
				000003C4			0444	349	.ADDRESS		M__GS		
				000003CB			0448	350	.ADDRESS		M__RS		
				000003D2			044C	351	.ADDRESS		M__US		
				000003D9			0450	352	.ADDRESS		M_DEL		
				0454			353						



ZZ\_EVAAA1  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ 12-AUG-1986 10:43:12 МОС ВП MACRO V04-00  
СЕКЦИЯ ГЛОБАЛЬНЫХ ДАННЫХ ПРОГРАММЫ И СООБЩЕНИЯ ЦУТ 12-AUG-1986 10:42:50 DUA1:[IKJAAA1.MAR;3

```

0454 354 ;+++ LNO1 OVERSTRIKE TEST
0454 355
      4E 0D 5A 0454 356 ZN_1:: .ASCII /Z/<13>/N/
4E 20 0D 4E 20 0D 4E 0D 5A 5A 0457 357 ZN_2:: .ASCII /ZZ/<13>/N/<13><32>/N/
20 20 0D 4E 20 0D 4E 0D 5A 5A 5A 045E 358 ZN_3:: .ASCII /ZZZ/<13>/N/<13><32>/N/<13><32><32>/N/
      4E 20 20 20 0D 4E 0476 359 ZN_4:: .ASCII /ZZZZ/<13>/N/<13><32>/N/<13><32><32>/N/<13><32><32><32>/N/
      0A 0D 047C 360 CR_LF:: .ASCII <13><10>
      047E 361
      047E 362
      047E 363
      047E 364 ;+++ TOP OF FORM OPTION TEST
      047E 365
      047E 366 FF_FMT::
46 4F 20 48 54 47 4E 45 4C 2F 21 00 047E 367 .ASCIC @!/LENGTH OF FORM SWITCH TO !AC@-
48 43 54 49 57 53 20 4D 52 4F 46 20 048A
      43 41 21 20 4F 54 20 0496
20 54 45 53 20 53 53 45 52 50 2F 21 049D 368 @!/PRESS SET VFU BUTTON@
      4E 4F 54 54 55 42 20 55 46 56 04A9
      34 047E
      04B3 369
      04B3 370 READY::
      20 59 44 41 45 52 0A 0D 00 04B3 371 .ASCIC <13><10>@READY @
      08 04B3
      04BC 372
54 48 42 33 30 2A 24 21 18 15 12 04BC 373 FF_FILS:: .BYTE 18,21,24,33,36,42,48,51,66,72,84
      04C7 374
      33 20 20 00 04C7 375 M_FF_TBL:: .ASCIC / 3/
      03 04C7
      35 2E 33 00 04CB 376 .ASCIC /3.5/
      03 04CB
      34 20 20 00 04CF 377 .ASCIC / 4/
      03 04CF
      35 2E 35 00 04D3 378 .ASCIC /5.5/
      03 04D3
      36 20 20 00 04D7 379 .ASCIC / 6/
      03 04D7
      37 20 20 00 04DB 380 .ASCIC / 7/
      03 04DB
      38 20 20 00 04DF 381 .ASCIC / 8/
      03 04DF
      35 2E 38 00 04E3 382 .ASCIC /8.5/
      03 04E3
      31 31 20 00 04E7 383 .ASCIC / 11/
      03 04E7
      32 31 20 00 04EB 384 .ASCIC / 12/
      03 04EB
      34 31 20 00 04EF 385 .ASCIC / 14/
      03 04EF
      04F3 386
      04F3 387 ;+++ ECHO TEST OPERATOR PATTERN
      04F3 388
41 4B 43 20 52 45 54 4E 45 0A 0D 00 04F3 389 ECHO_MSG:: .ASCIC <13><10>@ENTER CHARACTERS TO BE ECHOED@-
42 20 4F 54 20 53 52 45 54 43 41 52 04FF
      44 45 4F 48 43 45 20 45 050B
42 20 44 45 57 4F 4C 4C 4F 46 0A 0D 0513 390 <13><10>@FOLLOWED BY CR, CNTL C TO STOP ECHO:@
43 20 4C 54 4E 43 20 2C 52 43 20 59 051F

```

ZZ\_EVAAA1  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ 12-AUG-1986 10:43:12 МОС ВП MACRO V04-00  
СЕКЦИЯ ГЛОБАЛЬНЫХ ДАННЫХ ПРОГРАММЫ И СООБЩЕНИЯ UUT 12-AUG-1986 10:42:50 DUA1: [IKIAAA1.MAR;3

```

48 43 45 20 50 4F 54 53 20 4F 54 20 052B
      3A 4F 0537
      45 04F3
      0539 391
57 4F 48 20 52 45 54 4E 45 0A 0D 00 0539 392 LNO1_MSG:: .ASCIC <13><10>@ENTER HOW MANY LINES TO BE ECHOED@-
20 53 45 4E 49 4C 20 59 4E 41 4D 20 0545
44 45 4F 48 43 45 20 45 42 20 4F 54 0551
42 20 44 45 57 4F 4C 4C 4F 46 0A 0D 055D 393 <13><10>@FOLLOWED BY CR:@
      3A 52 43 20 59 0569
      34 0539
      056E 394
      056E 395 ;+++ ERROR BIT TEST
      056E 396
      056E 397
      056E 398 ERRCOND_MSG::
4E 4F 43 20 52 4F 52 52 45 0A 0D 00 056E 399 .ASCIC <13><10>@ERROR CONDITION CREATED @
54 41 45 52 43 20 4E 4F 49 54 49 44 057A
      20 44 45 0586
      1A 056E
      0589 400
      0589 401 ERRNORCV_MSG::
49 44 4E 4F 43 20 52 4F 52 52 45 00 0589 402 .ASCIC @ERROR CONDITION NOT DETECTED BY DRIVER@
54 45 44 20 54 4F 4E 20 4E 4F 49 54 0595
49 52 44 20 59 42 20 44 45 54 43 45 05A1
      52 45 56 05AD
      26 0589
      05B0 403
      05B0 404
      05B0 405
      05B0 406 ERRRCV_MSG::
4E 4F 43 20 52 4F 52 52 45 0A 0D 00 05B0 407 .ASCIC <13><10>@ERROR CONDITON DETECTED BY DRIVER @
54 43 45 54 45 44 20 4E 4F 49 54 44 05BC
52 45 56 49 52 44 20 59 42 20 44 45 05C8
      20 05D4
      24 05B0
      05D5 408
      05D5 409 ;+++ DATA STORAGE
      05D5 410
      FA0A1F00 05D5 411 TIME:: .LONG -100000000 ; 10 СЕКУНД
      FFFFFFFF 05D9 412 .LONG ^XFFFFFFFF
      00000001 05DD 413 VTLFCNT:: .LONG 1
      00000000 05E1 414 TEMP_STORAGE2:: .LONG
      00000000 05E5 415 TEMP_STORAGE:: .LONG
      00000000 05E9 416 BUFFER_SIZE:: .LONG ; ПАРАМЕТР Q10
      00000000 05ED 417 EVFLAGS:: .LONG ; FOR FUTURE USE
      00000000 05F1 418 BUFSIZE:: .LONG ; FOR FUTURE USE
      00000000 05F5 419 PARAM3:: .LONG ; ПАРАМЕТР Q10
      00000000 05F9 420 PARAM4:: .LONG ; ПАРАМЕТР Q10
      00000000 05FD 421 PARAM5:: .LONG ; ПАРАМЕТР Q10
      00000000 0601 422 PARAM6:: .LONG ; ПАРАМЕТР Q10
      00000000 0605 423 CHANNEL_NUMBER:: .LONG ; ПАРАМЕТР Q10
      0000012C 0609 424 DELAY_TIME:: .LONG 300 ; 3 СЕКУНДЫ
      00000084 060D 425 COLUMN_WIDTH:: .LONG 132 ;
      00000001 0611 426 LP_TYPE:: .LONG 1 ;
      0000000C 0615 427 NUMBER_OF_LINES:: .LONG 12 ;
      00000621 0619 428 ID_STATUS_BLK:: .BLKL 2 ;
      00000000 0621 429 LOG_UNIT:: .LONG ; НАЧАЛЬНЫЙ НОМЕР УСТРОЙСТВА

```

ZZ\_EVAAA1  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ 12-AUG-1986 10:43:12 МОС ВП MACRO V04-00  
СЕКЦИЯ ГЛОБАЛЬНЫХ ДАННЫХ ПРОГРАММЫ И СООБЩЕНИЯ OUT 12-AUG-1986 10:42:50 DUA1:[IK]AAA1.MAR;3

```

00000000 0625 430 HBASEADDR:: .LONG ; HARDWARE P-TABLE
00000000 0629 431 LENGTH: .LONG ; THESE AREA USED
00000000 062D 432 BUFFER: .LONG 12 ;
00000635 0631 433 .ADDRESS CHANINFO ; ЧЕТВЕРНОЕ СЛОВО ОПИСАТЕЛЬ
00000645 0635 434 CHANINFO:: .BLKL 4 ;
00000000 0645 435 DEV_REG: .LONG ;
0649 436 ; NOT USED
0649 437
00000849 0649 438 OUT_BUFFER:: .BLKB 512 ; БУФЕР ДЛЯ ЗАПИСИ
000008CE 0849 439 ECHO_BUFFER:: .BLKB 133 ;
08CE 440
08CE 441 .ALIGN PAGE
0A00 442 xDS_DEVTYP <LP04,LP05,LP06,LP07,LP11,LP14,LP25,LP26,LA180>,<T_LP27,T_LN01>
34 30 50 4C 00 0A00 T_LP04: .ASCIC \LP04\
04 0A00
35 30 50 4C 00 0A05 T_LP05: .ASCIC \LP05\
04 0A05
36 30 50 4C 00 0A0A T_LP06: .ASCIC \LP06\
04 0A0A
37 30 50 4C 00 0A0F T_LP07: .ASCIC \LP07\
04 0A0F
31 31 50 4C 00 0A14 T_LP11: .ASCIC \LP11\
04 0A14
34 31 50 4C 00 0A19 T_LP14: .ASCIC \LP14\
04 0A19
35 32 50 4C 00 0A1E T_LP25: .ASCIC \LP25\
04 0A1E
36 32 50 4C 00 0A23 T_LP26: .ASCIC \LP26\
04 0A23
30 38 31 41 4C 00 0A28 T_LA180: .ASCIC \LA180\
05 0A28
00 0A2E
0000000B 0A2F AL_DEVTYP: .BYTE 0
0000003C 0A33 .LONG xDN
00000051 0A37 .LONG T_LP27
00000A00 0A3B .LONG T_LP04
00000A05 0A3F .LONG T_LP05
00000A0A 0A43 .LONG T_LP06
00000A0F 0A47 .LONG T_LP07
00000A14 0A4B .LONG T_LP11
00000A19 0A4F .LONG T_LP14
00000A1E 0A53 .LONG T_LP25
00000A23 0A57 .LONG T_LP26
00000A2B 0A5B .LONG T_LA180
0A5F 443
00000000 444 .PSECT HEADER

```

```

0000 446 .SBTTL КОД ИНИЦИАЛИЗАЦИИ
0000 447
0000 448 ; ++++++
0000 449 ;
0000 450 ; ФУНКЦИОНАЛЬНОЕ ОПИСАНИЕ::
0000 451 ;
0000 452 ; ЭТА ПРОГРАММА ВЫПОЛНЯЕТ СЛЕДУЮЩИЕ ДЕЙСТВИЯ:
0000 453 ;
0000 454 ; 1. ОТСОЕДИНЯЕТ РАНЕЕ ПОДСОЕДИННОЕ УСТРОЙСТВО
0000 455 ; 2. ПОДСОЕДИНЯЕТ ТЕКУЩЕЕ УСТРОЙСТВО ДЛЯ ПРОВЕРКИ
0000 456 ; 3. ОПРЕДЕЛЯЕТ, ЯВЛЯЕТСЯ ЛИ УСТРОЙСТВО ПЕЧАТАЮЩИМ УСТРОЙСТВОМ
0000 457 ; 4. СООБЩАЕТ ХАРАКТЕРИСТИКИ УСТРОЙСТВА,
0000 458 ; НАЗНАЧЕННОГО ДЛЯ ПРОВЕРКИ.
0000 459 ;
0000 460 ; ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫЗОВА:
0000 461 ;
0000 462 ; ВЫЗЫВАЕТСЯ СУПЕРВИЗОРОМ
0000 463 ;
0000 464 ; ТРЕБОВАНИЯ:
0000 465 ;
0000 466 ; P-ТАБЛИЦА ЗАПОЛНЯЕТСЯ КОРРЕКТНО
0000 467 ; ИНФОРМАЦИОННЫЙ КАНАЛ ВОЗВРАЩЕН КОРРЕКТНО
0000 468 ; БЛОК ДАННЫХ ЗАГОЛОВКА СОСТАВЛЕН КОРРЕКТНО
0000 469 ;
0000 470 ; ++++++
0000 471
0000 472 ;ADS_BGNINIT <R2> ; МАКРОКОМАНДА НАЧАЛА ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ
0000 .SAVE
00000000 .FSECT INITIALIZE, LONG
0000 INITIALIZE:
0004 0000 .WORD ^M<R2>
0002 473 ;ADS_BNPASSO 2x
0A 0000FE00 9F 1D E1 0002 BBC #DSA^V_PASSO, @#DSA^GL_FLAGS, 2x
00000621^EF 00 D0 000A 474 1x: MOVL #0, LOG_UNIT ; ПЕРВЫЙ ПРОХОД
008C 31 0011 475 BRW 5x ; ПОЛУЧИТЬ БАЗОВЫЙ АДРЕС P-ТАБЛИЦЫ
00000621^EF D6 0014 476 2x: INCL LOG_UNIT ; СЛЕДУЮЩЕЕ УСТРОЙСТВО
00000621^EF 0000FE0C^EF D1 001A 477 CMPL DSA^GL_UNITS, LOG_UNIT ; ВСЕ УСТРОЙСТВА ПРОВЕРЕНЫ?
33 13 0025 478 BEQL 3x ; ДА
0027 479 ;ADSASSN_S CHANNEL_NUMBER ; ОТСОЕДИНИТЬ РАНЕЕ ПРИСОЕДИНЕННОЕ УСТ-ВО
7E 00000605^EF 3C 0027 MOVZWL CHANNEL_NUMBER, -(SP)
000102E0^BF 01 FB 002E CALLS #1, G^SYS^ADSASSN
68 50 E8 0035 480 BLBS R0, 5x ; УСПЕШНО
0038 481 ;ADS_ERRSYS_S, LOG_UNIT, DASSN_MSG, DASSN ; ВЫДАТЬ СООБЩЕНИЕ ОБ ОШИБКЕ
00000000^EF DF 0038 PUSHAL DASSN
0000009C^EF DF 003E PUSHAL DASSN_MSG
00000621^EF DD 0044 PUSHL LOG_UNIT
01 DD 004A PUSHL #^ER
004C ;: INITIALIZATION ERROR 1
000100C0 9F 04 FB 004C CALLS #^M, @#DS^ERRSYS
0053 482 ;ADS_ABORT
00010020 9F 6C FA 0053 CALLG (AP), @#DS^ABORT
00000621^EF D7 005A 483 3x: DECL LOG_UNIT ; В СЛУЧАЕ ПОСЛЕДНЕГО ПРОХОДА
0060 484 ;ADS_ENDPASS_G ; ДА
00010010 9F 6E FA 0060 CALLG (SP), @#DS^ENDPASS
0067 485 ;ADSASSN_S CHANNEL_NUMBER ; ОТСОЕДИНИТЬ РАНЕЕ ПРИСОЕДИНЕННОЕ УСТ-ВО
7E 00000605^EF 3C 0067 MOVZWL CHANNEL_NUMBER, -(SP)
000102E0^BF 01 FB 006E CALLS #1, G^SYS^ADSASSN

```

```

22 50 EB 0075 486 BLBS R0,4x ; УСПЕШНО
0078 487 xDS_ERRSYS_S ,LOG_UNIT,DASGN_MSG,DASGN
00000000/EF DF 0078 PUSHAL DASGN
0000009C/EF DF 007E PUSHAL DASGN_MSG
00000621/EF DD 0084 PUSHL LOG_UNIT
02 DD 008A PUSHL #xER
008C ;; INITIALIZATION ERROR 2
000100C0 9F 04 FB 008C CALLS #xM,@#DSxERRSYS
0093 488 xDS_ABORT
00010020 9F 6C FA 0093 CALLG (AP),@#DSxABORT
00000621/EF D4 009A 489 4x: CLRL LOG_UNIT
00A0 490 5x: xDS_GPHARD_S LOG_UNIT,HBASEADDR ; ПОЛУЧИТЬ БАЗОВЫЙ АДРЕС Р-ТАБЛИЦЫ
00000625/EF DF 00A0 PUSHAL HBASEADDR
00000621/EF DD 00A6 PUSHL LOG_UNIT
00010018 9F 02 FB 00AC CALLS #2,@#DSxGPHARD
56 00000625/EF D0 00B3 491 MOVL HBASEADDR,R6 ; БАЗОВЫЙ АДРЕС Р-ТАБЛИЦЫ
00000611/EF D4 00BA 492 CLRL LP_TYPE ; ОЧИСТИТЬ ЯЧЕЙКУ LP_TYPE
00000A00/EF 26 A6 05 29 00C0 493 CMPC3 #5,HPxT_TYPE(R6),T_LP04 ; УСТРОЙСТВО ТИПА LP04?
0A 12 00C9 494 BNEQ 6x ; ПЕРЕЙТИ, ЕСЛИ НЕТ
00000611/EF 04 C8 00CB 495 BISL2 #LP_04,LP_TYPE ; ЕСЛИ ДА, УСТАНОВИТЬ БИТ LP04
00D2 31 00D2 496 BRW 15x ; ПЕРЕЙТИ НА ПРОДОЛЖЕНИЕ
00000A05/EF 26 A6 05 29 00D5 497 6x: CMPC3 #5,HPxT_TYPE(R6),T_LP05 ;
0A 12 00DE 498 BNEQ 7x ;
00000611/EF 08 C8 00E0 499 BISL2 #LP_05,LP_TYPE ;
00BD 31 00E7 500 BRW 15x ;
00000A0A/EF 26 A6 05 29 00EA 501 7x: CMPC3 #5,HPxT_TYPE(R6),T_LP06 ;
0A 12 00F3 502 BNEQ 8x ;
00000611/EF 10 C8 00F5 503 BISL2 #LP_06,LP_TYPE ;
00AB 31 00FC 504 BRW 15x ;
00000A0F/EF 26 A6 05 29 00FF 505 8x: CMPC3 #5,HPxT_TYPE(R6),T_LP07 ;
0A 12 0108 506 BNEQ 9x ;
00000611/EF 20 C8 010A 507 BISL2 #LP_07,LP_TYPE ;
0093 31 0111 508 BRW 15x ;
00000A14/EF 26 A6 05 29 0114 509 9x: CMPC3 #5,HPxT_TYPE(R6),T_LP11 ;
09 12 011D 510 BNEQ 10x ;
00000611/EF 01 C8 011F 511 BISL2 #LP_11,LP_TYPE ;
7F 11 0126 512 BRB 15x ;
00000A19/EF 26 A6 05 29 0128 513 10x: CMPC3 #5,HPxT_TYPE(R6),T_LP14 ;
0D 12 0131 514 BNEQ 11x ;
00000611/EF 00000080 8F C8 0133 515 BISL2 #LP_14,LP_TYPE ;
67 11 013E 516 BRB 15x ;
00000A1E/EF 26 A6 05 29 0140 517 11x: CMPC3 #5,HPxT_TYPE(R6),T_LP25 ;
0D 12 0149 518 BNEQ 12x ;
00000611/EF 00000100 8F C8 014B 519 BISL2 #LP_25,LP_TYPE ;
4F 11 0156 520 BRB 15x ;
00000A23/EF 26 A6 05 29 0158 521 12x: CMPC3 #5,HPxT_TYPE(R6),T_LP26 ;
0D 12 0161 522 BNEQ 13x ;
00000611/EF 00000200 8F C8 0163 523 BISL2 #LP_26,LP_TYPE ;
37 11 016E 524 BRB 15x ;
0000003C/EF 26 A6 05 29 0170 525 13x: CMPC3 #5,HPxT_TYPE(R6),T_LP27 ;
0D 12 0179 526 BNEQ 26x ;
00000611/EF 00000400 8F C8 017B 527 BISL2 #LP_27,LP_TYPE ;
1F 11 0186 528 BRB 15x ;
00000051/EF 26 A6 05 29 0188 529 26x: CMPC3 #5,HPxT_TYPE(R6),T_LN01 ;
0D 12 0191 530 BNEQ 14x ;
00000611/EF 00000800 8F C8 0193 531 BISL2 #LN_01,LP_TYPE ;
07 11 019E 532 BRB 15x ;

```

```

00000611/EF 02 08 01A0 533 14x: BISL2 #LA_180,LP_TYPE
                                01A7 534 15x: @ASSIGN_S HP@Q_DEVICE(R6),CHANNEL_NUMBER ; ПОДСОЕДИНИТЬ ТЕКУЩЕЕ УСТРОЙСТВО
                                7E 7C 01A7 CLRQ -(SP)
00000605/EF 3F 01A9 PUSHAW CHANNEL_NUMBER
00010250/GF 04 FB 01AF 66 7F 01AF PUSHAQ HP@Q_DEVICE(R6)
                                03 50 E9 01B1 535 BLBC R0,16x ; ОШИБКА
                                0091 31 01BB 536 BRW 20x ; ПЕРЕНТИ, ЕСЛИ НЕТ ОШИБКИ
0000008F 50 0000/8F B1 01BE 537 16x: CMPW #SS@_NOIOCHAN,R0 ; ПРОВЕРИТЬ СТАТУС
                                1E 12 01C3 538 BNEQ 17x
                                01C5 539 @DS_ERRSYS_S ,LOG_UNIT,ASGNCHAN_M1 ; ОШИБКА
                                00 DD 01C5 PUSHL #0
000000CA/EF DF 01C7 PUSHAL ASGNCHAN_M1
00000621/EF DD 01CD PUSHL LOG_UNIT
00000003 DD 01D3 PUSHL #@ER
                                01D5 ;; INITIALIZATION ERROR 3
000100C0 9F 04 FB 01D5 CALLS #@M,@#DS@ERRSYS
                                01DC 540 @DS_ABORT
00010020 9F 6C FA 01DC CALLG (AP),@#DS@ABORT
0000008F 50 0000/8F B1 01E3 541 17x: CMPW #SS@_NOSUCHDEV,R0 ; ПРОВЕРИТЬ СТАТУС
                                22 12 01E8 542 BNEQ 18x
                                01EA 543 @DS_ERRSYS_S ,LOG_UNIT,ASGNCHAN_M2,ASGN
                                DF 01EA PUSHAL ASGN
                                DF 01F0 PUSHAL ASGNCHAN_M2
00000621/EF DD 01F6 PUSHL LOG_UNIT
00000004 DD 01FC PUSHL #@ER
                                01FE ;; INITIALIZATION ERROR 4
000100C0 9F 04 FB 01FE CALLS #@M,@#DS@ERRSYS
                                0205 544 @DS_ABORT
00010020 9F 6C FA 0205 CALLG (AP),@#DS@ABORT
0000008F 50 0000/8F B1 020C 545 18x: CMPW #SS@_DEVALLOC,R0 ;
                                1E 12 0211 546 BNEQ 19x ;
                                0213 547 @DS_ERRSYS_S ,LOG_UNIT,ASGNCHAN_M4
                                00 DD 0213 PUSHL #0
00000100/EF DF 0215 PUSHAL ASGNCHAN_M4
00000621/EF DD 021B PUSHL LOG_UNIT
00000005 DD 0221 PUSHL #@ER
                                0223 ;; INITIALIZATION ERROR 5
000100C0 9F 04 FB 0223 CALLS #@M,@#DS@ERRSYS
                                022A 548 @DS_ABORT
00010020 9F 6C FA 022A CALLG (AP),@#DS@ABORT
                                0231 549 19x: @DS_ERRSYS_S ,LOG_UNIT,ASGNCHAN_M3
                                00 DD 0231 PUSHL #0
00000111/EF DF 0233 PUSHAL ASGNCHAN_M3
00000621/EF DD 0239 PUSHL LOG_UNIT
00000006 DD 023F PUSHL #@ER
                                0241 ;; INITIALIZATION ERROR 6
000100C0 9F 04 FB 0241 CALLS #@M,@#DS@ERRSYS
                                0248 550 @DS_ABORT
00010020 9F 6C FA 0248 CALLG (AP),@#DS@ABORT
                                024F 551 20x: ; ПОЛУЧИТЬ ДАННЫЕ О КАНАЛЕ
                                024F 552 @GETCHN_S CHANNEL_NUMBER,LENGTH,BUFFER
                                00 DD 024F PUSHL #0
                                00 DD 0251 PUSHL #0
0000062D/EF 7F 0253 PUSHAQ BUFFER
00000629/EF 3F 0259 PUSHAW LENGTH
00000605/EF 7E 0000/605 3C 025F MOVZWL CHANNEL_NUMBER,-(SP)

```

ZZ\_EVAAA1  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ  
КОД ИНИЦИАЛИЗАЦИИ

12-AUG-1986 10:43:12 MOC ВП MACRO V04-00  
12-AUG-1986 10:42:50 DUA1: [IKJAAA1.MAR;3

PAGE 20  
(1)

```

000104CB^GF 05 FB 0266          CALLS #5,6^SYS^GETCHN
                1E 50 EB 026D 553      BLBS RO,21^      ; УСПЕШНО
                0270 554          ^DS_ERRSYS_S ,LOG_UNIT,GETCHN_M1
                00 DD 0270          PUSHL #0
00000127^EF DF 0272          PUSHAL GETCHN_M1
00000621^EF DD 0278          PUSHL LOG_UNIT
                07 DD 027E          PUSHL #^ER
                0280          ;; INITIALIZATION ERROR 7
000100C0 9F 04 FB 0280          CALLS #^M,@#DS^ERRSYS
                0287 555          ^DS_ABORT
00010020 9F 6C FA 0287          CALLG (AP),@#DS^ABORT
00000639^EF 43 8F 91 028E 556 21^:  CMPB #DC^LP,CHANINFO+4      ; ЭТО УСТРОЙСТВО ПЕЧАТИ?
                1E 13 029E 557      BEQL 22^          ; ДА
                0298 558          ^DS_ERRSYS_S ,LOG_UNIT,NOTLP_MSG
                00 DD 0298          PUSHL #0
                00000066^EF DF 029A          PUSHAL NOTLP_MSG
00000621^EF DD 02A0          PUSHL LOG_UNIT
                08 DD 02A6          PUSHL #^ER
                02AB          ;; INITIALIZATION ERROR 8
000100C0 9F 04 FB 02AB          CALLS #^M,@#DS^ERRSYS
                02AF 559          ^DS_ABORT
00010020 9F 6C FA 02AF          CALLG (AP),@#DS^ABORT
0000060D^EF 0000063B^EF 3C 02B6 560 22^:  MOVZWL CHANINFO+6,COLUMN_WIDTH ; ПОЛУЧИТЬ ДЛИНУ СТРОКИ
01 0000063A^EF 91 02C1 561      CMPB CHANINFO+5,#LP^LP11 ; ОПРЕДЕЛИТЬ ТИП
                10 12 02C8 562      BNEQ 23^          ;
00000611^EF 01 C8 02CA 563      BISL2 #LP_11,LP_TYPE ;
00000615^EF 0C D0 02D1 564      MOVL #12,NUMBER_OF_LINES ;
                66 11 02D8 565      BRB 27^          ;
03 0000063A^EF 91 02DA 566 23^:  CMPB CHANINFO+5,#LP^LA180 ; ПОЛУЧИТЬ ХАРАКТЕРИСТИКИ
                10 12 02E1 567      BNEQ 24^          ; ОПРЕДЕЛИТЬ ТИП
00000611^EF 02 C8 02E3 568      BISL2 #LA_180,LP_TYPE ; НЕ LA180
00000615^EF 0C D0 02EA 569      MOVL #12,NUMBER_OF_LINES ; ИСПОЛЬЗУЕТСЯ ТЕСТОМ ПУТЕЙ ДАННЫХ
                4D 11 02F1 570      BRB 27^          ; ПОЛУЧИТЬ ХАРАКТЕРИСТИКИ
                02F3 571 ;
                02F3 572 ; ВАРИАНТ ДЛЯ DMF32 SD/PG 9/16/82
                02F3 573 ;
0A 0000063A^EF 91 02F3 574 24^:  CMPB CHANINFO+5,#DT^DMF32 ;
                26 12 02FA 575      BNEQ 25^          ;
00000611^EF 00000800 8F D3 02FC 576      BITL #LN_01,LP_TYPE ; Тестируется LN01?
                19 12 0307 577      BNEQ 25^          ;
                0309 578          ;
00000615^EF 0C D0 0309 579      MOVL #12,NUMBER_OF_LINES ;
00000611^EF 02 D3 0310 580      BITL #LA_180,LP_TYPE ;
                27 12 0317 581      BNEQ 27^          ;
00000611^EF 01 C8 0319 582      BISL2 #LP_11,LP_TYPE ;
                1E 11 0320 583      BRB 27^          ; ПОЛУЧИТЬ ХАРАКТЕРИСТИКИ
                0322 584 25^:  ^DS_ERRSYS_S ,LOG_UNIT,UNRECOGLP_MSG
                00 DD 0322          PUSHL #0
00000083^EF DF 0324          PUSHAL UNRECOGLP_MSG
                00000611^EF 00 032A          PUSHL LOG_UNIT
                09 DD 0330          PUSHL #^ER
                0332          ;; INITIALIZATION ERROR 9
000100C0 9F 04 FB 0332          CALLS #^M,@#DS^ERRSYS
                0339 585          ^DS_ABORT
00010020 9F 6C FA 0339          CALLG (AP),@#DS^ABORT
54 0000063D^EF D0 0340 586 27^:  MOVL CHANINFO+8,R4 ;
                0347 587          ^DS_CVTREG_S #23,R4,LPCHARS,LPCHARMSG,#2

```

13.00078-01 33 01

ZZ\_EVAAA1

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ 12-AUG-1986 10:43:12 МОС ВП МСРО V04-00  
КОД ИНИЦИАЛИЗАЦИИ 12-AUG-1986 10:42:50 DUA1: [IK]AAA1.MAR;3

5.3

■ ■

◀▶E000 ■

```

PUSHL #0 0370 588
      00000167 8F DD 0370
      00000146 EF 9F 0376
000100E0 9F 02 FB 037C
      0383
      0383
0001005B 9F 6E FA 0383
      04 038A
      00000000

```

```

      DDS PRINTB_S FMT_LPCHAR, #LPCHARMSG ; НАПЕЧАТАТЬ ОСНОВ. СООБЩ. ОБ ОШИБКУ
PUSHL #LPCHARMSG
PUSHAB FMT_LPCHAR
CALLS #DQN,@#DS%DPRINTB
589      DDS_ENDINIT ; МАКРОКОМАНДА ОКОНЧАНИЯ ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ
INITIALIZE_X:
CALLG (SP),@#DS%DBREAK
RET
.RESTORE

```



ZZ\_EVAAA1  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ  
КОД ОЧИСТКИ

12-AUG-1986 10:43:12 МОС ВП MACRO V04-00  
12-AUG-1986 10:42:50 DUA1: [IKJAAA1.MAR;3

```

0000 591 .SBTTL КОД ОЧИСТКИ
0000 592
0000 593 ;+++++
0000 594 ;
0000 595 ; ФУНКЦИОНАЛЬНОЕ ОПИСАНИЕ:
0000 596 ;
0000 597 ; ОТСОЕДИНЕНИЕ ТЕКУЩЕГО УСТРОЙСТВА ПОСЛЕ ТЕСТИРОВАНИЯ
0000 598 ;
0000 599 ;+++++
0000 600
0000 601      %DS %B%NCLEAN      ; МАКРОКОМАНДА НАЧАЛА СЕКЦИИ ОЧИСТКИ
0000      .SAVE
00000000      .PSECT CLEANUP, LONG
0000      CLEAN_UP:
0000 0000      .WORD ^M<>
0002 602      %DASSGN_S CHANNEL_NUMBER      ; ОТСОЕДИНИТЬ КАНАЛ
7E 00000605/EF 3C 0002      MOVZWL CHANNEL_NUMBER, -(SP)
000102E0/6F 01 FB 0009      CALLS #1, G^SYS%DASSGN
      1B 50 E8 0010 603      BLBS R0, 10%      ; УСПЕШНО
      0013 604      %DS %ERRSYS_S , LOG_UNIT, DASSGN_MSG, DASSGN
      0013      PUSHAL DASSGN
      0019      PUSHAL DASSGN_MSG
      001F      PUSHL LOG_UNIT
      01 DD 0025      PUSHL #%ER
      0027      ;; INITIALIZATION ERROR 1
000100C0 9F 04 FB 0027      CALLS #%M, @%DS%ERRSYS
      01 002E 605 10%:      NOP
      002F 606      %DS %ENDCLEAN      ; МАКРОКОМАНДА ОКОНЧАНИЯ СЕКЦИИ ОЧИСТКИ
00010058 9F 6E FA 002F      CLEAN_UP_X:
      04 0036      CALLG (SP), @%DS%BREAK
00000000      RET
      .RESTORE

```

ZZ EVA001  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ  
КОД ЗАКЛЮЧИТЕЛЬНОГО СООБЩЕНИЯ

12-AUG-1986 10:43:12 MOC ВП MACRO V04-00  
12-AUG-1986 10:42:50 DUA1:[IK]AAA1.MAR;3

```

0000 608 .SBTTL КОД ЗАКЛЮЧИТЕЛЬНОГО СООБЩЕНИЯ
0000 609
0000 610 ;+++++
0000 611 ;
0000 612 ; ФУНКЦИОНАЛЬНОЕ ОПИСАНИЕ:
0000 613 ;
0000 614 ; ВЫЗОВ ЗАКЛЮЧИТЕЛЬНОГО СООБЩЕНИЯ НЕ ЯВЛЯЕТСЯ
0000 615 ; ОБЯЗАТЕЛЬНЫМ ДЛЯ МАКРОКОМАНДЫ %DS_HEADER
0000 616 ;
0000 617 ;+++++
0000 618
0000 619      %DS_BGN SUMMARY ; МАКРОКОМАНДА НАЧАЛА СЕКЦИИ ЗАКЛЮЧИТЕЛЬНОГО СООБЩЕНИЯ
0000      .SAVE
00000000      .PSECT SUMMARY, LONG
0000      SUMMARY:
0000 0000      .WORD ^M<>
0002 620      %DS_END SUMMARY ; МАКРОКОМАНДА ОКОНЧАНИЯ СЕКЦИИ ЗАКЛЮЧИТЕЛЬНОГО СООБЩЕНИЯ
0002      SUMMARY_X:
00010058 9F 6E FA 0002      CALLG (SP), @#DS%BREAK
0004 0009      RET
00000000      .RESTORE

```

```

0000 622 .SVTTL ПОДПРОГРАММА ПЕЧАТИ
0000 623
0000 624 ;+++++
0000 625 ;
0000 626 ; ФУНКЦИОНАЛЬНОЕ ОПИСАНИЕ:
0000 627 ;
0000 628 ; ЭТА СЕКЦИЯ СОДЕРЖИТ ПОДПРОГРАММЫ,
0000 629 ; ПЕЧАТАЮЩИЕ СООБЩЕНИЯ ОБ ОШИБКЕ С
0000 630 ; РАЗЛИЧНОЙ СТЕПЕНЬЮ ПОДРОБНОСТИ
0000 631 ;
0000 632 ;+++++
0000 633 DASN:  QDS_BGNMESSAGE ; НАЧАТЬ СООБЩЕНИЕ
0000 0000 .WORD ^M<>
0002 634      QDS_PRINTB_S FMT_DASN,CHANNEL_NUMBER ; ПЕЧАТЬ ОСНОВ. ИНФОРМАЦИИ ОБ ОШИБКУ
00000605/EF DD 0002      PUSHL CHANNEL_NUMBER
000000B4/EF 9F 0008      PUSHAB FMT_DASN
000100E0 9F 02 FB 000E      CALLS #QDN,@#DSQPRINTB
0015 635      QDS_ENDMESSAGE ; ЗАКОНЧИТЬ СООБЩЕНИЕ
04 0015      RET
0016 636 ASN:   QDS_BGNMESSAGE ; НАЧАТЬ СООБЩЕНИЕ
0000 0016 .WORD ^M<>
62 DF 0018 637      PUSHAL HFQ_DEVICE(R2) ; ПЕЧАТЬ ОСНОВНОЙ ИНФОРМАЦИИ ОБ ОШИБКЕ
001A 638      QDS_PRINTB_S FMT_ASN
000000EE/EF 9F 001A      PUSHAB FMT_ASN
000100E0 9F 01 FB 0020      CALLS #QDN,@#DSQPRINTB
0027 639      QDS_ENDMESSAGE ; ЗАКОНЧИТЬ СООБЩЕНИЕ
04 0027      RET

```

```

0028 641 .SBTTL ПОДПРОГРАММА QIOW_S ВЫЗОВА СИСТЕМНОГО ОБСЛУЖИВАНИЯ
0028 642
0000 643 .PSECT
0000 644
0000 645 :+++++
0000 646 :
0000 647 : ФУНКЦИОНАЛЬНОЕ ОПИСАНИЕ: OUTPUT
0000 648 :
0000 649 : ЭТА ПОДПРОГРАММА ГЕНЕРИРУЕТ ВЫЗОВ QIOW_S.
0000 650 : ЕСЛИ СИСТЕМА НЕИСПРАВНА, ВЫДАЕТСЯ СООТВЕТСТВУЮЩЕЕ СООБЩЕНИЕ
0000 651 : И ПРОГРАММА АВАРИЙНО ЗАВЕРШАЕТСЯ.
0000 652 : ПРОГРАММА ПРОИЗВОДИТ ВЫВОД ИЗ ОБЛАСТИ ПАМЯТИ OUT_BUFFER.
0000 653 : РАЗМЕР ЭТОГО БУФЕРА УКАЗАН В ЯЧЕЙКЕ BUFFER_SIZE.
0000 654 :
0000 655 :
0000 656 :
0000 657 :
0000 658 : ПАРАМЕТРЫ ВЫВОДА:
0000 659 :
0000 660 : АДРЕС БУФЕРА
0000 661 : ДЛИНА БУФЕРА В БАЙТАХ
0000 662 : НОМЕР КАНАЛА В ДЛИННОМ СЛОВЕ (ПОЛУЧАЕТСЯ ИЗ QASSIGN)
0000 663 :
0000 664 :
0000 665 :
0000 666 :
0000 667 :
0000 668 :+++++
0000 669
0000 670
0000 671 OUTPUT::
0000 672 QIOW_S ; WRITEVBLK
0000 673 CHAN=CHANNEL_NUMBER, ;
0000 674 FUNC=#IDQ_WRITEVBLK, ;
0000 675 IOSB=IO_STATUS_BLK, ; RETURN STATUS
0000 676 P1=OUT_BUFFER, ; BUFFER ADDR
0000 677 P2=BUFFER_SIZE, ; MSG LENGTH
0000 678 P4=C_SPACE ; CARRIAGE CONTROL
7E 7C 0000 CLRQ -(SF)
000002A5/EF DD 0002 PUSHL C_SPACE
00 DD 0008 PUSHL #0
000005E9/EF DD 000A PUSHL BUFFER_SIZE
00000649/EF DF 0010 PUSHAL OUT_BUFFER
7E 7C 0016 CLRQ -(SF)
00000619/EF 7F 0018 PUSHAQ IO_STATUS_BLK
7E 0000/8F 3C 001E MOVZWL #IDQ_WRITEVBLK,-(SF)
7E 00000605/EF 3C 0023 MOVZWL CHANNEL_NUMBER,-(SF)
00 DD 002A PUSHL #0
00010200/8F 0C FB 002C CALLS #12,6^SYSQIOW
29 50 EB 0033 679 BLBS R0,100Q ; SUCCESSFUL
00000619/EF 0000/8F B1 0036 680 CMPW #SSQ_CONTROLD,IO_STATUS_BLK
1E 13 003F 681 BEQL 100Q ; OK TO CONT
0041 682 QDS_ERRSYS_S ,LOG_UNIT,QIOWRTMSG
00 DD 0041 PUSHL #0
0000025A/EF DF 0043 PUSHAL QIOWRTMSG
00000621/EF DD 0049 PUSHL LOG_UNIT
02 DD 004F PUSHL #QER

```

13.00078-01 33 01

ZZ\_EVAAA1  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ 12-AUG-1986 10:43:12 МОС ВП MACRO V04-00  
ПОДПРОГРАММА QDIOW\_S ВЫЗОВА СИСТЕМНОГО ОБСЛУЖИВАНИЯ 12-AUG-1986 10:42:50 DUA1:[IKJAAA1.MAR;3

PAGE 26

(1)

```
00010000 9F 04 FB 0051          ; ; GLOBAL ERROR 2
00010020 9F 6C FA 0051          CALLS  #QDM,@#DSQERRSYS
00010020 9F 6C FA 0058 683      QDS_ABORT
00010020 9F 6C FA 0058          CALLG  (AP),@#DSQABORT
00010020 9F 6C FA 005F 684
00010020 9F 6C FA 005F 685
00010020 9F 6C FA 05 005F 686 100Q: RSB          : BO3BPAT
```



13.00078-01 33 01

ZZ\_EVAAA1  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ 12-AUG-1986 10:43:12 МОС ВП MACRO V04-00  
ПОДПРОГРАММА %QIDW\_5 ВЫЗОВА СИСТЕМНОГО САМООБСЛУЖИВАНИЯ 12-AUG-1986 10:42:50 DUA1:[IK]AAA1.MAR;3

PAGE 28

(1)

```
      .OOBA      716  
      OOBA      717 %DS_ENDMOD      ; МАКРОКОМАНДА ОКОНЧАНИЯ ПРОГРАММНОГО МОДУЛЯ.  
      00000000      .PSECT %TSTCNT,NOEXE,NOWRT,OVR, LONG  
      00000000      0000      .LONG %TN  
      0004      718  
      0004      719 .END
```

ZZ\_EVAAA1  
SYMBOL TABLE

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ

12-AUG-1986 10:43:12 MDC ВП MACRO V04-00  
12-AUG-1986 10:42:50 DUA1:[IK]AAA1.MAR;3

PAGE 29

(1)

ααE	= 00000001			CHARACTER_TABLE	000002A5	RG	09
ααM	= 00000004			CLEAN_UP	00000000	R	0B
ααN	= 00000000			CLEAN_UP_X	0000002F	R	0B
ααS	= FFFFFFFF			CNTL_CHAR_TABLE	000002B5	RG	09
ααT1	= 00000001			COLUMN_WIDTH	0000060D	RG	09
αENV	= 00000002	G		CR_LF	0000047C	RG	09
αER	= 00000003			C_AT_SIGN	000002C5	RG	09
αMO	= 00000001	G		C_B	000002C7	RG	09
αST	= 00000000			C_BACKSLASH	000002E1	RG	09
αTN	= 00000000			C_BACKSPACE	0000028D	RG	09
AL_DEVTYP	00000A2F	R	09	C_BELL	0000028C	RG	09
ASGN	00000016	R	02	C_CANCEL	0000029D	RG	09
ASGNCHAN_M1	000000CA	R	09	C_CR	00000292	RG	09
ASGNCHAN_M2	000000DF	R	09	C_DELETE	00000304	RG	09
ASGNCHAN_M3	00000111	R	09	C_E	000002CA	RG	09
ASGNCHAN_M4	00000100	RG	09	C_ESCAPE	000002A0	RG	09
A_HEADEND	00000058	R	04	C_FORMFEED	00000291	RG	09
BIT...	= 00000005			C_FORWARDSLASH	000002B4	RG	09
BIT0	= 00000001			C_HTAB	0000028E	RG	09
BIT1	= 00000002			C_L	000002D1	RG	09
BIT10	= 00000400			C_LINEFEED	0000028F	RG	09
BIT11	= 00000800			C_M	000002D2	RG	09
BIT12	= 00001000			C_N	000002D3	RG	09
BIT13	= 00002000			C_PLUS	000002B0	RG	09
BIT14	= 00004000			C_QUES_MARK	000002C4	RG	09
BIT15	= 00008000			C_S	000002D8	RG	09
BIT16	= 00010000			C_SO	00000293	RG	09
BIT17	= 00020000			C_SPACE	000002A5	RG	09
BIT18	= 00040000			C_STAR	000002AF	RG	09
BIT19	= 00080000			C_SUB	0000029F	RG	09
BIT2	= 00000004			C_T	000002D9	RG	09
BIT20	= 00100000			C_U	000002DA	RG	09
BIT21	= 00200000			C_UNDER	000002E4	RG	09
BIT22	= 00400000			C_VTAB	00000290	RG	09
BIT23	= 00800000			C_X	000002DD	RG	09
BIT24	= 01000000			C_Y	000002DE	RG	09
BIT25	= 02000000			C_Z	000002DF	RG	09
BIT26	= 04000000			C_ZERO	000002B5	RG	09
BIT27	= 08000000			DASGN	00000000	R	02
BIT28	= 10000000			DASGN_MSG	0000009C	R	09
BIT29	= 20000000			DCα_LP	= 00000043		
BIT3	= 00000008			DEFAULT	= 00000000	G	
BIT30	= 40000000			DELAY_TIME	00000609	RG	09
BIT31	= 80000000			DEV_REG	00000645	R	09
BIT4	= 00000010			DISPATCH	00000000	R	07
BIT5	= 00000020			DSαABORT	00010020	G	
BIT6	= 00000040			DSαASKDATA	00010080	G	
BIT7	= 00000080			DSαASKLGCL	00010098	G	
BIT8	= 00000100			DSαASKSTR	000100A0	G	
BIT9	= 00000200			DSαASKVLD	00010088	G	
BUFFER	0000062D	R	09	DSαASLADR	00010090	G	
BUFFER_SIZE	000005E9	RG	09	DSαATTACH	000101A8	G	
BUFSIZE	000005F1	RG	09	DSαBGNSUB	00010030	G	
CEP_FUNCTIONAL	= 00000000			DSαBRANCH	000100A8	G	
CEP_REPAIR	= 00000001			DSαBREAK	00010058	G	
CHANINFO	00000635	RG	09	DSαCANWAIT	00010070	G	
CHANNEL_NUMBER	00000605	RG	09	DSαCHANNEL	00010180	G	



ZZ\_EVAAA1  
SYMBOL TABLE

ФАМЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ

12-AUG-1986 10:43:12 MDC ВП MACRO V04-00  
12-AUG-1986 10:42:50 DUA1: [IKJAAA1.MAR;3

DSACKLOOP	00010040	G	DSABGT_DEVNAM	0000FE5C		
DSACLREVC	00010168	G	DSADV_PASSO	= 0000001D		
DSACNTRLC	00010078	G	DTA_DMF32	= 0000000A		
DSACVTREG	000100B0	G	ECHO_BUFFER	00000849	RG	09
DSACNDPASS	00010010	G	ECHO_MSG	000004F3	RG	09
DSACENDSUB	00010038	G	ENVAM_DOMAIN	= 00000002		
DSACERRDEV	000100C8	G	ENVAM_LEVEL	= 00000001		
DSACERRHARD	000100D0	G	ENVAM_SUPER	= 000003FC		
DSACERRPREP	00010108	G	ENVAS_DOMAIN	= 00000001		
DSACERRSOFT	000100D8	G	ENVAS_LEVEL	= 00000001		
DSACERRSYS	000100C0	G	ENVAS_SUPER	= 00000008		
DSACESCAPE	00010050	G	ENVAV_DOMAIN	= 00000001		
DSACGETBUF	00010120	G	ENVAV_LEVEL	= 00000000		
DSACGETMEM	00010130	G	ENVAV_SUPER	= 00000002		
DSACGP HARD	00010018	G	ENVX_CPU	= 00000000		
DSACHELP	000101B0	G	ENVX_FUNCTIONAL	= 00000000		
DSACINITSCB	00010170	G	ENVX_REPAIR	= 00000001		
DSACINLOOP	00010048	G	ENVX_SUPER	= 00000001		
DSACLOAD	00010198	G	ENVX_SYSTEM	= 00000001		
DSACMAPDBGBLOCK	00010118	G	ERRCOND_MSG	0000056E	RG	09
DSACMMOFF	00010158	G	ERRNORCV_MSG	000005B9	RG	09
DSACMMON	00010150	G	ERRRCV_MSG	000005B0	RG	09
DSACMOVPHY	00010148	G	EVFLAGS	000005ED	RG	09
DSACMOVVRT	00010140	G	FF_FILS	000004BC	RG	09
DSACPARSE	000100B8	G	FF_FMT	0000047E	RG	09
DSACPRINTB	000100E0	G	FMT_ASGN	000000EE	R	09
DSACPRINTF	000100F0	G	FMT_DASGN	000000B4	R	09
DSACPRINTS	000100F8	G	FMT_LPCHAR	00000146	R	09
DSACPRINTSIG	00010100	G	FUNCTIONAL	= 00000002	G	
DSACPRINTX	000100E8	G	GETCHN_M1	00000127	R	09
DSACPROBE	000101A0	G	GROUP	= 00000003	G	
DSACRELBUF	00010128	G	HBASEADDR	00000625	RG	09
DSACRELMEM	00010138	G	HPXA_DEPENDENT	00000032		
DSACSETIFL	00010178	G	HPXA_DEVICE	00000018		
DSACSETMAP	00010188	G	HPXA_DVA	0000001C		
DSACSETPRIEXV	00010110	G	HPXA_LINK	00000020		
DSACSETVEC	00010160	G	HPXB_DRIVE	0000000B		
DSACSHOCHAN	00010190	G	HPXB_FLAGS	0000000A		
DSACSUMMARY	00010028	G	HPXM_ALLOC	= 00000001		
DSACWAITMS	00010060	G	HPXD_DEVICE	00000000		
DSACWAITUS	00010068	G	HPXT_DEVICE	0000000C		
DSACAL_APTMAIL	0000FE00		HPXT_TYPE	00000026		
DSACAT_APTTXT	0000FA00		HPXW_SIZE	0000000B		
DSACGL_APTCOM	0000FE04		HPXW_VECTOR	00000024		
DSACGL_DEVLEN	0000FE58		ILLEGALMSG	00000305	RG	09
DSACGL_ERRNO	0000FE44		INITIALIZE	00000000	R	0A
DSACGL_EVENT	0000FE48		INITIALIZE_X	000003B3	R	0A
DSACGL_FLAGS	0000FE00		IDX_WRITEPBLK	*****	X	01
DSACGL_MSGTYP	0000FE40		IDX_WRITEVBLK	*****	X	01
DSACGL_PASSES	0000FE08		ID_STATUS_BLK	00000619	RG	09
DSACGL_PASSNO	0000FE54		LXA_CCP	00000040	R	04
DSACGL_SECTNO	0000FE10		LXA_DEVP	0000001C	R	04
DSACGL_SID	0000FE14		LXA_DREG	00000024	R	04
DSACGL_SUBTNO	0000FE4C		LXA_ICP	0000003C	R	04
DSACGL_TESTNO	0000FE50		LXA_LASTAD	00000014	R	04
DSACGL_UNITS	0000FE0C		LXA_NAME	00000008	R	04
DSACGL_MSGPTR	0000FE68		LXA_REPP	00000044	R	04

ZZ\_EVAAA1  
SYMBOL TABLE

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ

12-AUG-1986 10:43:12 MOC ВП MACRO V04-00  
12-AUG-1986 10:42:50 DUA1:[IK]AAA1.MAR;3

PAGE 31

(1)

L0A_SECNAM	00000050	R	04	M_NAK	0000038C	RG	09
L0A_STATAB	00000048	R	04	M_NUL	00000315	RG	09
L0A_TSTCNT	00000054	R	04	M_SOH	0000031C	RG	09
L0L_DTP	00000018	R	04	M_STX	00000323	RG	09
L0L_ENVIRON	00000004	R	04	M_SUB	000003AF	RG	09
L0L_ERRTYP	0000004C	R	04	M_SYN	00000393	RG	09
L0L_HEADLENGTH	00000000	R	04	M_XOF	0000037E	RG	09
L0L_REV	0000000C	R	04	M_XON	00000370	RG	09
L0L_UNIT	00000020	R	04	M__BS	0000034D	RG	09
L0L_UPDATE	00000010	R	04	M__EM	000003A8	RG	09
LA180	= 00000009			M__FS	000003BD	RG	09
LASTAD	00000000	R	05	M__GS	000003C4	RG	09
LA_180	= 00000002	G		M__HT	00000354	RG	09
LENGTH	00000629	R	09	M__RS	000003CB	RG	09
LN010K_LEN	00000032			M__SI	00000362	RG	09
LN01_MSG	00000539	RG	09	M__SO	0000035B	RG	09
LN_01	= 00000800	G		M__US	000003D2	RG	09
LOG_UNIT	00000621	RG	09	NOTLP_MSG	00000066	R	09
LP0_LA180	= 00000003			NUMBER_OF_LINES	00000615	RG	09
LP0_LF11	= 00000001			OUTPUT	00000000	RG	01
LP04	= 00000001			OUTPUT_PASSALL	00000060	RG	01
LP05	= 00000002			OUT_BUFFER	00000649	RG	09
LP06	= 00000003			PARAM_ATDEF	= 00000010		
LP07	= 00000004			PARAM_ATHI	= 00000008		
LP11	= 00000005			PARAM_ATLO	= 00000004		
LP14	= 00000006			PARAM_NQDEF	= 00000002		
LP25	= 00000007			PARAM_VATDEF	= 00000004		
LP26	= 00000008			PARAM_VATHI	= 00000003		
LP270K_LEN	00000032			PARAM_VATLO	= 00000002		
LPCHARMSG	00000167	R	09	PARAM_VNODEF	= 00000001		
LPCHARS	0000022F	R	09	PARAM_BIN	= 00000002		
LP_04	= 00000004	G		PARAM_DEC	= 0000000A		
LP_05	= 00000008	G		PARAM_HEX	= 00000010		
LP_06	= 00000010	G		PARAM_NO	= 00000000		
LP_07	= 00000020	G		PARAM_OCT	= 00000008		
LP_11	= 00000001	G		PARAM_YES	= 00000001		
LP_14	= 00000080	G		PARAM3	000005F5	RG	09
LP_25	= 00000100	G		PARAM4	000005F9	RG	09
LP_26	= 00000200	G		PARAM5	000005FD	RG	09
LP_27	= 00000400	G		PARAM6	00000601	RG	09
LP_TYPE	00000611	RG	09	PD0_ADD	= 0000008A		
L_TSTCNT	00000000	R	06	PD0_CASE	= 0000008C		
MANUAL	= 00000003	G		PD0_COMPLEMENT	= 00000089		
M_ACK	0000033F	RG	09	PD0_DECIMAL	= 00000082		
M_ADDR_TBL	000003E0	RG	09	PD0_END	= 00000081		
M_BEL	00000346	RG	09	PD0_FETCH	= 00000087		
M_CAN	000003A1	RG	09	PD0_HEXADecimal	= 00000084		
M_DC2	00000377	RG	09	PD0_LITERAL	= 00000086		
M_DC4	00000385	RG	09	PD0_LOGICAL	= 0000008B		
M_DEL	000003D9	RG	09	PD0_NAME	= 0000008D		
M_DLE	00000369	RG	09	PD0_OCTAL	= 00000083		
M_ENQ	00000338	RG	09	PD0_START	= 00000080		
M_EDT	00000331	RG	09	PD0_STORE	= 00000088		
M_ESC	000003B6	RG	09	PD0_STRING	= 00000085		
M_ETB	0000039A	RG	09	PTD0M_CONTROLLER	= 00000002		
M_ETX	0000032A	RG	09	PTD0M_DEVICE	= 00000003		
M_FF_TBL	000004C7	RG	09	PTD0M_ENDDEVICE	= 0000001B		

ZZ\_EVAAA1  
SYMBOL TABLE

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ

12-AUG-1986 10:43:12 MOC ВП MACRO V04-00  
12-AUG-1986 10:42:50 DUA1: [IK]AAA1.MAR:3

PTDQM_INHERIT	=	00000018	.		SYSQULKFAG	00010460	G	
PTDQM_INHERITED	=	00000008			SYSQULWSET	00010468	G	
PTDQM_INHERIT_CON	=	00000010			SYSQUNWIND	00010470	G	
PTDQM_INHERIT_PRE	=	00000008			SYSQWAITFR	00010478	G	
PTDQM_NAME	=	00000004			SYSQWFLAND	00010488	G	
PTDQM_UNIT	=	00000001			SYSQWFLOR	00010490	G	
PTDQV_CONTROLLER	=	00000001			S_DEFAULT	00000000	R	09
PTDQV_INHERIT_CON	=	00000004			S_FUNCTIONAL	00000016	R	09
PTDQV_INHERIT_PRE	=	00000003			S_MANUAL	00000021	R	09
PTDQV_NAME	=	00000002			S_QUALIFICATION	00000008	R	09
PTDQV_UNIT	=	00000000			TEMP_STORAGE	000005E5	RG	09
QIORDMSG1		00000270	R	09	TEMP_STORAGE2	000005E1	RG	09
QIQWRTMSG		0000025A	R	09	TIME	000005D5	RG	09
QUALIFICATION	=	00000001	G		T_LA180	00000A28	R	09
READY		000004B3	RG	09	T_LN01	00000051	R	09
SECTION		00000028	R	09	T_LP04	00000A00	R	09
SEP_FUNCTIONAL	=	00000002			T_LP05	00000A05	R	09
SEP_REPAIR	=	00000003			T_LP06	00000A0A	R	09
SIZ...	=	00000001			T_LP07	00000A0F	R	09
SSQ_CONTROLD	*****		X	01	T_LP11	00000A14	R	09
SSQ_DEVALLOC	*****		X	0A	T_LP14	00000A19	R	09
SSQ_NOIOCHAN	*****		X	0A	T_LP25	00000A1E	R	09
SSQ_NOSUCHDEV	*****		X	0A	T_LP26	00000A23	R	09
STATISTIC		00000066	R	09	T_LP27	0000003C	R	09
SUMMARY		00000000	R	0C	T_NAME	00000058	R	04
SUMMARY_X		00000002	R	0C	UNRECOGLP_MSG	00000083	R	09
SYSQALLOC		00010238	G		VTLFCNT	000005DD	RG	09
SYSQASCTIM		00010248	G		ZN_1	00000454	RG	09
SYSQASSIGN		00010250	G		ZN_2	00000457	RG	09
SYSQBINTIM		00010258	G		ZN_3	0000045E	RG	09
SYSQCANCEL		00010260	G		ZN_4	0000046A	RG	09
SYSQCANTIM		00010268	G					
SYSQCLOSE		000105B8	G					
SYSQCLREF		00010298	G					
SYSQCONNECT		000105C0	G					
SYSQDALLOC		000102D8	G					
SYSQDASSGN		000102E0	G					
SYSQDISCONNECT		000105D0	G					
SYSQFAO		00010350	G					
SYSQFAOL		00010358	G					
SYSQGET		00010580	G					
SYSQGETCHN		000104C8	G					
SYSQGETTIM		00010378	G					
SYSQLKWSSET		000103A0	G					
SYSQNUMTIM		000103B8	G					
SYSQOFEN		00010608	G					
SYSQQIO		000103C8	G					
SYSQQIQW		00010200	G					
SYSQREAD		00010590	G					
SYSQREADEF		000103D0	G					
SYSQSETAST		000103F8	G					
SYSQSETEF		00010400	G					
SYSQSETIMR		00010420	G					
SYSQSETPRI		00010428	G					
SYSQSETPRT		00010430	G					
SYSQSETRWM		00010438	G					
SYSQSETSWM		00010448	G					

+-----+  
! PSECT SYNOPSIS !  
+-----+

PSECT NAME	ALLOCATION	PSECT NO.	ATTRIBUTES
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
. BLANK .	0000000A ( 186.)	01 ( 1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
HEADER	00000028 ( 40.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC PAGE
ABS	00010610 (67088.)	03 ( 3.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
HEADER	00000076 ( 118.)	04 ( 4.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC PAGE
_LAST	00000000 ( 0.)	05 ( 5.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC PAGE
TSTCNT	00000004 ( 4.)	06 ( 6.)	NOPIC USR OVR REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG
DISPATCH	00000000 ( 0.)	07 ( 7.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG
DISPATCH_X	00000018 ( 24.)	08 ( 8.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG
DATA	00000A5F ( 2655.)	09 ( 9.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC PAGE
INITIALIZE	0000038B ( 907.)	0A ( 10.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
CLEANUP	00000037 ( 55.)	0B ( 11.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
SUMMARY	0000000A ( 10.)	0C ( 12.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

+-----+  
! PERFORMANCE INDICATORS !  
+-----+

PHASE	PAGE FAULTS	CPU TIME	ELAPSED TIME
INITIALIZATION	33	00:00:00.34	00:00:00.80
COMMAND PROCESSING	119	00:00:00.94	00:00:01.52
PASS 1	758	00:00:53.32	00:01:01.64
SYMBOL TABLE SORT	0	00:00:03.53	00:00:03.94
PASS 2	183	00:00:17.63	00:00:19.70
SYSTEM TABLE OUTPUT	49	00:00:01.25	00:00:01.26
PSECT SYNOPSIS OUTPUT	5	00:00:00.17	00:00:00.17
CROSS-REFERENCE OUTPUT	0	00:00:00.00	00:00:00.00
ASSEMBLER RUN TOTALS	1147	00:01:17.19	00:01:29.06

THE WORKING SET LIMIT WAS 2000 PAGES.  
 276398 BYTES (540 PAGES) OF VIRTUAL MEMORY WERE USED TO BUFFER THE INTERMEDIATE CODE.  
 THERE WERE 40 PAGES OF SYMBOL TABLE SPACE ALLOCATED TO HOLD 770 NON-LOCAL AND 30 LOCAL SYMBOLS.  
 719 SOURCE LINES WERE READ IN PASS 1, PRODUCING 46 OBJECT RECORDS IN PASS 2.  
 81 PAGES OF VIRTUAL MEMORY WERE USED TO DEFINE 70 MACROS.

+-----+  
! MACRO LIBRARY STATISTICS !  
+-----+

MACRO LIBRARY NAME	MACROS DEFINED
SYS\SYSROOT:[SYSLIB]DIAG.MLB;1	40
SYS\SYSROOT:[SYSLIB]STARLET.MLB;2	16
TOTALS (ALL LIBRARIES)	56

1090 GETS WERE REQUIRED TO DEFINE 56 MACROS.

THERE WERE NO ERRORS, WARNINGS OR INFORMATION MESSAGES.

13.00078-01 33 01

ZZ\_EVAAA1                    ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ    12-AUG-1986 10:43:12    MDC B7 MACRO V04-00                    PAGE    34  
CM 1700 MACRO RUN STATISTICS                    12-AUG-1986 10:42:50    DUA1:[IK]AAA1.MAR;3                    (1)

MACRO/LIST EVAAA1.MAR

13.00078-01 33 01

ZZ\_EVAAA2  
TABLE OF CONTENTS

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ 7-AUG-1986 17:39:45 МОС ВП MACRO V04-00

PAGE 0

(1)	10	ЛОКАЛЬНЫЕ ПРИСВАИВАНИЯ
(1)	26	ТЕСТ 1: ТЕСТ ПУТЕЙ ДАННЫХ
(1)	123	ТЕСТ 2: ШАБЛОН РИТМИЧЕСКОЙ ПЕЧАТИ ДЛЯ УСТРОЙСТВ LP05/06/07/25/26/27
(1)	209	ТЕСТ 3: ПРОВЕРКА С ВЫБОРОМ ШАБЛОНОВ

ZZ\_EVAAA2  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ

7-AUG-1986 17:39:45

МОС ВП MACRO V04-00

PAGE

1

7-AUG-1986 17:33:40

DUA1: [IKJAAAT.MAR;1

(1)

```

0000      2 .TITLE  ZZ_EVAAA2 ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ
0000      3 .IDENT  /5.3/                ; HOMEР ВЕРСИИ
0000      4 .LIST  MEB
0000      5 .NLIST CND
0000      6 .LIBRARY      /SYS$LIBRARY:DIAG/
00000000  7 .PSECT HEADER, PAGE, NOWRT
0000      8 $DS_BGNMOD SEP_FUNCTIONAL
0000      ; MACROASSEMBLER LAB333
0000      9
0000     10 .SBTTL  ЛОКАЛЬНЫЕ ПРИСВАИВАНИЯ
0000     11 $DS_PARDEF
0000     12 $DS_BITDEF
0000     13 $DS_DSADEF
0000     14 $DS_HPODEF
0000     15 ; ++++++
0000     16 ;
0000     17 ; $DS_SECDEF МАКРОКОМАНДА, ОПРЕДЕЛЯЮЩАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ ТЕСТИРОВАНИЯ
0000     18 ;
0000     19 ; ++++++
0000     20
0000     21 $DS_SECDEF      QUALIFICATION, FUNCTIONAL, MANUAL
0000     22 $DS_DSDEF
0000     23 $DS_PAGE
0000     24

```

ZZ\_EVAAA2  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ  
ТЕСТ 1: ТЕСТ ПУТЕМ ДАННЫХ

7-AUG-1986 17:39:45 МОС ВП MACRO V04-00  
7-AUG-1986 17:33:40 DUA1:[IK]AAT.MAR;1

PAGE 2  
(1)

```

0000      .SBTTL  ТЕСТ 1: ТЕСТ ПУТЕМ ДАННЫХ
00000000  .PSECT  TEST_001,PAGE,NOWRT
0014      27
0014      28 ;+++++
0014      29 ;
0014      30 ; ОПИСАНИЕ ТЕСТА:
0014      31 ;
0014      32 ;
0014      33 ; ВЫВОД СЛЕДУЮЩЕГО ШАБЛОНА ПОВТОРЯЕТСЯ 3 РАЗА
0014      34 ;
0014      35 ;
0014      36 ;      *U*U*U.....
0014      37 ;      U*U*U*U.....
0014      38 ;      @?@?@?.....
0014      39 ;      ?@?@?@.....
0014      40 ;
0014      41 ; ТЕСТ ВЫПОЛНЯЕТСЯ В СЕКЦИЯХ QUALIFICATION, FUNCTIONAL, DEFAULT
0014      42 ;
0014      43 ; ТЕСТ ВЫПОЛНЯЕТСЯ В РЕЖИМЕ БЫСТРОЙ ПРОВЕРКИ
0014      44 ;
0014      45 ;
0014      46 ;
0014      47 ;
0014      48 ;
0014      49 ; ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ ТЕСТА:
0014      50 ;
0014      51 ; 1. ЕСЛИ LP_TYPE=GROUP ПРОДОЛЖАТЬ, ИНАЧЕ - ВЫХОД
0014      52 ; 2. ЗАСЫЛКА В БУФЕР 101010101010..... (B) И ВЫВОД
0014      53 ; 3. ЗАСЫЛКА В БУФЕР 010101010101..... (B) И ВЫВОД
0014      54 ; 4. ЗАСЫЛКА В БУФЕР 0000000011111111000... (B) И ВЫВОД
0014      55 ; 5. ЗАСЫЛКА В БУФЕР 1111111100000000111... (B) И ВЫВОД
0014      56 ; 9. ПОВТОРЕНИЕ ШАГОВ СО 2 ПО 5 ЕЩЕ ДВА РАЗА
0014      57 ;
0014      58 ; ОШИБКА:
0014      59 ;
0014      60 ; 1. СИСТЕМНАЯ НЕИСПРАВНОСТЬ QIO
0014      61 ;
0014      62 ;
0014      63 ;
0014      64 ;+++++
0014      65 %DS_BGNTST <QUALIFICATION,FUNCTIONAL,DEFAULT> ; МАКРОКОМАНДА НАЧАЛА ТЕСТА
0014      DATA_001:
00000000 0014      .LONG  0
0018      TEST_001::
0000      0018      .WORD  ^M<>
001A      66
00000000^EF 00000000^8F D3 001A 67      BITL  #GROUP,LP_TYPE
0014      68      BNEQ  50
0014      69      CLRL  R0
0014      70      %DS_EXIT TEST ; ВЫХОД ИЗ ТЕСТА
0014      BRW  TEST_001_X
00000000^EF 00000000^EF 012B 31 0029 71 50:  MOVZWL COLUMN_WIDTH, BUFFER_SIZE ; ДЛЯ QIO
00000000^EF 00000000^EF 02 C0 0037 72      ADDL2  #2,BUFFER_SIZE ; СКОРРЕКТИРОВАТЬ РАЗМЕР БУФЕРА
0014      73 100:  MOVL   #3,R4 ; СЧЕТЧИК ШАБЛОНА
0014      74 150:  MOVZWL COLUMN_WIDTH,R2 ; УСТАНОВИТЬ СТРОКУ ВЫВОДА
0014      75      ASHL  #-1,R2,R2 ; ИЗ *U*U...
0014      76      MOVAL  OUT_BUFFER,R3 ; ПЕРЕСЛАТЬ СТАРТОВЫЙ АДРЕС ИЗ

```



```

83 00000000/EF 90 0054 77 20x:  MOVB  C_STAR, (R3)+      ; БУФЕРА В РЕГИСТР R3
83 00000000/EF 90 005B 78      MOVB  C_U, (R3)+          ; ЧЕРЕДОВАНИЕ
      EF 52  F5 0062 79      SOBGR  R2, 20x           ; 1 И 0
83 00000000/EF 90 0065 80      MOVB  C_CR, (R3)+        ; ПОМЕСТИТЬ <ВОЗВРАТ КАРЕТКИ> В БУФЕР
83 00000000/EF 90 006C 81      MOVB  C_LINEFEED, (R3)+  ; ПОМЕСТИТЬ <ПЕРЕВОД СТРОКИ> В БУФЕР
      00000000/EF 16 0073 82      JSB   OUTPUT_PASSALL    ; ВЫВЕСТИ ИЗ БУФЕРА
52 00000000/EF 3C 0079 83 25x:  MOVZWL COLUMN_WIDTH, R2   ; СТРОКУ ИЗ УЖУЖ...
      52 52  FF 8F 78 0080 84      ASHL  #-1, R2, R2
53 00000000/EF DE 0085 85      MOVAL OUT_BUFFER, R3
83 00000000/EF 90 008C 86 30x:  MOVB  C_U, (R3)+
83 00000000/EF 90 0093 87      MOVB  C_STAR, (R3)+
      EF 52  F5 009A 88      SOBGR  R2, 30x           ; СЧЕТЧИК ПАРЫ СИМВОЛОВ
83 00000000/EF 90 009D 89      MOVB  C_CR, (R3)+        ; ПОМЕСТИТЬ <ВОЗВРАТ КАРЕТКИ> В БУФЕР
83 00000000/EF 90 00A4 90      MOVB  C_LINEFEED, (R3)+  ; ПОМЕСТИТЬ <ПЕРЕВОД СТРОКИ> В БУФЕР
      00000000/EF 16 00AB 91      JSB   OUTPUT_PASSALL
52 00000000/EF 3C 00B1 92 35x:  MOVZWL COLUMN_WIDTH, R2
      52 52  FF 8F 78 00B8 93      ASHL  #-1, R2, R2
53 00000000/EF DE 00BD 94      MOVAL OUT_BUFFER, R3
83 00000000/EF 90 00C4 95 40x:  MOVB  C_AT_SIGN, (R3)+
83 00000000/EF 90 00CB 96      MOVB  C_QUES_MARK, (R3)+
      EF 52  F5 00D2 97      SOBGR  R2, 40x
83 00000000/EF 90 00D5 98      MOVB  C_CR, (R3)+        ; ПОМЕСТИТЬ <ВОЗВРАТ КАРЕТКИ> В БУФЕР
83 00000000/EF 90 00DC 99      MOVB  C_LINEFEED, (R3)+  ; ПОМЕСТИТЬ <ПЕРЕВОД СТРОКИ> В БУФЕР
      00000000/EF 16 00E3 100     JSB   OUTPUT_PASSALL
52 00000000/EF 3C 00E9 101 45x:  MOVZWL COLUMN_WIDTH, R2
      52 52  FF 8F 78 00F0 102     ASHL  #-1, R2, R2
53 00000000/EF DE 00F5 103     MOVAL OUT_BUFFER, R3
83 00000000/EF 90 00FC 104 50x:  MOVB  C_QUES_MARK, (R3)+
83 00000000/EF 90 0103 105     MOVB  C_AT_SIGN, (R3)+
      EF 52  F5 010A 106     SOBGR  R2, 50x
83 00000000/EF 90 010D 107     MOVB  C_CR, (R3)+        ; ПОМЕСТИТЬ <ВОЗВРАТ КАРЕТКИ> В БУФЕР
83 00000000/EF 90 0114 108     MOVB  C_LINEFEED, (R3)+  ; ПОМЕСТИТЬ <ПЕРЕВОД СТРОКИ> В БУФЕР
      00000000/EF 16 011B 109     JSB   OUTPUT_PASSALL
      02 54  F5 0121 110     SOBGR  R4, 70x           ; ПОВТОРИТЬ ШАБЛОНЫ
      03 11  0124 111     BRB   100x             ; ВЫХОД ИЗ ТЕСТА
      FF1B 31 0126 112 70x:  BRW   15x
00000000/EF 00000000/8F D3 0129 113 100x:  BITL  #LN_01, LF_TYPE   ; УСТРОЙСТВО ТИПА LN01?
      1B 13 0134 114     BEQL  110x             ; НЕТ, ТОГДА НОРМАЛЬНЫЙ ВЫХОД
00000000/EF 01 3C 0136 115     MOVZWL #1, BUFFER_SIZE  ; ДА, ТОГДА ВЫВОД С <ПЕРЕВОДА ФОРМАТА>
53 00000000/EF DE 013D 116     MOVAL OUT_BUFFER, R3
83 00000000/EF 90 0144 117     MOVB  C_FORMFEED, (R3)+
      00000000/EF 16 014B 118     JSB   OUTPUT
      0151 119 110x:
      0151 120 8DS_ENDTEST ; МАКРОКОМАНДА ОКОНЧАНИЯ ТЕСТА
      50 01  D0 0151     MOVL  #1, R0
      0154     TEST_001_X:
00000000/9F 6E  FA 0154     CALLG (SP), @#DS8BREAK
      04 015B     RET
      015C 121 8DS_PAGE

```

```

015C      \.SBTTL  ТЕСТ 2: ШАБЛОН РИТМИЧЕСКОЙ ПЕЧАТИ ДЛЯ LP05/06/07/25/26/27
00000000 .PSECT  TEST_002,PAGE,NOVRT
0028      124
0028      125 ; ++++++
0028      126 ;
0028      127 ; ОПИСАНИЕ ТЕСТА:
0028      128 ;
0028      129 ; ШАБЛОН РИТМИЧЕСКОЙ ПЕЧАТИ СОСТОИТ ИЗ 8 ГРУПП
0028      130 ; ПО 17 СИМВОЛОВ В КАЖДОМ ПЕЧАТИ.
0028      131 ; КАЖДАЯ ГРУППА В ОДНОМ РЯДУ ОДИНАКОВА.
0028      132 ; КАЖДЫЙ СИМВОЛ В ГРУППЕ В КОДЕ КОИ-8 НА ЕДИНИЦУ
0028      133 ; БОЛЬШЕ, ЧЕМ ПРЕДШЕСТВУЮЩИЙ.
0028      134 ; КАЖДЫЙ ПОСЛЕДОВАТЕЛЬНЫЙ РЯД ПЕЧАТИ ИМЕЕТ СТАРТОВЫЙ
0028      135 ; СИМВОЛ ГРУППЫ, ЗНАЧЕНИЕ КОТОРОГО В КОДЕ КОИ-8 НА ЕДИНИЦУ БОЛЬШЕ,
0028      136 ; ЧЕМ СТАРТОВЫЙ СИМВОЛ ГРУППЫ РЯДА, ПРЕДШЕСТВУЮЩЕГО ЕМУ
0028      137 ;
0028      138 ;
0028      139 ;
0028      140 ;
0028      141 ; ПОРЯДОК ВЫПОЛНЕНИЯ ТЕСТА:
0028      142 ; 1. ЕСЛИ УСТАНОВЛЕН ФЛАГ QUICK - ВЫХОД, ИНАЧЕ ПРОДОЛЖАТЬ
0028      143 ;
0028      144 ; 2. ЕСЛИ LP_TYPE=LP11 - ПРОДОЛЖИТЬ, ИНАЧЕ ВЫХОД
0028      145 ;
0028      146 ; 3. УСТАНОВИТЬ BUFFER_SIZE ДЛЯ QIO
0028      147 ;
0028      148 ; 4. ПОМЕСТИТЬ СТАРТОВЫЙ СИМВОЛ В R7
0028      149 ;
0028      150 ; 5. ПОМЕСТИТЬ СИМВОЛ (R7) В БУФЕР
0028      151 ;
0028      152 ; 6. ПОМЕСТИТЬ СЛЕДУЮЩИЙ СИМВОЛ В R7
0028      153 ;
0028      154 ; 7. ЕСЛИ ГРУППА ГОТОВА - ПРОДОЛЖАТЬ, ИНАЧЕ - 5
0028      155 ;
0028      156 ; 8. ЕСЛИ ЛИНИЯ ГОТОВА - ПРОДОЛЖАТЬ, ИНАЧЕ - 4
0028      157 ;
0028      158 ; 9. ВЫВОД ЛИНИИ
0028      159 ;
0028      160 ; 10. ЕСЛИ ГОТОВЫ 96 ЛИНИИ - ВЫХОД, ИНАЧЕ - ПРОДОЛЖАТЬ
0028      161 ;
0028      162 ; 11. ПОЛУЧИТЬ СЛЕДУЮЩИЙ СТАРТОВЫЙ СИМВОЛ
0028      163 ;
0028      164 ; 12. ПЕРЕЙТИ К 5
0028      165 ;
0028      166 ; ОШИБКИ:
0028      167 ; 1. СИСТЕМНАЯ НЕИСПРАВНОСТЬ QIO
0028      168 ;
0028      169 ; 2. ПОКАЗАН НЕВЕРНЫЙ ШАБЛОН
0028      170 ;
0028      171 ;
0028      172 ;
0028      173 ;
0028      174 ;
0028      175 ;
0028      176 ; ++++++
0028      177
0028      178 QDS_BGNTTEST <FUNCTIONAL,DEFAULT> ; МАКРОКОМАНДА НАЧАЛА ТЕСТА

```

ZZ\_EVAAA2  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ 7-AUG-1986 17:39:45 МОС ВП MACRO V04-00  
ТЕСТ 2: ШАБЛОН РИТМИЧЕСКОЙ ПЕЧАТИ LP05/06/07/25/26/27 7-AUG-1986 17:33:40 DUA1: [IKJAAAT.MAR;1

PAGE 5  
(1)

```

0028      DATA_002:
00000000 0028      .LONG    0
002C      TEST_002::
0000      002C      .WORD    ^M<>
002E      179      @DS_BNQUICK    2@          ; ПЕРЕЙТИ ПО МЕТКЕ, ЕСЛИ НЕ УСТАНОВЛЕН ФЛАГ
QUICK
007A      31      002E      BBC #DSA@V_QUICK,-
05 0000FE00 9F      0030      @#DSA@GL_FLAGS,2@
0036      180      CLRL    R0
003B      181      @DS_EXIT TEST          ; ВЫХОД ИЗ ТЕСТА
007A      31      003B      BRW  TEST_002_X
00000000^EF 00000000^BF D3 003B 182 2@:  BITL    #LP_11,LP_TYPE          ;
0005      12      0046      183      BNEQ    5@          ;
0005      D4      0048      184 99@:  CLRL    R0
004A      185      @DS_EXIT TEST
0068      31      004A      BRW  TEST_002_X
00000000^EF 00000000^BF D3 004D 186 5@:  BITL    #LN_01,LP_TYPE          ; УСТРОЙСТВО ТИПА LN01?
000E      12      0058      187      BNEQ    99@          ; ДА, ТОГДА ВЫХОД ИЗ ТЕСТА
00000000^EF 00000000^EF D0 005A 188      MOVL   COLUMN_WIDTH,BUFFER_SIZE ; СЧЕТЧИК БАЙТОВ ДЛЯ QIO
0052      00000060 8F D0 0065 189      MOVL   #96,R2          ; СЧЕТЧИК СТРОК
00000000^EF 0020      D0 006C 190      MOVL   #^X20,TEMP_STORAGE ; СТАРТОВЫЙ СИМВОЛ
0058      00000000^EF DE 0073 191 10@:  MOVAL  OUT_BUFFER,R8   ; АДРЕС БУФЕРА
00000000^EF 0006      D6 007A 192      INCL  TEMP_STORAGE   ; СЛЕДУЮЩИЙ СТАРТОВЫЙ СИМВОЛ
0055      08      D0 0080 193      MOVL   #8,R5          ; СЧЕТЧИК ГРУППЫ
0057      00000000^EF D0 0083 194 15@:  MOVL   TEMP_STORAGE,R7 ; СТАРТОВЫЙ СИМВОЛ
0056      11      D0 008A 195      MOVL   #17,R6         ; СЧЕТЧИК СИМВОЛОВ/ГРУПП
0057      D6 008D 196 20@:  INCL  R7          ; СЛЕДУЮЩИЙ СИМВОЛ
000000080 8F 0057 D1 008F 197      CMPL  R7,#^XB0       ; ЛЕГАЛЬНЫЙ СИМВОЛ?
0007      19      0096      198      BLSS  25@          ; ДА
0057      00000060 8F C2 0098 199      SUBL2 #^X60,R7       ; СДЕЛАТЬ ЛЕГАЛЬНЫМ
0088      57      90 009F 200 25@:  MOVB  R7,(R8)+       ; БУФЕР ЗАПОЛНЕН
00E8      56      F5 00A2 201      SOBGTR R6,20@        ; ГРУППА СФОРМИРОВАНА
00DB      55      F5 00A5 202      SOBGTR R5,15@        ; СТРОКА СФОРМИРОВАНА
00000000^EF 0016      F5 00AB 203      JSB   OUTPUT         ; ВЫВЕСТИ СТРОКУ
00C2      52      F5 00AE 204      SOBGTR R2,10@        ; 96 СТРОК СФОРМИРОВАНЫ
0001      00B1 205      NOP
0005      00B2 206 @DS_ENDTEST ; МАКРОКОМАНДА ОКОНЧАНИЯ ТЕСТА
0050      01      D0 00B2      MOVL  #1,R0
00000000^9F 006E      FA 00B5      CALLG (SP),@#DS@BREAK
0004      00BC      RET
0005      00BD 207 @DS_PAGE

```

```

00BD      .SBTTL  ТЕСТ 3: ПРОВЕРКА С ВЫБОРОМ ШАБЛОНОВ
00000000 .PSECT  TEST_003,PAGE,NOWRT
001C 210 ;+++++
001C 211 ;
001C 212 ; ОПИСАНИЕ ТЕСТА:
001C 213 ;
001C 214 ; ОПЕРАТОР ВВОДИТ ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ,
001C 215 ; ПОСЫЛАЯ ИХ НА УСТРОЙСТВО ПЕЧАТИ.
001C 216 ; ПЕРЕВОД СТРОКИ ПОСЫЛАЕТСЯ ПЕРЕД БУФЕРОМ,
001C 217 ; ВОЗВРАТ КАРЕТКИ - ПОСЛЕ.
001C 218 ; ВОЗВРАТ ПОД УПРАВЛЕНИЕМ СУПЕРВИЗОРА ОСУЩЕСТВЛЯЕТСЯ
001C 219 ; С ПОМОЩЬЮ CTRL/C. ВЫБРАВ ШАБЛОН,
001C 220 ; МОЖНО ЗАПУСТИТЬ ТЕСТ СНОВА
001C 221 ;
001C 222 ;
001C 223 ;
001C 224 ;
001C 225 ; ПОРЯДОК ВЫПОЛНЕНИЯ ТЕСТА
001C 226 ;
001C 227 ; 1. ЕСЛИ УСТАНОВЛЕН ФЛАГ QUICK - ПРОДОЛЖАТЬ, ИНАЧЕ - ВЫХОД
001C 228 ; 2. ЕСЛИ LP_TYPE=GROUP - ПРОДОЛЖИТЬ, ИНАЧЕ - ВЫХОД
001C 229 ; 3. ПОЛУЧИТЬ БУФЕР У ОПЕРАТОРА
001C 230 ; 4. ПОМЕСТИТЬ БУФЕР В OUT_BUFFER
001C 231 ; 5. ЕСЛИ LP_TYPE=LN01, ТО ЗАПРОСИТЬ У ОПЕРАТОРА
001C 232 ;     СКОЛЬКО ЛИНИИ ОН ЖЕЛАЕТ ВЫВЕСТИ
001C 233 ;
001C 234 ; 6. ВЫВЕСТИ БУФЕР
001C 235 ;
001C 236 ; 7. ЕСЛИ CTRL/C - ВЫХОД, ИНАЧЕ - 55
001C 237 ;
001C 238 ; ОШИБКИ
001C 239 ;
001C 240 ; 1. СИСТЕМНАЯ НЕИСПРАВНОСТЬ QIO
001C 241 ;
001C 242 ; DEBUG:
001C 243 ;+++++
001C 244 ;DS_BGNTST <MANUAL> ; МАКРОКОМАНДА НАЧАЛА ТЕСТА
001C     DATA_003:
00000000 001C     .LONG  0
0020     TEST_003::
0000     0020     .WORD  ^M<>
0022     245     ;DS_BNQUICK  2; ; ПЕРЕЙТИ ПО МЕТКЕ, ЕСЛИ УСТАНОВЛЕН ФЛАГ QUICK
0022     BBC #DSA;V_QUICK,-
05_0000FE00 9F 0024     @#DSA;GL_FLAGS,2;
002A     246     CLRL  R0
002C     247     ;DS_EXIT TEST ; ВЫЙТИ ИЗ ТЕСТА
00CB     31 002C     BRW  TEST_003_X
00000000^EF 00000000^BF D3 002F     248 2;: BITL  #GROUP,LP_TYPE
005     12 003A     249     BNEQ  5;
005     D4 003C     250     CLRL  R0
003E     251     ;DS_EXIT TEST ; ВЫЙТИ ИЗ ТЕСТА
00B9     31 003E     BRW  TEST_003_X
0041     252 5;: ;DS_ASKSTR_S ECHO_MSG,ECHO_BUFFER,#132 ; ЗАПРОСИТЬ У ОПЕРАТОРА СТРОКУ
00 DD 0041     PUSHL #0
00 DD 0043     PUSHL #0
00 DD 0045     PUSHL #0
000000B4 8F DD 0047     PUSHL #132

```

ZZ\_EVAAA2  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ  
ТЕСТ 3: ПРОВЕРКА С ВЫБОРОМ ШАБЛОНОВ

7-AUG-1986 17:39:45 MDC ВП MACRO V04-00  
7-AUG-1986 17:33:40 DUA1: [IK]AAT.MAR;1

PAGE 7  
(1)

```

00000000/EF DF 004D PUSHAL ECHO_BUFFER
00000000/EF DF 0053 PUSHAL ECHO_MSG
00000000/9F 06 FB 0059 CALLS #6,@#DS%ASKSTR
00000001/EF DF 0060 253 PUSHAL ECHO_BUFFER+1 ;
00000000/EF DF 0066 254 PUSHAL OUT_BUFFER ;
7E 00000000/EF 9A 006C 255 MOVZBL ECHO_BUFFER,-(SP) ; СЧЕТЧИК БАЙТОВ
00000000/EF 03 FB 0073 256 CALLS #3,MOV_BUF ;
00000000/EF 00000000/EF 9A 007A 257 MOVZBL ECHO_BUFFER,BUFFER_SIZE ; СЧЕТЧИК БАЙТОВ ДЛЯ QIO
00000000/EF 00000000/8F D3 0085 258 BITL #LN_01,LP_TYPE ;
56 13 0090 259 BEQL 10% ;
0092 260 %DS_ASKDATA_S LN01_MSG,TEMP_STORAGE,,#1,#10000 ; ЗАП.У ОПЕР.ЧИСЛ.ЗНАЧЕНИЕ
00 DD 0092 PUSHL #0
00 DD 0094 PUSHL #0
00 DD 0096 PUSHL #0
00002710 8F DD 0098 PUSHL #10000
01 DD 009E PUSHL #1
FFFFFFF 8F DD 00A0 PUSHL #-1
0A DD 00A6 PUSHL #PAR%_DEC
00000000/EF DF 00A8 PUSHAL TEMP_STORAGE
00000000/EF DF 00AE PUSHAL LN01_MSG
00000000/9F 09 FB 00B4 CALLS #9,@#DS%ASKDATA
52 00000000/EF D0 00BB 261 MOVL TEMP_STORAGE,R2
00000000/EF 16 00C2 262 8%: JSB OUTPUT
F7 52 F5 00C8 263 SOBGR R2,8%
53 00000000/EF DE 00CB 264 MOVAL OUT_BUFFER,R3
83 00000000/EF 90 00D2 265 MOVBL C_FORMFEED,(R3)+
00000000/EF 01 3C 00D9 266 MOVZWL #1,BUFFER_SIZE
00000000/EF 16 00E0 267 JSB OUTPUT
0F 11 00E6 268 BRB 20%
00000000/EF 16 00E8 269 10%: JSB OUTPUT ;
00EE 270 %DS_BREAK
00000000/9F 6E FA 00EE CALLG (SP),@#DS%BREAK
F1 11 00F5 271 BRB 10%
00F7 272 20%:
00F7 273 %DS_ENDTEST ; МАКРОКОМАНДА ОКОНЧАНИЯ ТЕСТА
50 01 D0 00F7 MOVL #1,R0
00FA TEST_003_X::
00000000/9F 6E FA 00FA CALLG (SP),@#DS%BREAK
04 0101 RET
0102 274 %DS_PAGE

```

13.00078-01 33 01

ZZ\_EVAAA2  
5.3

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ  
ТЕСТ 3: ПРОВЕРКА С ВЫБОРОМ ШАБЛОНОВ

7-AUG-1986 17:39:45  
7-AUG-1986 17:33:40

MOC BP MACRO V04-00  
DUA1:CIKJAAAT.MAR;1

PAGE 8  
(1)

```
00000000 .PSECT RTSTCNT,NOEXE,NOWRT,OVR, LONG
00000003 0000 .LONG RTN
0004 277 .END
```

ZZ\_EVAAA2  
SYMBOL TABLE

ФАНЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ

7-AUG-1986 17:39:45

MOC ВП MACRO V04-00

PAGE 9

7-AUG-1986 17:33:40

DUA1: [IKJAAAT.MAR;1

(1)

DSM	= 00000008			DATA_003	0000001C	R	06
DSN	= 00000004			DEFAULT	= 00000000	B	
DSB	= FFFFFFFF			DSASKDATA	*****	X	06
DSENV	= 00000002	G		DSASKSTR	*****	X	06
DSER	= FFFFFFFF			DSBREAK	*****	X	03
DSMO	= 00000001	G		DSK_ERROR	= 00000002		
DSST	= 00000000			DSK_NORMAL	= 00000001		
DSN	= 00000003			DSK_SEVERE	= 00000004		
BASE_001	00000000	R	03	DSK_SUBSYS	= 00000066		
BASE_002	00000000	R	05	DSK_WARNING	= 00000000		
BASE_003	00000000	R	06	DS_ARITM	= 006600D0		
BIT...	= 00660110			DS_BADLINK	= 006600F0		
BIT0	= 00000001			DS_BADTYPE	= 006600E8		
BIT1	= 00000002			DS_CHME	= 006600A8		
BIT10	= 00000400			DS_CHMK	= 006600E0		
BIT11	= 00000800			DS_DEVNAME	= 00660108		
BIT12	= 00001000			DS_ERROR	= 00660002		
BIT13	= 00002000			DS_FHWE	= 00660068		
BIT14	= 00004000			DS_FRAGBUF	= 00660080		
BIT15	= 00008000			DS_ICBUSY	= 006600C8		
BIT16	= 00010000			DS_ICERR	= 006600C0		
BIT17	= 00020000			DS_IHWE	= 00660060		
BIT18	= 00040000			DS_ILLCHAR	= 00660018		
BIT19	= 00080000			DS_ILLPAGECNT	= 00660078		
BIT2	= 00000004			DS_ILLUNIT	= 00660100		
BIT20	= 00100000			DS_INSMEM	= 00660050		
BIT21	= 00200000			DS_IPL2HI	= 006600B8		
BIT22	= 00400000			DS_IVADDR	= 00660040		
BIT23	= 00800000			DS_IVVECT	= 00660038		
BIT24	= 01000000			DS_KRNLSTK	= 00660090		
BIT25	= 02000000			DS_LOGIC	= 00660070		
BIT26	= 04000000			DS_MCHK	= 006600B8		
BIT27	= 08000000			DS_MM0FF	= 00660058		
BIT28	= 10000000			DS_NEEDUNIT	= 006600F8		
BIT29	= 20000000			DS_NORMAL	= 00660001		
BIT3	= 00000008			DS_NOSUPPORT	= 006600B1		
BIT30	= 40000000			DS_NOTDON	= 00660030		
BIT31	= 80000000			DS_NOTIMP	= 006600B0		
BIT4	= 00000010			DS_NULLSTR	= 00660010		
BIT5	= 00000020			DS_OVERFLOW	= 00660008		
BIT6	= 00000040			DS_POWER	= 00660098		
BIT7	= 00000080			DS_PROGERR	= 00660020		
BIT8	= 00000100			DS_SEVERE	= 00660004		
BIT9	= 00000200			DS_TRANSL	= 006600A0		
BUFFER_SIZE	*****	X	03	DS_TRUNCATE	= 0066002B		
CEP_FUNCTIONAL	= 00000000			DS_VASFULL	= 00660048		
CEP_REPAIR	= 00000001			DS_VNEXPINT	= 006600DB		
COLUMN_WIDTH	*****	X	03	DS_WARNING	= 00660000		
C_AT_SIGN	*****	X	03	DSAL_APTMAIL	0000FE00		
C_CR	*****	X	03	DSAT_APTTXT	0000FA00		
C_FORMFEED	*****	X	03	DSAGL_APTCOM	0000FE04		
C_LINEFEED	*****	X	03	DSAGL_DEVLEN	0000FE58		
C_QUES_MARK	*****	X	03	DSAGL_ERRNO	0000FE44		
C_STAR	*****	X	03	DSAGL_EVENT	0000FE48		
C_U	*****	X	03	DSAGL_FLAGS	0000FE00		
DATA_001	00000014	R	03	DSAGL_MSGTYP	0000FE40		
DATA_002	00000028	R	05	DSAGL_PASSES	0000FE08		

ZZ\_EVAAA2  
SYMBOL TABLE

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ

7-AUG-1986 17:39:45

MOC ВП MACRO V04-00

PAGE 10

7-AUG-1986 17:33:40

DUA1:[IKJAAAT.MAR;1

(1)

DSA0GL_PASSNO	0000FE54			PAR0V_NODEF	= 00000001		
DSA0GL_SECTNO	0000FE10			PAR0_BIN	= 00000002		
DSA0GL_SID	0000FE14			PAR0_DEC	= 0000000A		
DSA0GL_SUBTNO	0000FE4C			PAR0_HEX	= 00000010		
DSA0GL_TESTNO	0000FE50			PAR0_NO	= 00000000		
DSA0GL_UNITS	0000FE0C			PAR0_OCT	= 00000008		
DSA0GQ_MSGPTR	0000FE68			PAR0_YES	= 00000001		
DSA0GT_DEVNAM	0000FE5C			QUALIFICATION	= 00000001	G	
DSA0V_QUICK	= 0000000B			SEP_FUNCTIONAL	= 00000002		
ECHO_BUFFER	*****	X	06	SEP_REPAIR	= 00000003		
ECHO_MSG	*****	X	06	SIZ...	= 00000001		
ENV0M_DOMAIN	= 00000002			TEMP_STORAGE	*****	X	05
ENV0M_LEVEL	= 00000001			TEST_001	00000018	RG	03
ENV0M_SUPER	= 000003FC			TEST_001_X	00000154	RG	03
ENV0S_DOMAIN	= 00000001			TEST_002	0000002C	RG	05
ENV0S_LEVEL	= 00000001			TEST_002_X	000000B5	RG	05
ENV0S_SUPER	= 00000008			TEST_003	00000020	RG	06
ENV0V_DOMAIN	= 00000001			TEST_003_X	000000FA	RG	06
ENV0V_LEVEL	= 00000000						
ENV0V_SUPER	= 00000002						
ENV0_CPU	= 00000000						
ENV0_FUNCTIONAL	= 00000000						
ENV0_REPAIR	= 00000001						
ENV0_SUPER	= 00000001						
ENV0_SYSTEM	= 00000001						
FUNCTIONAL	= 00000002	G					
GROUP	*****	X	03				
HP0A_DEPENDENT	00000032						
HP0A_DEVICE	00000018						
HP0A_DVA	0000001C						
HP0A_LINK	00000020						
HP0B_DRIVE	0000000B						
HP0B_FLAGS	0000000A						
HP0Q_DEVICE	00000000						
HP0T_DEVICE	0000000C						
HP0T_TYPE	00000026						
HP0W_SIZE	0000000B						
HP0W_VECTOR	00000024						
LN01_MSG	*****	X	06				
LN_01	*****	X	03				
LP_11	*****	X	05				
LP_TYPE	*****	X	03				
MANUAL	= 00000003	G					
MOV_BUF	*****	X	06				
MSG_001	00000002	R	03				
MSG_002	00000002	R	05				
MSG_003	00000002	R	06				
OUTPUT	*****	X	03				
OUTPUT_PASSALL	*****	X	03				
OUT_BUFFER	*****	X	03				
PAR0M_ATDEF	= 00000010						
PAR0M_ATHI	= 00000008						
PAR0M_ATLO	= 00000004						
PAR0M_NODEF	= 00000002						
PAR0V_ATDEF	= 00000004						
PAR0V_ATHI	= 00000003						
PAR0V_ATLO	= 00000002						



ZZ\_EVAAA2  
PSECT SYNOPSIS

ФАЙЛ ТЕСТА ПРОВЕРКИ ПЕЧАТАЮЩИХ УСТРОЙСТВ

7-AUG-1986 17:39:45 MOC ВП MACRO V04-00  
7-AUG-1986 17:33:40 DUA1:[IKJAAAT.MAR;1

PAGE 11  
(1)

+-----+  
! PSECT SYNOPSIS !  
+-----+

PSECT NAME	ALLOCATION	PSECT NO.	ATTRIBUTES
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
HEADER	00000000 ( 0.)	01 ( 1.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC PAGE
ABS	0000FE70 (65136.)	02 ( 2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
TEST_001	0000015C ( 348.)	03 ( 3.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC PAGE
DISPATCH	0000004B ( 72.)	04 ( 4.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG
TEST_002	000000BD ( 189.)	05 ( 5.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC PAGE
TEST_003	00000102 ( 258.)	06 ( 6.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC PAGE
RTSTCNT	00000004 ( 4.)	07 ( 7.)	NOPIC USR OVR REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG

+-----+  
! PERFORMANCE INDICATORS !  
+-----+

PHASE	PAGE FAULTS	CPU TIME	ELAPSED TIME
INITIALIZATION	39	00:00:00.24	00:00:00.44
COMMAND PROCESSING	116	00:00:00.91	00:00:01.33
PASS 1	323	00:00:20.57	00:00:25.92
SYMBOL TABLE SORT	0	00:00:00.90	00:00:00.90
PASS 2	62	00:00:05.54	00:00:05.72
SYMBOL TABLE OUTPUT	22	00:00:00.54	00:00:00.54
PSECT SYNOPSIS OUTPUT	3	00:00:00.12	00:00:00.12
CROSS-REFERENCE OUTPUT	0	00:00:00.00	00:00:00.00
ASSEMBLER RUN TOTALS	565	00:00:28.86	00:00:35.00

THE WORKING SET LIMIT WAS 1350 PAGES.  
83919 BYTES (164 PAGES) OF VIRTUAL MEMORY WERE USED TO BUFFER THE INTERMEDIATE CODE.  
THERE WERE 20 PAGES OF SYMBOL TABLE SPACE ALLOCATED TO HOLD 239 NON-LOCAL AND 25 LOCAL SYMBOLS.  
277 SOURCE LINES WERE READ IN PASS 1, PRODUCING 26 OBJECT RECORDS IN PASS 2.  
50 PAGES OF VIRTUAL MEMORY WERE USED TO DEFINE 42 MACROS.

+-----+  
! MACRO LIBRARY STATISTICS !  
+-----+

MACRO LIBRARY NAME	MACROS DEFINED
SYS\SYSROOT:[SYSLIB]DIAG.MLB;1	25
SYS\SYSROOT:[SYSLIB]STARLET.MLB;2	6
TOTALS (ALL LIBRARIES)	

485 GETS WERE REQUIRED TO DEFINE 31 MACROS.

THERE WERE NO ERRORS, WARNINGS OR INFORMATION MESSAGES.

MACRO/LIST EVAAA2.MAR

- 202 -

.Э.00078-01 33 01

ПРИЛОЖЕНИЕ 3

**РИСУНКИ**

СХЕМА ФУНКЦИОНИРОВАНИЯ ДИАГНОСТИЧЕСКОГО СУПЕРВИЗОРА

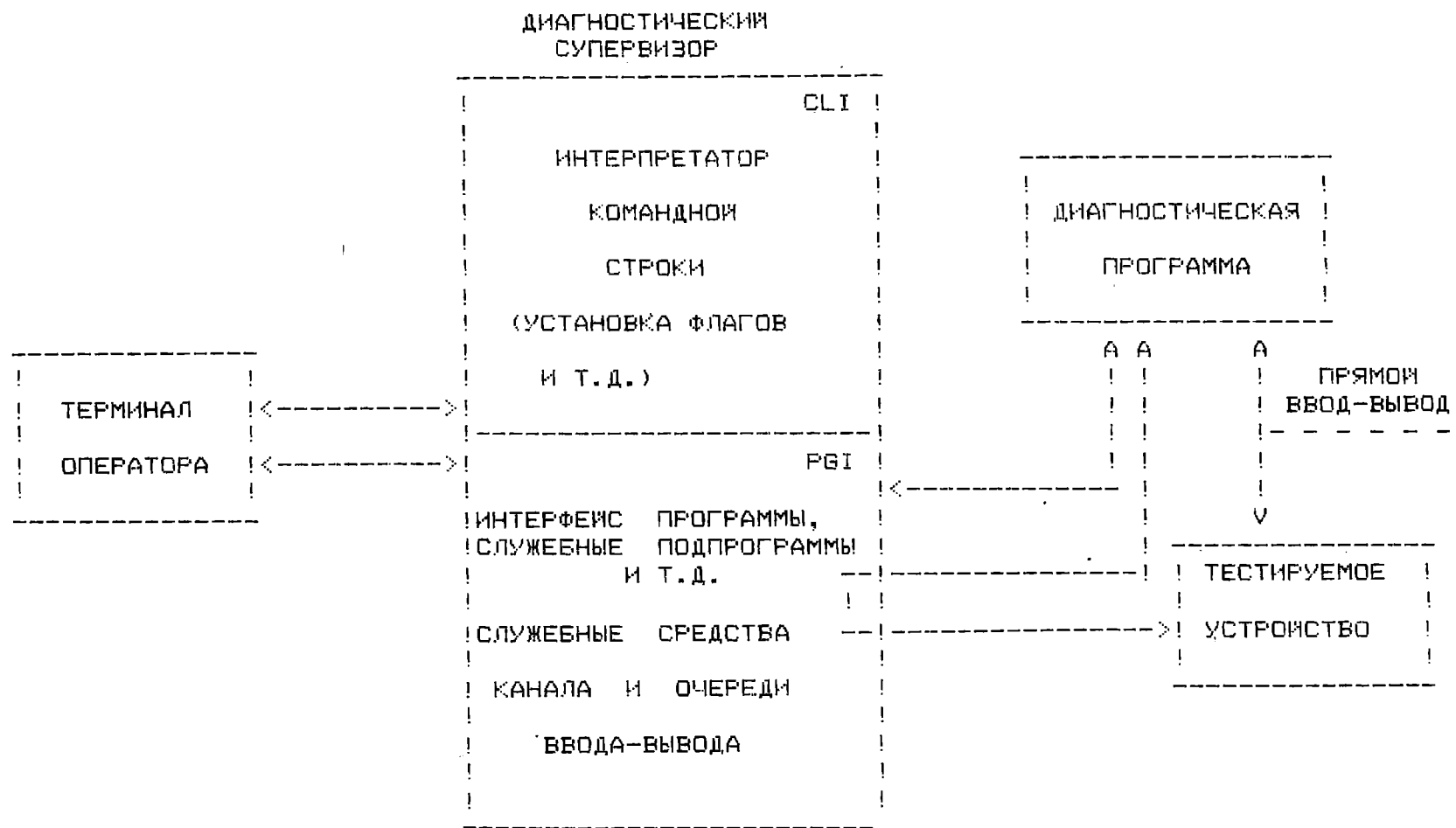


РИС. 1

13.00078-01 33 01

ВЗАИМОДЕЙСТВИЕ ДИАГНОСТИЧЕСКОГО СУПЕРВИЗОРА И  
ДИАГНОСТИЧЕСКОЙ ПРОГРАММЫ

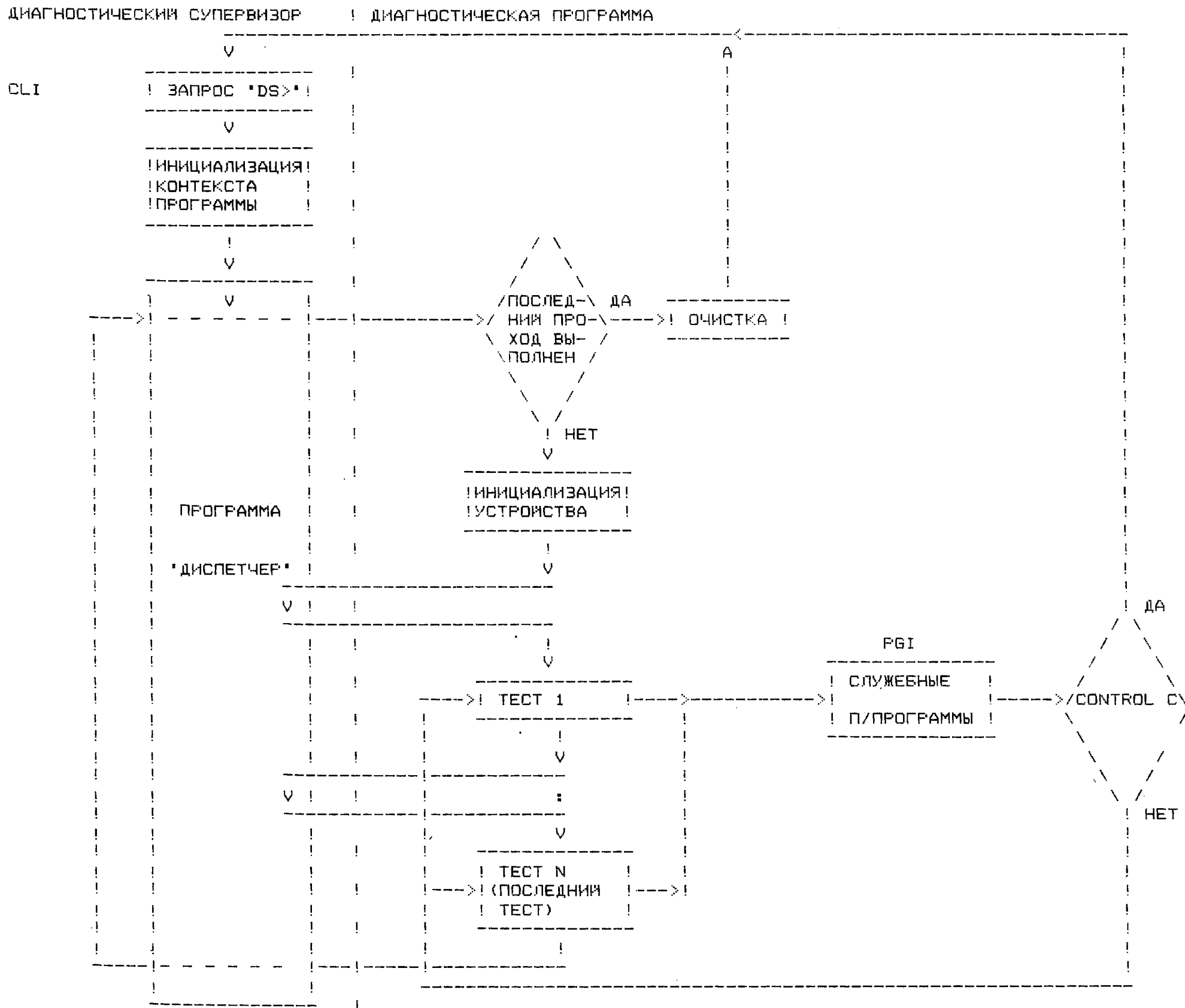


РИС. 2

## РАСПРЕДЕЛЕНИЕ ПАМЯТИ ДИАГНОСТИЧЕСКОЙ СИСТЕМЫ



РИС. 3

СХЕМА ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ  
ДЛЯ ПАРАЛЛЕЛЬНОГО ТЕСТИРОВАНИЯ

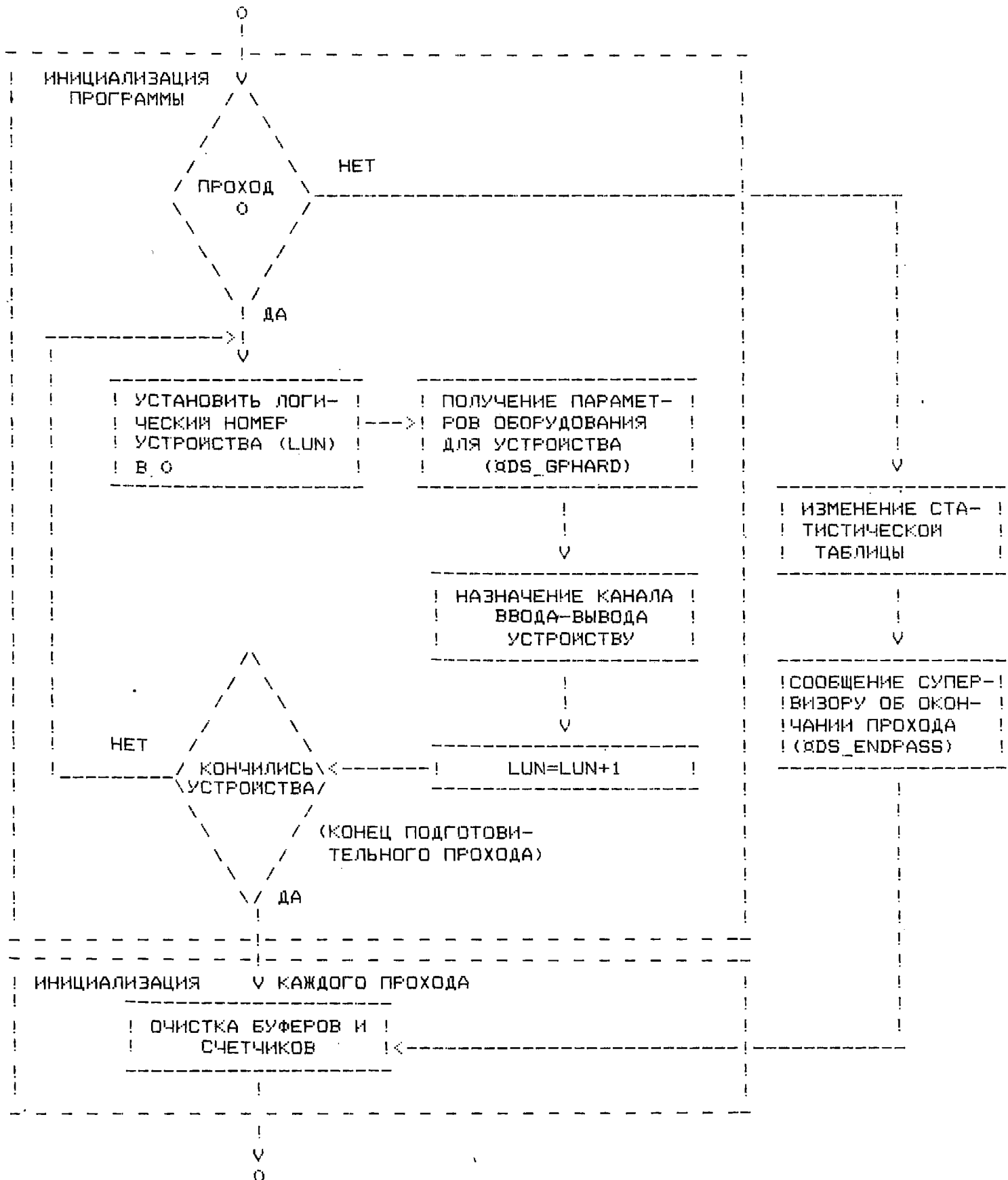


РИС. 4

СХЕМА ПОДПРОГРАММЫ ИНИЦИАЛИЗАЦИИ  
 ДЛЯ ПОСЛЕДОВАТЕЛЬНОГО ТЕСТИРОВАНИЯ УСТРОЙСТВ

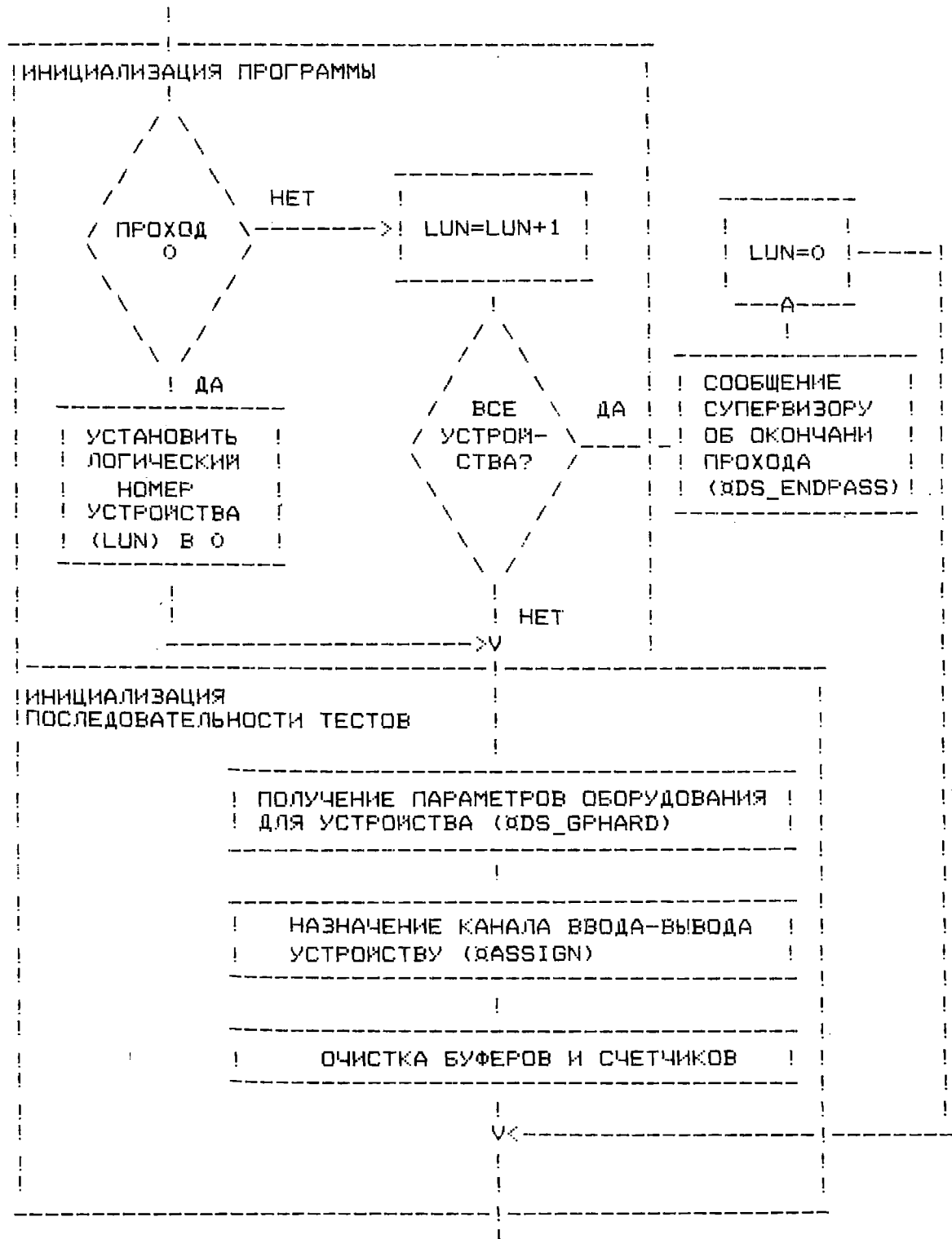


РИС. 5

СХЕМА ТИПИЧНОЙ ПОДПРОГРАММЫ ОЧИСТКИ

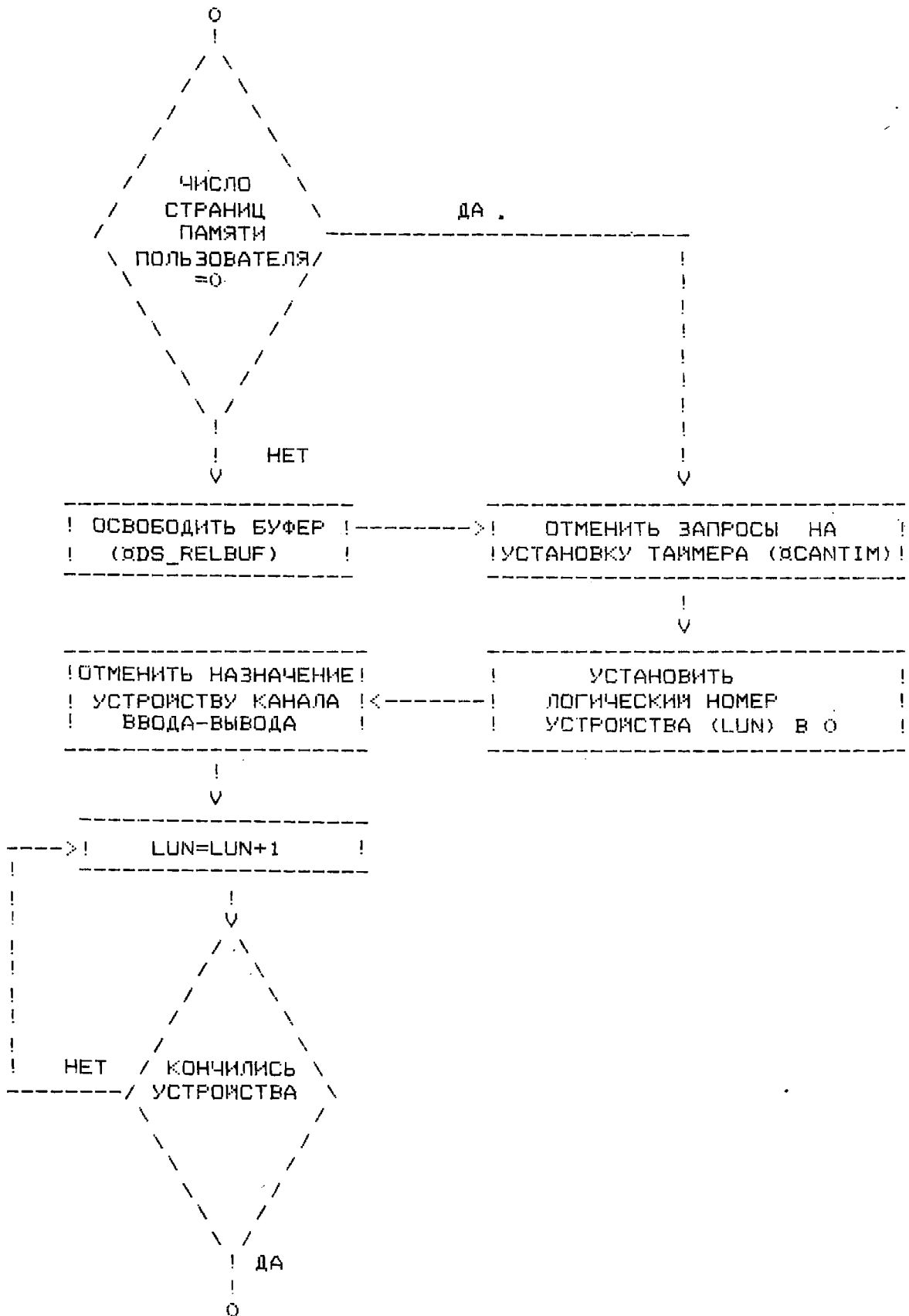


РИС. 6



СХЕМА ТИПИЧНОЙ ПОДПРОГРАММЫ ЗАКЛЮЧИТЕЛЬНОГО ОТЧЕТА

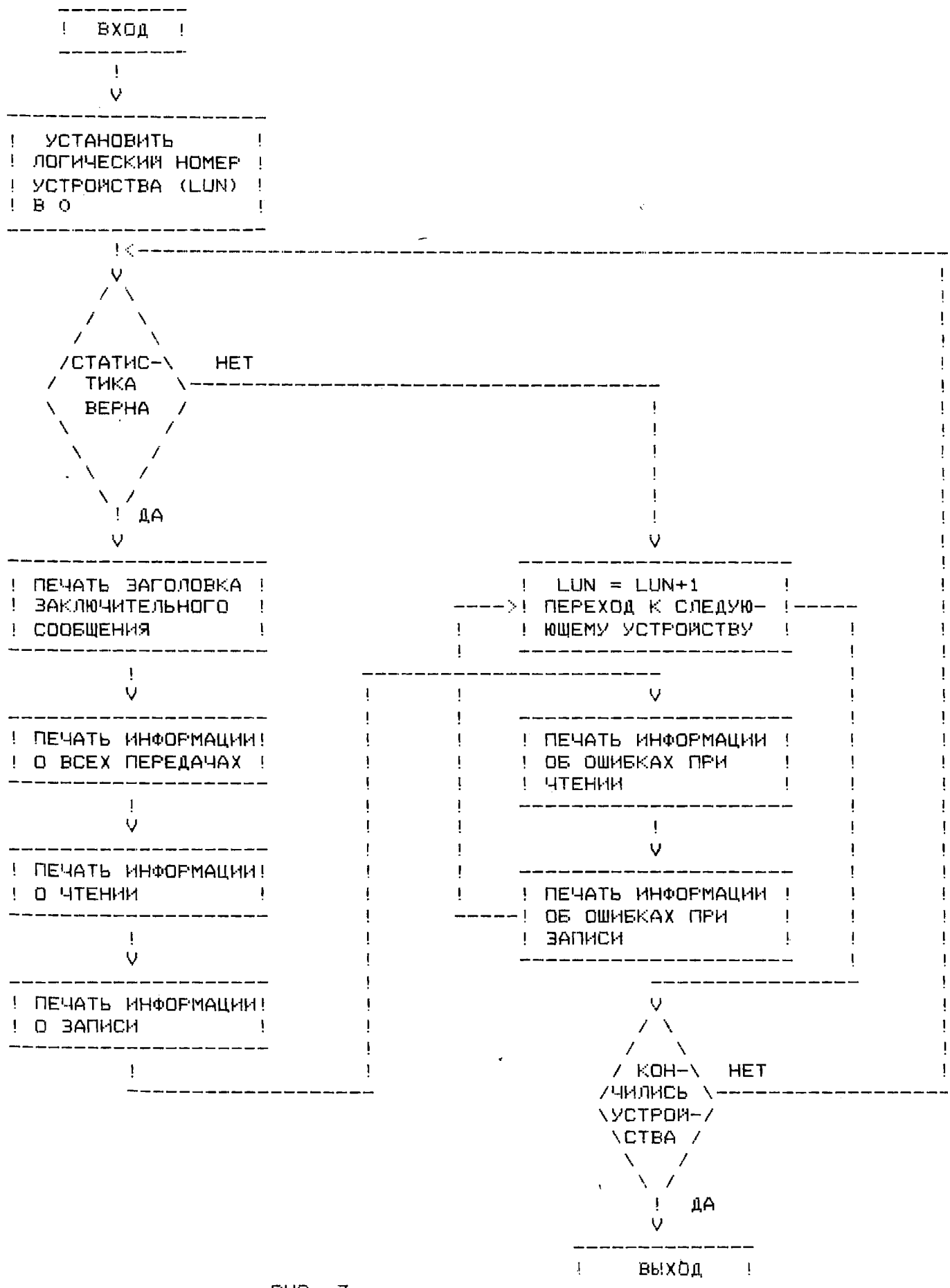


РИС. 7

13.00078-01 33 01

СХЕМА РАСПЕЧАТКИ СООБЩЕНИЯ ОБ ОШИБКЕ

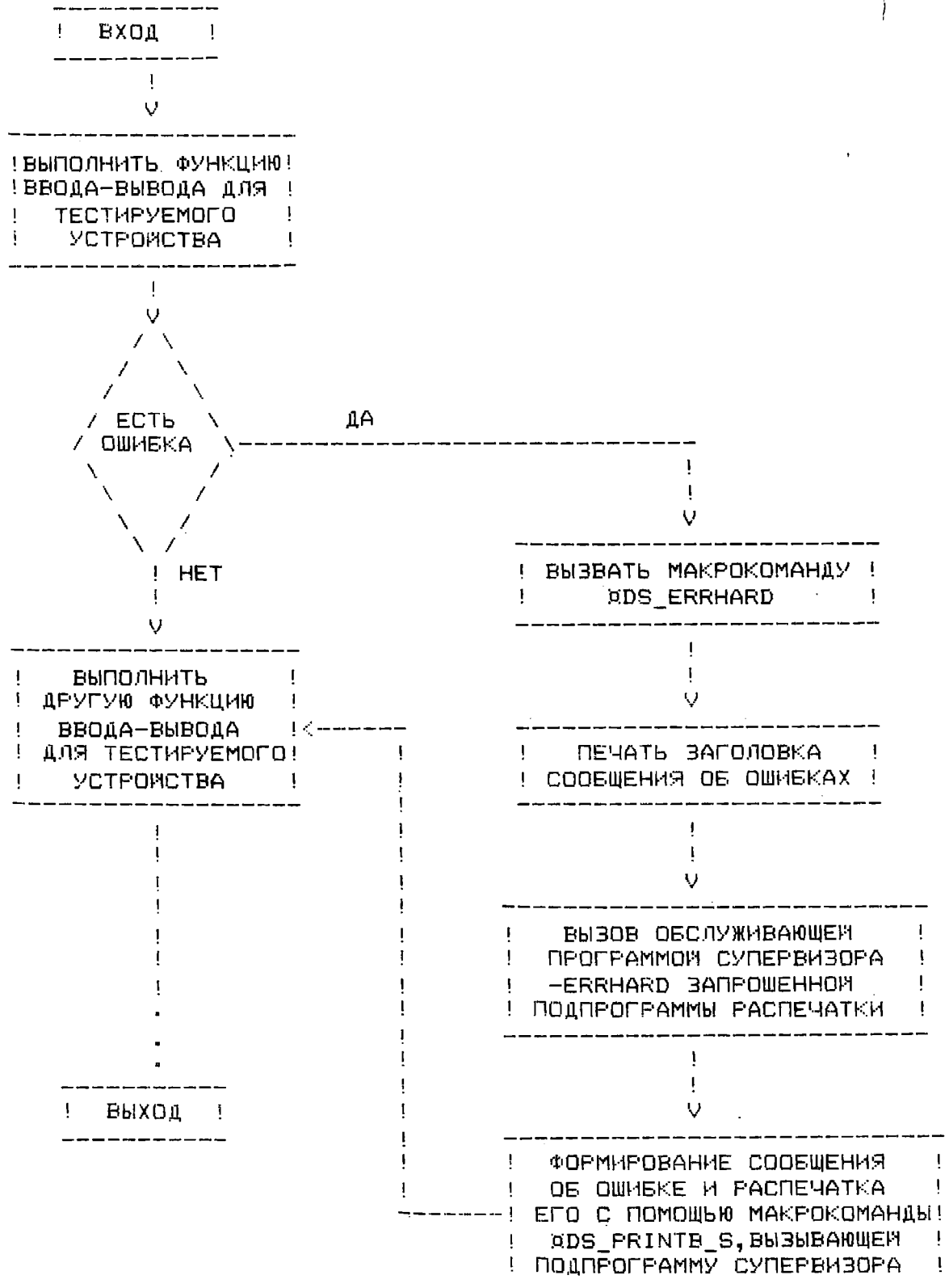


РИС. 8

## СХЕМА ДИАЛОГА МЕЖДУ ПРОГРАММОЙ И ОПЕРАТОРОМ



РИС. 9

## СТРУКТУРА ДЕРЕВА ИНТЕРПРЕТАТОРА КОМАНД

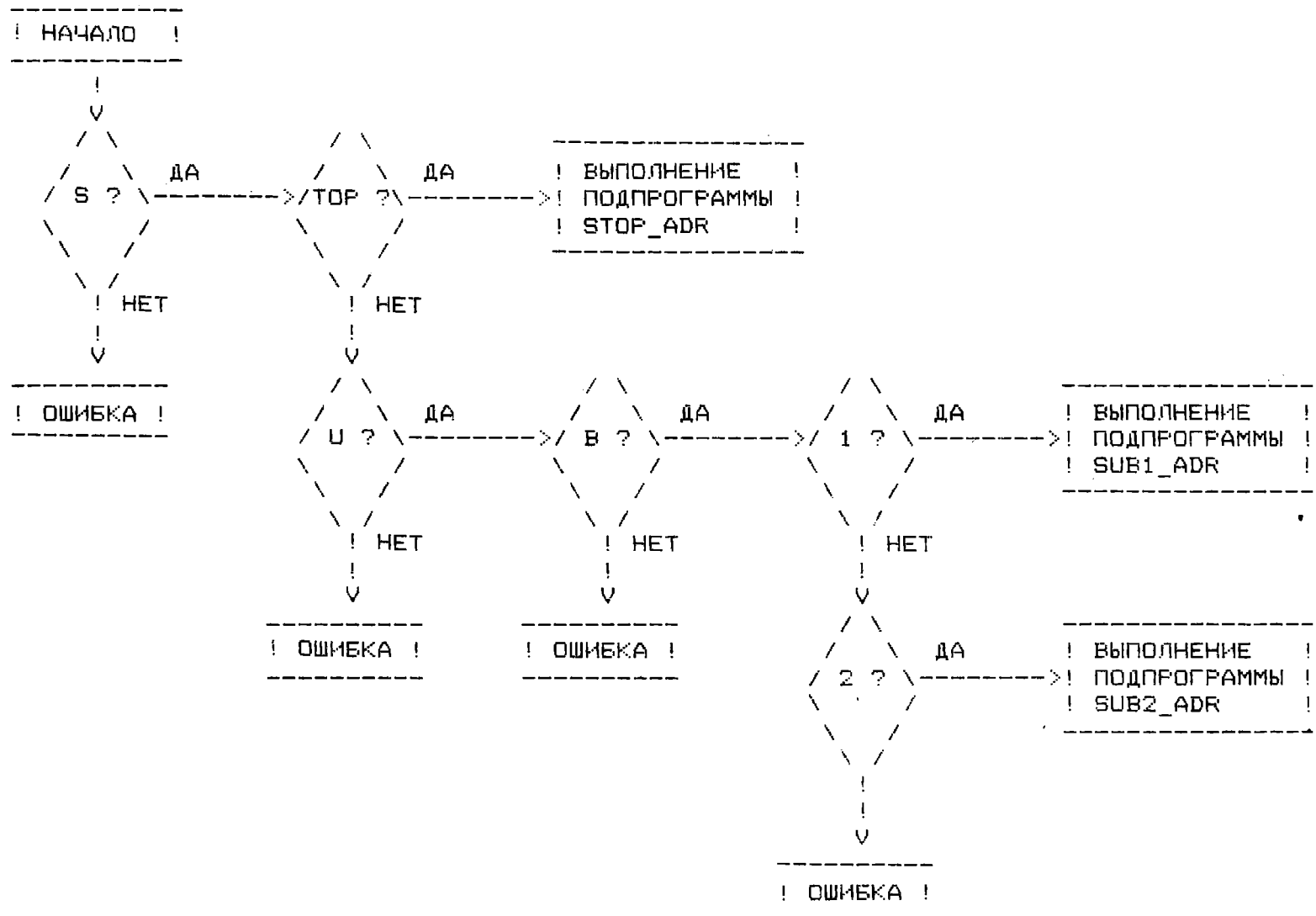


РИС. 10

13.00078-01 33 01

**ПЕРЕЧЕНЬ ССЫЛОЧНЫХ ДОКУМЕНТОВ**

1. ДИАГНОСТИЧЕСКОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВК СМ 1700.  
ОПИСАНИЕ ПРОГРАММЫ. 00056 01 13 01.

