

**УСТРОЙСТВО
ЧИСЛОВОГО ПРОГРАММНОГО УПРАВЛЕНИЯ
NC-110, NC-310
NC-200, NC-201, NC-201M, NC-202, NC-210, NC-220, NC-230**

Программирование интерфейса PLC

**Санкт-Петербург
2008 г**

АННОТАЦИЯ

Документ «Программирование интерфейса PLC» (версия В2.2) распространяется на устройства числового программного управления NC-110, NC-200, NC-201, NC-201M, NC-202, NC-210, NC-220, NC-230 и NC-310 (далее УЧПУ). В данном документе изложены сведения о программном интерфейсе PLC, используемом в УЧПУ, а также сведения о языке PLC, на котором составляют программы логики управляемого оборудования (ПЛ).

Документ предназначен для технологов-программистов, разрабатывающих ПЛ с учётом минимального взаимодействия между УЧПУ и управляемым оборудованием.

СОДЕРЖАНИЕ

1.	СТРУКТУРА ИНТЕРФЕЙСА PLC	7
2.	ЛИНИИ СВЯЗИ МЕЖДУ МОДУЛЯМИ ПРО И PLC.....	9
3.	ПРИНЦИП РАБОТЫ ПЛ.....	10
4.	ОТЛАДКА ПЛ	11
5.	ЯЗЫК PLC.....	12
5.1.	ЭЛЕМЕНТЫ ЯЗЫКА	12
5.1.1.	Операнды	12
5.1.1.1.	Сигналы	12
5.1.1.2.	Слово	13
5.1.1.3.	Константы	14
5.1.1.4.	Логические уровни и значения переменных	14
5.1.2.	Метаоперанды	14
5.1.2.1.	Программируемые таймеры	14
5.1.2.2.	Программные счётчики	17
5.1.2.3.	Одновибраторы (генератор импульсов)	19
5.1.2.4.	Компараторы	20
5.1.2.5.	Компаратор ASCII	21
5.1.3.	Функции	21
5.1.3.1.	Шифратор (кодирование)	21
5.1.3.2.	Дешифратор	22
5.1.3.3.	Преобразование в BCD	22
5.1.3.4.	Преобразование в двоичный код	23
5.1.3.5.	Знак	23
5.1.3.6.	Модуль	23
5.1.3.7.	Мультиплексор	23
5.1.3.8.	Полуслово	24
5.1.3.9.	Условные переходы	24
5.1.4.	Операторы	26
5.1.4.1.	Присвоение	26
5.1.4.2.	Логические операторы	26
5.1.4.3.	Применение скобок для выполнения логических схем	27
5.1.4.4.	Арифметические операторы	27
6.	СТРУКТУРА ЯЗЫКА	29
6.1.	ОПЕРАТОРЫ ОПИСАНИЯ	29
6.1.1.	Сообщения для оператора	29
6.1.2.	Цикл	29
6.1.3.	Комментарии в программе	29
6.1.4.	Символьное представление ПЛ	29
6.2.	ИСПОЛНИТЕЛЬНЫЕ ОПЕРАТОРЫ	30
6.2.1.	Операторы для сигналов	30
6.2.2.	Операторы для слов	30
7.	ПОРЯДОК ПРИМЕНЕНИЯ ЯЗЫКА PLC.....	32
8.	КРАТКОЕ ИЗЛОЖЕНИЕ ЯЗЫКА PLC.....	33
8.1.	ЭЛЕМЕНТАРНЫЕ ОПЕРАЦИИ	33
8.2.	ТАЙМЕРЫ (ПОСТОЯННЫЕ ИЛИ ПАРАМЕТРИЧЕСКИЕ)	33
8.3.	СЧЕТЧИКИ (ПОСТОЯННЫЕ ИЛИ ПАРАМЕТРИЧЕСКИЕ)	33
8.4.	КОМПАРАТОРЫ С 8 БИТАМИ	33
8.5.	ДЕКОДИРОВАНИЕ 8 ВЫХОДОВ	33
8.6.	КОДИРОВАНИЕ В ДВОИЧНОМ КОДЕ 8 СИГНАЛОВ	33
8.7.	ОТДЕЛЬНЫЕ СИГНАЛЫ	34
8.8.	СЛОЖНЫЕ СИГНАЛЫ ИЗ 8 БИТ: СЛОВО	34
8.9.	СЛОЖНЫЕ СИГНАЛЫ ИЗ 4 БИТ: ПОЛУСЛОВО	34

8.10.	УСЛОВНЫЕ ПЕРЕХОД	34
8.11.	ОБОЗНАЧЕНИЕ СИГНАЛОВ	34
8.12.	СООБЩЕНИЯ ДЛЯ ОПЕРАТОРА	34
8.13.	УСЛОВНОЕ ПРИСВОЕНИЕ ИЛИ МУЛЬТИПЛЕКСОР	34
8.14.	ОДНОВИБРАТОРЫ	34
8.15.	АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ В ДВОИЧНОМ КОДЕ	34
8.16.	ПРЕОБРАЗОВАНИЕ ЧИСЛА В ФОРМАТЕ BCD В ДВОИЧНЫЙ КОД	35
8.17.	ПРЕОБРАЗОВАНИЕ ДВОИЧНОГО КОДА В ФОРМАТ BCD	35
8.18.	МОДУЛЬ ИЛИ АБСОЛЮТНОЕ ЗНАЧЕНИЕ	35
8.19.	ЗНАК ЧИСЛА	35
8.20.	КОМПАРАТОР ASCII (СООБЩЕНИЕ ОПЕРАТОРА)	35
9.	ПРОЦЕДУРЫ PLC. ДИРЕКТИВЫ	36
9.1.	ВВОД ПРОГРАММЫ В ПАМЯТЬ	36
9.2.	ДИРЕКТИВЫ КОМПИЛЯЦИИ	36
9.2.1.	«Среда»	37
9.2.1.1.	Описание параметров, вводимых в «Среде»	38
9.2.1.2.	Значения параметров «Среды» для компилирования и выполнения отладки ПЛ	41
9.3.	КОМПИЛЯЦИЯ С ЗАПРОСОМ ДЕЙСТВИЙ	42
9.3.1.	Процедура компиляции	42
9.4.	БЫСТРАЯ КОМПИЛЯЦИЯ С ОПЦИЯМИ, ОБЪЯВЛЕННЫМИ В «СРЕДЕ»	43
9.5.	АТТЕСТАЦИЯ ПРОГРАММЫ ЛОГИКИ	43
9.5.1.	Создание файла для отладки	43
9.5.2.	Загрузка объектного файла для отладки	44
9.5.3.	Подключение программы отладки	44
9.5.4.	Выполнение объектного файла для отладки	45
9.5.5.	Отключение и разъединение объектной программы для отладки	45
9.5.6.	Цифровая и графическая визуализация переменных в состоянии отладки	45
9.5.6.1.	Визуализация цифрового значения переменной	45
9.5.6.2.	Графическая визуализация переменных	46
9.5.7.	Присвоение значений переменным	47
9.6.	ПЕРЕЧЕНЬ ДИРЕКТИВ, ИСПОЛЪЗУЕМЫХ PLC	47
10.	ДИАГНОСТИКА ПРИ РАБОТЕ С PLC	49
10.1.	ОШИБКИ ЯЗЫКА PLC	49
10.2.	ОШИБКИ ПРОГРАММЫ, ВЫЯВЛЯЕМЫЕ ПРИ КОМПИЛЯЦИИ	49
10.3.	ОШИБКИ КОМПИЛЯЦИИ	50
10.4.	ОШИБКИ ОТЛАДЧИКА	51
11.	ГЕНЕРАЦИЯ ФАЙЛА ОБЪЕКТНОЙ ПРОГРАММЫ ЛОГИКИ СТАНКА	52
12.	ИНТЕРФЕЙС PLC	53
12.1.	ОБЩИЕ СВЕДЕНИЯ	53
13.	ИНИЦИАЛИЗАЦИЯ УЧПУ	55
13.1.	ПРОЦЕДУРА ВКЛЮЧЕНИЕ	55
13.2.	ВКЛЮЧЕНИЕ СТАНКА ПОСЛЕ АВАРИИ	55
13.3.	ПРОЦЕДУРА «СБРОС» («RESET»)	55
14.	РЕЖИМЫ РАБОТЫ УЧПУ	56
14.1.	ОТСОЕДИНЕНИЕ ОСЕЙ	56
14.2.	ИСКЛЮЧЕНИЕ КОНТРОЛЯ УПРАВЛЕНИЯ ПРИВОДОМ	56
14.3.	ПЕРЕКЛЮЧЕНИЕ ОСЕЙ	56
14.4.	РУЧНЫЕ ПЕРЕМЕЩЕНИЯ	57
14.5.	АВТОМАТИЧЕСКИЙ РЕЖИМ	57
15.	БЛОКИРОВКИ БЕЗОПАСНОСТИ СТАНКА	59
15.1.	ОГРАНИЧЕНИЯ ПЕРЕМЕЩЕНИЙ	59
15.2.	ОГРАНИЧЕНИЯ ПЕРЕМЕЩЕНИЙ (ECDF)	59
15.3.	ОБЪЕМНАЯ ЗАЩИЩЕННАЯ ЗОНА «КУБ» (ECDF)	60

15.4.	ПРОГРАММНО-КОНТРОЛИРУЕМЫЕ ЗОНЫ (ECDF).....	61
15.5.	КОНТРОЛЬ СКОРОСТИ ОСЕЙ (ECDF).....	61
15.5.1.	Контроль нулевой скорости оси.....	61
15.5.2.	Контроль нулевой скорости вращения шпинделя.....	61
15.5.3.	Контроль скорости вращения шпинделя при значениях <i>S</i> больше нуля.....	62
15.6.	АВАРИЯ.....	62
15.7.	ПРИОСТАНОВ.....	63
15.8.	ОСТАНОВ ПЕРЕМЕЩЕНИЯ.....	63
16.	ВЫПОЛНЕНИЕ ВСПОМОГАТЕЛЬНЫХ ФУНКЦИЙ.....	65
16.1.	ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ СТАНДАРТНОГО ЦИКЛА.....	65
16.2.	ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ НАЧАЛА ПЕРЕМЕЩЕНИЯ.....	66
16.3.	ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ КОНЦА ПЕРЕМЕЩЕНИЯ.....	66
16.4.	ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ НЕМЕДЛЕННОГО ДЕЙСТВИЯ.....	67
17.	ФУНКЦИЯ «S».....	68
18.	ФУНКЦИЯ «Т».....	69
18.1.	СПОСОБЫ УПРАВЛЕНИЯ ФУНКЦИЕЙ «Т».....	69
19.	ФУНКЦИИ «М».....	72
20.	ФУНКЦИЯ «ИНДЕКСНАЯ ОСЬ».....	73
21.	УПРАВЛЕНИЕ ЗАПРОСАМИ ОТ ПЛ.....	74
21.1.	АСИНХРОННЫЕ ЗАПРОСЫ.....	74
21.1.1.	Обновление сигнала ЦАП шпинделя.....	74
21.1.2.	Сигналы управления шпинделем.....	74
21.1.3.	Позиционирование осей «от точки к точке» от ПЛ.....	76
21.1.4.	Позиционирование осей «от точки к точке» от ПЛ (ECDF).....	77
21.1.4.1.	Включение осей.....	77
21.1.4.2.	Работа оси до её выхода в ноль.....	78
21.1.4.3.	Выход оси в ноль.....	78
21.1.4.4.	Работа оси «ТТ» после выхода в ноль.....	79
21.1.5.	Общий сброс.....	80
21.1.6.	Ручные перемещения.....	80
21.1.7.	Программные перемещения оси.....	80
21.1.8.	Обновление инструмента шпинделя и коррекций.....	81
21.1.9.	Запрос на визуализацию сообщения.....	81
21.1.10.	Принудительная установка сигнала в канале ЦАП.....	82
21.1.11.	Запрос управления пультом от ПЛ.....	82
21.1.12.	Управление штурвалами. Версии ПрО Z.33Р(РИБ).....	82
21.1.13.	Режим обучения (TEACHING). Версия Z.70.10.....	83
21.1.14.	Компенсация смещения нуля привода.....	83
21.1.15.	Компенсация смещения нуля привода (ECDF).....	84
21.1.16.	Компенсация трения покоя оси на круговом контуре (ECDF).....	85
21.2.	СИНХРОННЫЕ ЗАПРОСЫ.....	85
21.2.1.	Выполнение записи файла перемещений осей от системы.....	85
21.2.2.	Запрос на выбор программы или управления с клавиатуры.....	86
21.2.3.	Запрос на «СТОП» программы.....	87
21.2.4.	Запрос на «ПУСК» программы.....	87
21.2.5.	Коммутация шпинделя.....	87
22.	АКТИВИЗАЦИЯ И ИСПОЛЬЗОВАНИЕ КОМАНДЫ «СТОП».....	88
23.	АКТИВИЗАЦИЯ И ИСПОЛЬЗОВАНИЕ КОМАНДЫ «ПУСК».....	89
24.	СИГНАЛЫ ИНТЕРФЕЙСА PLC.....	90
24.1.	СИГНАЛЫ ПАКЕТА «К».....	90
24.1.1.	Разъём I00K(26K-52K-78K-104K).....	90

24.1.2.	Разъём I01K(27K-53K-79K-105K).....	93
24.1.3.	Разъём I02K(28K-54K-80K-106K).....	93
24.1.4.	Разъём I03K(29K-55K-81K-107K).....	93
24.1.5.	Разъём I04K(30K-56K-82K-108K).....	94
24.1.6.	Разъём I05K(31K-57K-83K-109K).....	96
24.1.7.	Разъём I06K(32K-58K-84K-110K).....	97
24.1.8.	Разъём I07K(33K-59K-85K-111K).....	99
24.1.9.	Разъём I08K(34K-60K-86K-112K).....	99
24.1.10.	Разъём I09K(35K-61K-87K-113K).....	99
24.1.11.	Разъём U10K(36K-62K-88K-114K).....	101
24.1.12.	Разъём U11K(37K-63K-89K-115K).....	106
24.1.13.	Разъём U12K(38K-64K-90K-116K).....	108
24.1.14.	Разъём U13K(39K-65K-91K-117K).....	110
24.1.15.	Разъём U14K(40K-66K-92K-118K).....	111
24.1.16.	Разъёмы U15K(41K-67K-93K-119K) и U16K(42K-68K-94K-120K).....	111
24.1.17.	Разъём U17K(43K-69K-95K-121K).....	112
24.1.18.	Разъёмы U18K(44K-70K-96K-122K) и U19K(45K-71K-97K-123K).....	113
24.1.19.	Разъём U20K(46K-72K-98K-124K).....	114
24.1.20.	Разъёмы U21K(47K-73K-99K-125K) и U22K(48K-74K-100K-126K).....	114
24.2.	ТАБЛИЦА БАЗОВЫХ СИГНАЛОВ ПАКЕТА «К».....	115
24.3.	СИГНАЛЫ ПАКЕТА «N».....	138
24.3.1.	Назначение сигналов пакета «N».....	138
24.3.2.	Разъём 00N (26N-52N-78N-104N).....	138
24.3.3.	Разъём 01N (27N-53N-79N-105N).....	138
24.3.4.	Разъём 03N (29N-55N-81N-107N).....	139
24.3.5.	Разъём 04N (30N-56N-82N-108N).....	139
24.3.6.	Разъём 05N (31N-57N-83N-109N).....	139
24.3.7.	Разъём 06N (32N-58N-84N-110N).....	140
24.3.8.	Разъём 10N (36N-62N-88N-114N).....	140
24.3.9.	Разъём 15N (41N-67N-93N-119N).....	140
24.3.10.	Разъём 16N (42N-68N-94N-120N).....	141
24.4.	ТАБЛИЦА БАЗОВЫХ СИГНАЛОВ ПАКЕТА «N».....	145
25.	СИГНАЛЫ СТАНОЧНОГО ПУЛЬТА.....	154
25.1.	СИГНАЛЫ ПРОГРАММИРУЕМЫХ КНОПОК СП NC110-7.....	154
25.1.1.	Разъёмы 157K-160K.....	154
25.2.	СИГНАЛЫ СП УЧПУ NC-200, NC-201, NC-201M, NC-202, NC-210, NC-220, NC-230.....	159
25.2.1.	Разъём 02N.....	159
25.2.1.1.	Разъём 02N для NC-200, NC-210, NC-220, NC-230.....	159
25.2.1.2.	Разъём 02N для NC-201, NC-201M, NC-202.....	159
25.2.2.	Разъёмы 11N, 12N, 13N, 14N.....	162
25.3.	СИГНАЛЫ СП УЧПУ NC-310.....	165
25.3.1.	Способы организации СП УЧПУ NC-310.....	165
25.3.2.	Сигналы программируемых кнопок СП NC310-7.....	165
25.3.3.	Сигналы СП УЧПУ NC-310 в видеокадре #7.....	167
26.	БИБЛИОТЕКА ИКОНОК CNC.USR УЧПУ NC-200, NC-201, NC-201M, NC-202, NC-210, NC-220, NC-230, NC-310 (ВЕРСИИ ПРО Z.60 И ВЫШЕ).....	168
26.1.	ФАЙЛ-АРХИВ BMP32.RAR.....	168
26.2.	СОЗДАНИЕ БИБЛИОТЕКИ ИКОНОК CNC.USR.....	169
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ.....		171
ПЕРЕЧЕНЬ РИСУНКОВ.....		171
ПЕРЕЧЕНЬ ТАБЛИЦ.....		171
ПЕРЕЧЕНЬ ПРИЛОЖЕНИЙ.....		171
ПРИЛОЖЕНИЕ А (ОБЯЗАТЕЛЬНОЕ) БЛОК-СХЕМЫ АЛГОРИТМОВ ИНТЕРФЕЙСА PLC.....		172

1. СТРУКТУРА ИНТЕРФЕЙСА PLC

1.1 Интерфейс **PLC** – программный интерфейс, предназначенный для реализации логики взаимодействия между УЧПУ и управляемым оборудованием. Интерфейс **PLC** имеет свой язык, на котором составляется программа логики управляемого оборудования (ПЛ), представляющая собой часть Про, используемую для конкретизации управляемого от УЧПУ оборудования. Язык **PLC** для разработки и испытания ПЛ реализован в программе **CNC.RTB**.

Язык **PLC**, используя простые логические выражения, позволяет составлять ПЛ, которая предназначена для управления электроаппаратной частью оборудования, учитывая при этом его специфические требования. Таким образом ПЛ должна адаптировать управляемое оборудование к техническим возможностям УЧПУ.

ПЛ составляется по тем же правилам, что и УП, поэтому также как и УП, она может быть введена с клавиатуры или загружена с имеющегося у пользователя периферийного устройства. Эта программа записывается во **FLASH**. Проверка функционирования и корректировка ПЛ осуществляется на УЧПУ.

Интерфейс **PLC** обеспечивает взаимодействие базового Про УЧПУ с управляемым оборудованием через ПЛ, используя при этом конкретные алгоритмы (протоколы связи). Блок-схемы алгоритмов интерфейса **PLC** приведены в приложении **A**.

1.2 Функционирование интерфейса **PLC** обеспечивают сигналы, сгруппированные в четыре пакета. Каждый пакет имеет свою структуру и свою рабочую зону (назначение). Каждый пакет разбит на разъёмы по 32 сигнала каждый:

- «**A**» – физический пакет: 32 разъёма (00A-31A);
- «**T**» – логический пакет: 16 разъёмов (00T-15T);
- «**K**» – логический пакет; 256 разъёмов (000K-255K);
- «**N**» – логический пакет; 256 разъёмов (000N-255N).

Определение рабочих зон сигналов ПЛ:

- 1) Сигналы, представляющие собой физические входы/выходы, относятся к физическому пакету «**A**». Сигналы пакета «**A**» делятся на входные и выходные по отношению к интерфейсу **PLC**. Сигналы входа/выхода – это сигналы напряжения постоянного тока 24 В, обеспечивающие связь УЧПУ с электрооборудованием объекта управления.

За входными сигналами в пакете «**A**» закреплены разъёмы **00-03, 08-11, 16-19**, а за выходными – разъёмы **04-06, 12, 13, 20, 21**. Номера разъёмов дискретных входов/выходов, установленных в УЧПУ, должны быть объявлены в первой секции файла **IOSFIL**. Количество используемых сигналов пакета «**A**» зависит от конфигурации УЧПУ, т.е. от количества модулей дискретных входов/выходов.

- 2) Сигналы, представляющие собой константы, используемые в ПЛ, относятся к пакету «**T**». Сигналы пакета «**T**» – это 64 записи

или 512 внутренних сигналов, зарезервированных за пользователем для записи информации, сохраняющейся при отключении УЧПУ. Запись значений для этих сигналов может быть осуществлена через четвёртую секцию файла характеристики логики **IOCFIL** или из ПЛ.

- 3) Сигналы, представляющие собой рабочую память ПЛ, относятся к пакетам «**K**» и «**N**» (логические пакеты).

Определённая часть этих сигналов имеет специальное назначение, поскольку при активном состоянии выполняет определённый диалог между базовым ПрО и ПЛ для реализации требуемого пользователю цикла электроавтоматики. Эта часть сигналов является базовыми сигналами интерфейса, обеспечивающими протокол обмена между базовым ПрО и ПЛ.

Базовые сигналы интерфейса пакетов «**K**» и «**N**» указаны в таблицах разделов 24 и 25.

Часть сигналов пакета «**K**», которая не указана в таблицах, может быть использована ПЛ как рабочая память.

Пакет «**N**» находится в стадии формирования и предназначен для развития базового интерфейса. Сигналы пакета «**N**» не рекомендуется использовать в качестве пользовательских, т.к. однажды вы можете заметить, что они стали частью базового пакета «**N**», возникнут проблемы с обновлением версий ПрО, или потребуются корректировки ПЛ.

1.3 Время выполнения ПЛ прямо пропорционально её размеру. Максимальный размер ПЛ – 64 МБ.

2. ЛИНИИ СВЯЗИ МЕЖДУ МОДУЛЯМИ ПРО И PLC

2.1. Архитектура Про управления процессом представлена на рисунке 2.1.

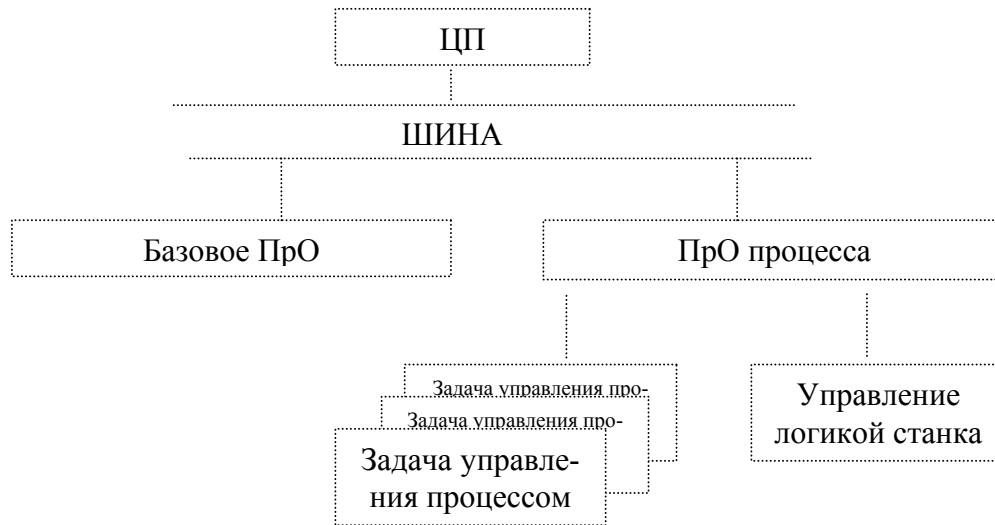


Рисунок 2.1 - Архитектура Про управления процессом

2.2. Линии связи между модулями управления логикой показаны на рисунке 2.2.

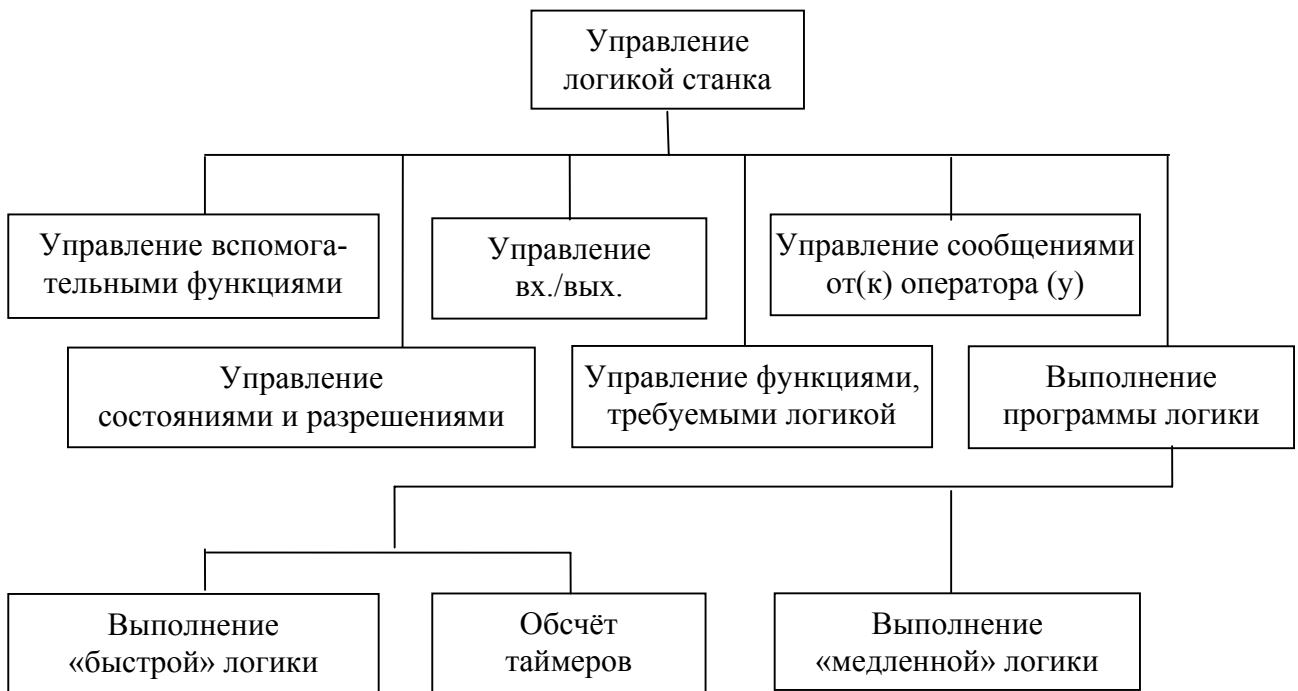


Рисунок 2.2 - Линии связи между модулями управления логикой

3. ПРИНЦИП РАБОТЫ ПЛ

3.1 ПЛ представляет собой описание на языке **PLC** аппаратной схемы, отдельные ветви которой активны в интервалах времени, зависящих от длительности различных циклов электроавтоматики управляемого оборудования.

3.2 Обработка команд ПЛ выполняется центральным процессором УЧПУ в режиме деления времени с другими функциями, такими как управление осями, анализ управляющей программы обработки детали, управление заданиями и т.п. Поэтому необходимо предварительно регулировать время исполнения событий, требующих небольшого времени реакции. С этой целью ПЛ подразделяется на две части: «быструю» и «медленную». Разделение «быстрой» части ПЛ от «медленной» осуществляется записью символа «\$» между ними.

4 . О Т Л А Д К А П Л

4.1 Отладочная система **PLC** предоставляет в распоряжение пользователя следующую методику работы с ПЛ:

- 1) компиляция ПЛ (файла программы-источника или файлов программ-источников) в объектный файл;
- 2) отладка объектной программы;
- 3) исправление программы.

4.2 Запуск компилятора и отладчика **PLC** выполняется из главного меню режима «**КОМАНДА**» по клавише «**F3**» (опция «**PLC**»).

5. ЯЗЫК PLC

Язык программирования **PLC** обеспечивает разработку программы логики станка с минимальным взаимодействием между УЧПУ и станком. От пользователя языка потребуется знание алгебры Буля и понимание различных систем счисления:

- 1) десятичной;
- 2) двоичной;
- 3) двоично-десятичной (формат BCD);
- 4) восьмеричной;
- 5) шестнадцатеричной.

5.1. Элементы языка

Основным элементом языка является команда, посредством которой можно выполнять определенную функцию. Команда состоит из элементов языка, составляемых по определенным правилам.

Данные элементы могут подразделяться на 4 группы:

- 1) операнды;
- 2) метаоперанды;
- 3) функции;
- 4) операторы.

5.1.1. Операнды

Операнды - часть языка, содержащая обрабатываемую информацию. В состав языка входят операнды типа «СИГНАЛ» (элементарная информация по состояниям ВКЛ./ВЫКЛ.) и типа «СЛОВО» (информация, состоящая из 8 сигналов).

Под словом «операнд» понимаются также константы.

5.1.1.1. Сигналы

Сигналы определяются следующим образом:

Sxxuzz,

где:

S - определяет тип сигнала и может быть:

- 1) **I** - сигнал на входе (считывание),
- 2) **U** - сигнал на выходе (запись);

xx - определяет номер разъёма в пакете.

Условно под разъёмом понимается группа из 32 отдельных сигналов или 4 слов. При этом сигналы будут иметь сплошную нумерацию от 0 до 31, а слова - от 0 до 3.

у - определяет тип пакета, которому принадлежит разъём:

- 1) **A** - пакет «**A**» (физический пакет);
- 2) **K** - пакет «**K**» (логический пакет);
- 3) **N** - пакет «**N**» (логический пакет);
- 4) **T** - пакет «**T**» (логический пакет);

Идентификаторы **xx** и **y** связаны между собой:

$$\begin{array}{ll}
 \mathbf{y} = \mathbf{A} & 0 \leq \mathbf{xx} \leq 31; \\
 \mathbf{y} = \mathbf{K} & 0 \leq \mathbf{xx} \leq 255; \\
 \mathbf{y} = \mathbf{N} & 0 \leq \mathbf{xx} \leq 255; \\
 \mathbf{y} = \mathbf{T} & 0 \leq \mathbf{xx} \leq 15.
 \end{array}$$

zz - определяет номер сигнала в разъёме.

Пример

Определим сигнал **I1A16**:

- I** - входной сигнал;
- 1** - номер разъёма из 32 сигналов;
- A** - пакет принадлежности разъёма 1;
- 16** - номер сигнала в разъёме 1 пакета «**A**».

Примечание - Соответствие сигналов пакета «**A**» контактам разъёмов на модуле дискретных вх./вых. (**I/O**) приведено в документе «Руководство по эксплуатации».

В пакете «**K**» имеются виртуальные (мнимые) разъёмы внутренней памяти (номера их расположены сразу за последним номером, указанным в таблице базового пакета «**K**»). Они определяются как промежуточные сигналы протокола обмена между базовым ПрО и ПЛ, могут назначаться пользователем. Сигналы базового интерфейса определены и описаны в данном документе.

Пакет «**T**» определяет виртуальные (мнимые) разъёмы сигналов внутренней памяти, сохраняющиеся при отключении УЧПУ.

5.1.1.2. Слово

Слова определяются как для входных, так и для выходных операций. Идентификаторы слова **xx** и **y** имеют те же значения, что и для сигналов.

z - определяет группу из 8 сигналов внутри разъёма и может принимать значения от 0 до 3 (включительно).

Пример

Определим слово **W1A0**:

- W** - группа из 8 сигналов;
- 1** - номер разъёма из 32 сигналов;
- A** - пакет принадлежности разъёма 1;
- 0** - первая группа разъёма 1 пакета «**A**», состоящая из сигналов от 0 до 7.

5.1.1.3. Константы

Константы принадлежат группе слов, которые состоят из 8 сигналов. Они могут быть выражены в формате следующим образом:

zzzy,

где:

- zzz** - набор цифр, определяющий константу. Если первая цифра является «0», то она не должна быть опущена;
- y** - определяет формат константы:
 - D** - десятичный формат;
 - O** - восьмеричный формат;
 - H** - шестнадцатеричный формат.

5.1.1.4. Логические уровни и значения переменных

Имеются два класса переменных:

- 1) единичный сигнал (бит);
- 2) группа из 8 сигналов (слово или байт).

Единичный сигнал может принимать только два значения: «0» или «1».

Физическим сигналам вх./вых. на уровне логической «1» соответствует напряжение +24В.

Слово может принимать значения:

- 1) десятичное от 0 до 255;
- 2) шестнадцатеричное от 0 до FFH;
- 3) восьмеричное от 0 до 3770.

5.1.2. Метаоперанды

Кроме элементарных операндов имеются также сложные операнды, так называемые «МЕТАОПЕРАНДЫ». К ним относятся особые электронные компоненты, такие как:

- 1) таймеры;
- 2) счетчики;
- 3) генератор импульсов;
- 4) компараторы;
- 5) компараторы ASCII.

5.1.2.1. Программируемые таймеры

Имеется 48 длительных таймеров, которые могут быть запрограммированы на время от 1/10 сек до 255/10 сек, и 16 коротких таймеров - от 1/100 сек до 255/100 сек. С функциональной точки зрения таймер представляет собой элемент, определяемый входом, сигналами разрешения и задержки счёта и импульсным выходом.

Таймер задаётся буквой «Т» по формату:

Txxу,

где:

- xx** - определяет номер таймера 0-63;
- у** - определяет сигнал таймера, т.е. логический выход данного компонента, и может принимать следующие значения:

- 1) **I** - входной сигнал;
- 2) **A** - разрешающий сигнал (инвертируемый);
- 3) **U** - выходной сигнал;
- 4) **D** - производный сигнал (является инверсным по отношению к **U**).

Сигналы **TxxA** и **TxxI** могут быть как входными, так и выходными. Сигналы **TxxD** и **TxxU** являются только выходными сигналами, определяющими состояние таймера.

Если имеется сигнал **TxxI** на выходе, формат определения сигнала становится:

TxxI (время выдержки),

где:

- время выдержки** - определяется в базовых единицах времени. Базовое время выражено:

- десятые доли секунды для таймеров с номерами 0-47;
- сотые доли секунды для таймеров с номерами 48-63.

Параметр «**время выдержки**» может быть задан константой или параметром - словом. В последнем случае значение времени будет меняться в зависимости от содержимого выбранного слова.

С логической точки зрения таймер можно рассматривать как компонент с четырьмя связями: 2 - на входе, 2 - на выходе. Схема таймера представлена на рисунке 5.1.

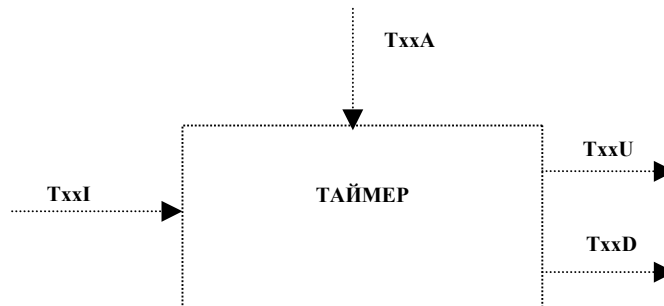
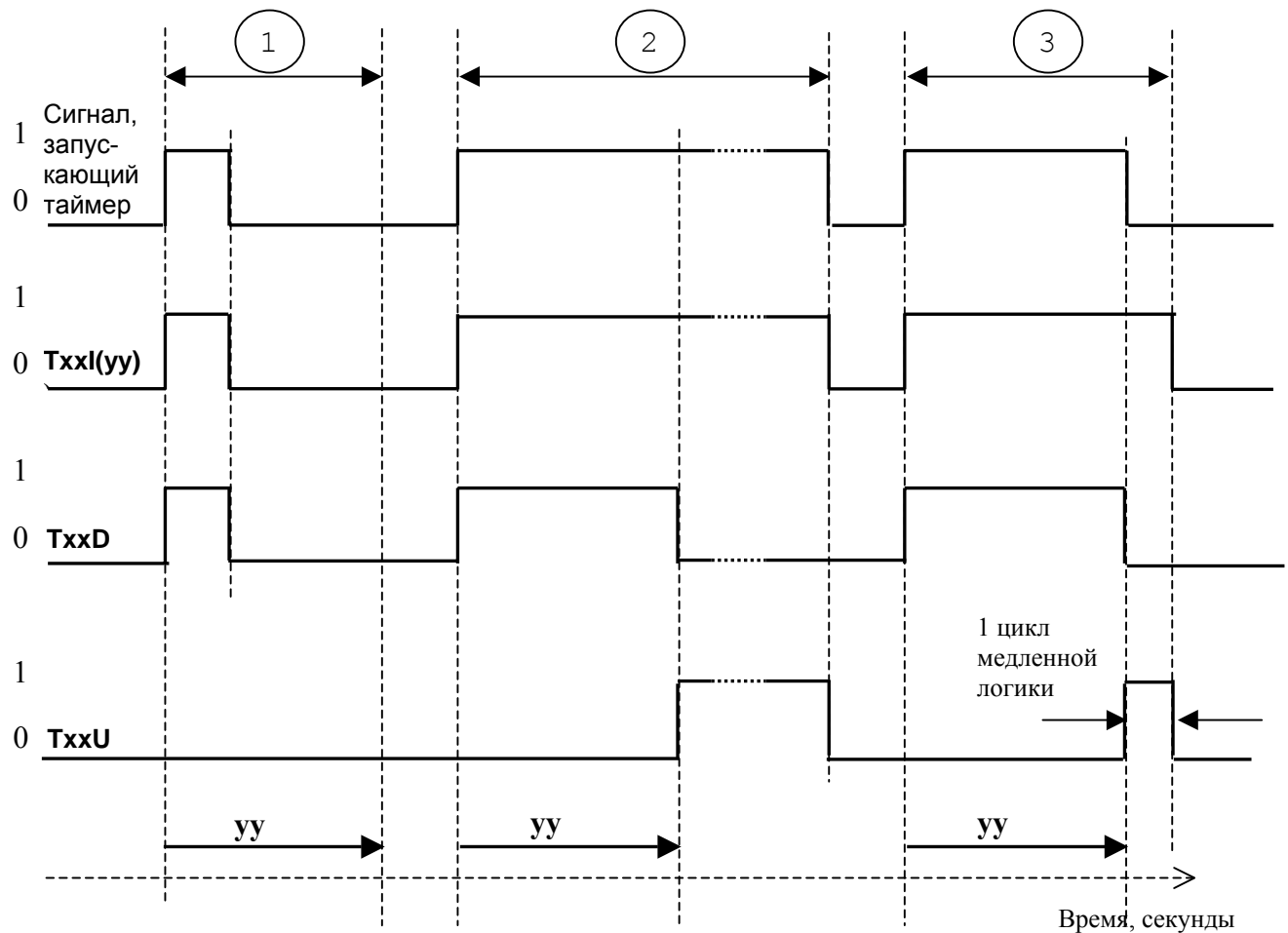


Рисунок 5.1 - Схема таймера

Таймер начинает счёт, когда на вход **TxxI** подана логическая «1». Если за время счёта установится в «1» сигнал **TxxA**, счёт остановится

до пропадания **TxxA**. Сигнал **TxxU** будет на логическом уровне «1» по окончании счёта таймера.

Пример диаграммы работы таймера изображён на рисунке 5.2.



yy - время выдержки

① - длительность запускающего сигнала менее **yy**

② - длительность запускающего сигнала более **yy**

③ - длительность запускающего сигнала равна **yy**

Рисунок 5.2 - Диаграмма работы таймера

Примеры задания таймеров:

- явный способ задания длительного таймера на 10 секунд:
T00I(100)= /U200K0;
- явный способ задания короткого таймера на 2,5 секунды:
T48I(250)= /U200K0;
- неявный способ задания длительного таймера на 10 секунд:

При задании значения времени в неявном виде его необходимо считать по следующей формуле:

$$\frac{\text{Требуемое время работы таймера, с}}{\text{Время, объявленное в инструкции TIM}} * N$$

где:

N - может принимать значение:

- 100, если расчёт выполняется для длительного таймера (номера таймеров 0-47);
- 1000, если расчёт выполняется для короткого таймера (номера таймеров 48-63).

В нашем случае:

- требуемое время работы таймера - 10 с;
- константа - 100;
- время, объявленное в инструкции **TIM** - 2 мс.

Рассчитанное по формуле значение равно 500. Это значение в шестнадцатичном коде равно **1F4**. Для записи этого значения требуется два соседних байта, например, **W250K0** (младший) и **W250K1** (старший). Тогда окончательно необходимо записать:

```
W250K0=0F4H
W250K1=1H
T00I(W250K0)=/U200K0
```

При использовании задания параметров из секции 4 файла **IOCFIL** (пакет «Т») значение **1F4** необходимо записать в следующем виде:

```
*4
T01=F4
T02=01
Тогда в программе логики станка PLC необходимо записать:
T00I(W00T0)=/U200K0
```

Для задания значений времени таймерам необходимо помнить, что для этого используется 2 байта. Если для записи значения достаточно только младшего байта, старший байт не должен использоваться в данной ПЛ.

5.1.2.2. Программные счётчики

Существуют 100 счётчиков с модулями счёта (программируются) от 2 до 255.

Функционально они обеспечивают вход, пересылку из входа на выход, выход типа **WORD**. Счётчик обозначается буквой «С» в следующем формате:

СХХУ,

где:

- XX** - определяет номер счётчика (0-99);
- Y** - определяет тип обрабатываемого сигнала, он может быть:
 - I** - входной сигнал;
 - R** - сигнал переноса;
 - Z** - восстановление (сброс счета);
 - A** - метод счета (уменьшение/увеличение);
 - W** - слово, содержащее величину счета.

Сигнал **CXXR** может быть только выходным, в то время как все другие сигналы могут быть как входными, так и выходными.

Сигнал **CXXI** можно определить как выходной, используя определённое выражение:

CXXI (счёт),

где величина «счёт» определяет модуль счёта.

Так же как и таймеры, счётчики могут быть абсолютного типа, когда величина счёта равна константе, или параметрического, когда величина счёта равна **WORD** (см. примечание при описании таймеров). В этом случае модуль счёта будет изменяться сразу, как только изменится содержимое **WORD**.

CXXA определяет направление счёта. Если значение равно «1», счётчик работает в режиме обратного счёта, если «0» - в режиме прямого счёта.

С логической точки зрения счётчик можно рассматривать как элемент, имеющий три входных сигнала и один выходной с присвоенным значением счёта.

Схема счётчика представлена на рисунке 5.3.

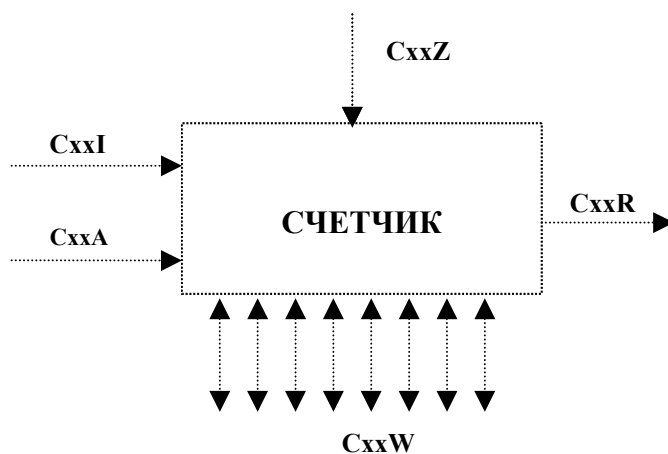


Рисунок 5.3 - Схема счётчика

Значение счёта может зависеть от варианта входных сигналов, или может быть задано прямым назначением.

Счёт активизируется сигналом **CXXI**, а **CXXW** будет увеличиваться или уменьшаться в зависимости от задания **CXXA**.

CXXR становится «1» при достижении модулем счёта заданной величины.

Значение **CXXA** очень важно. Так если **CXXA=0**, **CXXR** принимает значение «1», когда **CXXW** будет равно заданной величине счёта (после которой **CXXW=0**). Если **CXXA=1**, **CXXR** будет равно «1», когда счёт станет отрицательным (после чего **CXXW = «счёт»-1**).

Когда «1» приходит на **CXXZ**, счётчик сбрасывается. Пока он в таком состоянии на нём невозможны никакие операции за исключением прямого назначения. При сбросе счётчик устанавливает модуль счёта и все остальные сигналы в «0» (кроме **CXXZ=1**).

5.1.2.3. Одновибраторы (генератор импульсов)

Существует 100 одновибраторов. Одновибратор определяется символом

Pxx,

где:

xx - определяет номер одновибратора, имеющего значение от 0 до 99 (включительно).

Сигнал **Pxx** активизируется по переднему фронту и остаётся на уровне логической «1» в течение одного цикла «медленной» части логики. В течение данной фазы сигнал не подвергается никаким изменениям.

Данный сигнал может быть как на входе, так и на выходе. Если этот сигнал находится на входе, считывается текущее состояние логической микросхемы, а если на выходе - выполняется вышеопределённая функция. Работа одновибратора иллюстрируется диаграммой рисунка 5.4.

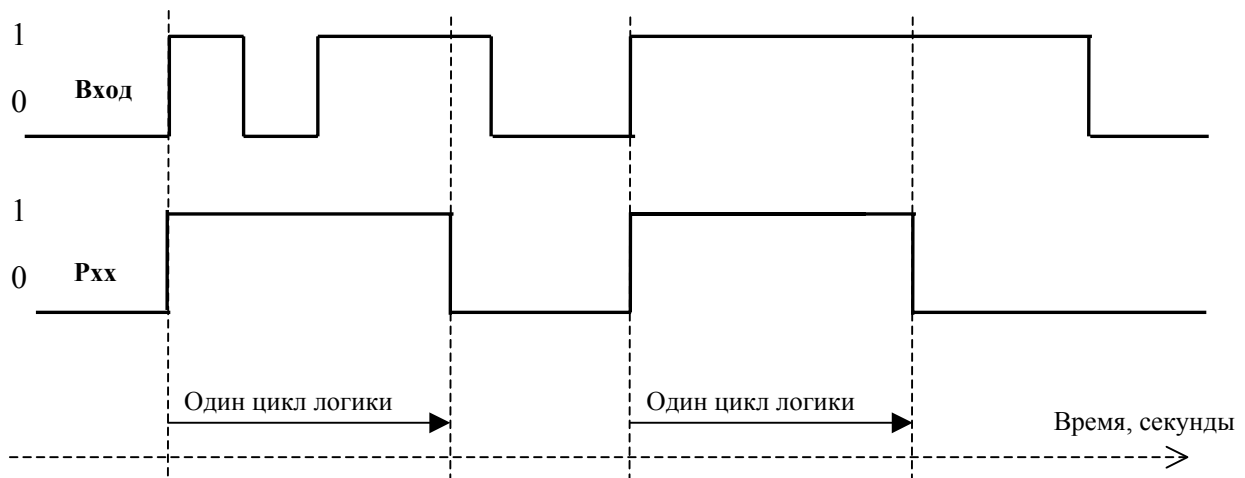


Рисунок 5.4 - Диаграмма работы одновибратора

Пример

P6=I1A3. Сигнал **P6** поднимается с **I1A3** и остается равным «1» в течение одного цикла «медленной» части логики. Диаграмма работы одновибратора в данном случае представлена на рисунке 5.5.

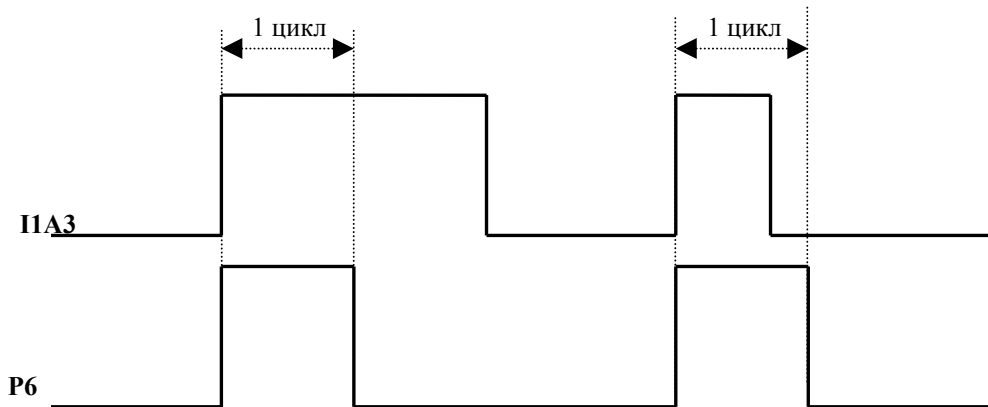


Рисунок 5.5 - Диаграмма работы одновибратора

5.1.2.4. Компараторы

Компаратор идентифицируется символом:

[WORD1 компаратор WORD2].

Он выполняет тест между двумя словами: компаратор определяет данный тип теста и может быть:

- 1) **«>»** ----- **WORD1** больше **WORD2**;
- 2) **«=»** ----- **WORD1** равно **WORD2**;
- 3) **«<»** ----- **WORD1** меньше **WORD2**.

Операцию сравнения между сигналами или словами необходимо заключить между служебными символами «[« и «]». Можно отрицать операцию сравнения слов, указывая символ «/» перед открытой квадратной скобкой.

Пример

U23K12 = /[W15K1 = W13T3].

Сигнал U23K12 будет на уровне «1», если 2 слова: W15K1 и W13T3 будут отличаться.

Следует обратить внимание на тип кода, использованного в словах сравнения, с тем, чтобы он был одним для всех слов (двоичный, BCD или другой).

Например, при необходимости создания сигнала декодирования вспомогательной функции M25, активной в конце хода осей, будет использовано следующее выражение:

U100K15 = [W03K0 = 25H] */U27K3,

где:

- 1) **U100K15** = декодирование M25;
- 2) **W03K0** = слово передачи функций «М» от базового ПрО;
- 3) **25H** = код BCD (шестнадцатеричный) числа 25;
- 4) **U27K3** = сигнал, который равняется «0» в конце хода.

5.1.2.5. Компаратор ASCII

Компаратор **ASCII** определяется форматом типа:

«**MESS**»,

где:

MESS – последовательность знаков **ASCII** (максимум 32).

Если эта строка соответствует (равна) строке, содержащейся в какой-то ячейке памяти, текущий сигнал устанавливается на логическом уровне «**1**», иначе сигнал устанавливается на уровне «**0**». Этот сигнал – импульсного типа, продолжительностью 2 цикла логики.

Последовательность «**MESS**» вводится с клавиатуры и засылается посредством выбора опции меню «**ПОСЛАТЬ В ЛОГИКУ**» в видеокадре **#1** режима «**УПРАВЛЕНИЕ СТАНКОМ**».

Пример

U04A3 = U12K6 * «**RIFCU**»

Это выражение устанавливает третий контакт четвёртого разъёма пакета «**A**» равным «**1**», если U12K6 = 1, и «**RIFCU**» задано с клавиатуры. Выход остается на «**1**» в течение 2-х циклов логики.

5.1.3. Функции

Под функцией понимается преобразование значения данных. Они определяются трёхбуквенным кодом.

Имеются следующие функции:

- 1) шифратор;
- 2) дешифратор;
- 3) преобразование в BCD;
- 4) преобразование в двоичный код;
- 5) знак;
- 6) модуль;
- 7) мультиплексор;
- 8) полуслово;
- 9) условные переходы.

Очень часто функции относятся к особым компонентам аппаратных средств.

5.1.3.1. Шифратор (кодирование)

Данная функция выражается в формате:

ENC (WORD).

Под шифратором понимается логический компонент с 16 выводами: 8 выводов входных и 8 выводов выходных.

Функция состоит в преобразовании значения позиционного положения числа, содержащегося в слове (операнд функции), в двоичное значение, учитываемое полностью как слово. Если данное **WORD** имеет более одного бита на логическом уровне «**1**», кодируется только старший бит.

Значение данной операции можно увидеть в следующей таблице, в которой для каждого входа приведён соответствующий выход (двоичные значения). Символы **X** определяют незначимые сигналы.

Таблица соответствия вх./вых. (двоичные значения) для шифратора:

ВХОД	ВЫХОД
00000000	00000000
00000001	00000001
0000001X	00000010
000001XX	00000011
00001XXX	00000100
0001XXXX	00000101
001XXXXX	00000110
01XXXXXX	00000111
1XXXXXXX	00001000

5.1.3.2. Дешифратор

Данная функция выражается в формате:

DEC (WORD) .

Под дешифратором понимается логический компонент с 16 выводами: 8 выводов входных и 8 выводов выходных.

Данная функция преобразует двоичное значение любого слова или выхода счётчика (операнд функции) в слово, содержащее значение позиционного положения.

Таблица соответствия вх./вых. (двоичные значения) для шифратора:

ВХОД	ВЫХОД
00000000	00000000
00000001	00000001
00000010	00000010
00000011	00000100
00000100	00001000
00000101	00010000
00000110	00100000
00000111	01000000
00001000	10000000

Для всех остальных входных значений выход равен «0».

5.1.3.3. Преобразование в BCD

Выражается в формате:

BCD (WORD) .

Преобразует двоичное содержание слова (параметр функции) в значение в формате **BCD**, также определяемое словом.

Результат преобразования можно считать правильным в случае, если содержание входного слова меньше двоичного значения 100 (исключ.).

5.1.3.4. Преобразование в двоичный код

Выражается в формате:

BIN (WORD) .

Преобразует содержание слова в формате BCD (параметр функции) в двоичное значение, также определяемое словом. Результат преобразования можно считать правильным в случае, если входное слово содержит число в формате BCD.

5.1.3.5. Знак

Выражается в формате:

SGN [математическое выражение] .

Представляет на выходе сигнал, который устанавливается на уровне «1» при отрицательном результате математического выражения и на уровне «0» – в противном случае.

Пример

U100K8 = SGN [C00W-W1T1].

5.1.3.6. Модуль

Выражается в формате:

[ABS (математическое выражение)] .

Представляет на выходе слово, содержащее модуль выполненного математического выражения.

Пример

W100K0 = (17 - 19)

W100K1 = [ABS(W100K0)]

5.1.3.7. Мультиплексор

Мультиплексор определяется форматом:

MUX (WORD1, ..., WORD8), (SIG1, ..., SIG8) .

Мультиплексор представляет на выходе слово, содержащее значение **WORD1**, если сигнал **SIG1** находится на логическом уровне «1». Если он находится на уровне «0» – будут анализироваться последующие два элемента. На самом деле, выходному слову присваивается значение **WORDn**, если сигнал **SIGn** находится в состоянии «1» с приоритетом, уменьшающимся в правую сторону. Если ни один сигнал **SIG** не установлен в «1», ни одному выходу не присваивается значение.

Примеры

1) $W56K1 = \text{MUX}(W1A0), (U27K5)$.

Выход будет иметь значение $W1A0$, когда $U27K5 = 1$, и не будет изменяться при $U27K5 = 0$.

2) $W55K3 = \text{MUX}(12H, 25H), (U101K4, U101K5)$.

Выход преобразуется от значения $12H$ при $U101K4 = 1$ в значение $25H$ при $U101K4 = 0$ и $U101K5 = 1$.

3) $W56K1 = \text{MUX}(W0A0, W0A1, 16H), (U101K3, U101K4, U101K5)$.

Выход приобретает значение $W0A0$, если $U101K3 = 1$ или $W0A1$, если $U101K4 = 1$ или значение $16H$, если $U101K5 = 1$. Если ни одно из значений не равно «1», выход не изменяется.

5.1.3.8. Полуслово

Данная функция может быть определена в трёх форматах:

1) **HIG (WORD)** - представляет на выходе слово, содержащее значение четырёх старших битов, выравненных справа.

Пример

ВХОД	ВЫХОД
01011111	00000101
11110001	00001111
00001111	00000000

2) **LOW (WORD)** - представляет на выходе слово, содержащее значение четырёх младших битов.

Пример

ВХОД	ВЫХОД
01011111	00001111
11110001	00000001
11101010	00001010

3) **XCH (WORD)** - обменяет четыре старших бита с четырьмя младшими битами и наоборот.

Пример

ВХОД	ВЫХОД
01101011	10110110
11110000	00001111

5.1.3.9. Условные переходы

Данная функция определяется общим форматом:

DOF [имя блока]: <условие> ... ENDF [имя блока]

DOE [имя блока]: <условие> ... ENDE [имя блока] ,

где буква после **DO** определяет тип инструкции, в частности: **IF**, **ELSE** по следующим условиям:

- 1) **DOF [имя блока]: <условие>**
 <блок уравнений>
END F [имя блока]

При проверке условия обрабатывается блок уравнений, включённый между инструкцией **DOF** и концом инструкции **END F**. Если условие оказывается ложным, выполняется переход на первое уравнение после **END F**.

Примеры

а) 008 0 **DOF BLOCK1: U100K0**
 009 1 U100K3=U100K2
 010 1 U100K4=U100K3 * U00T0
 011 1 **END F BLOCK1**
 012 0 W13K=MUX(W1A0), (I1A1)

Если в течение выполнения программы в кадре 008 подтверждается условие **U100K0=1**, обрабатываются уравнения до инструкции **END F**, в противном случае выполняется переход к кадру 012.

б) 008 0 **DOF BLOCK1: U225K0**
 009 1 U100K0=U100K5
 010 1 U100K1=U100K8 * U00T0
 011 1 **DOF BLOCK2: U123K0**
 012 2 U100K3=U100K2
 013 2 U100K4=U100K6 * U00T0
 014 1 **END F BLOCK2**
 015 1 U100K10=U100K2
 016 1 U100K12=U100K3 * U00T0
 017 1 **END F BLOCK1**
 018 0 W13K=MUX(W1A0), (I1A1)

Максимальное количество уровней вложенности равно 9. Кадры 12 и 13 будут выполнены, если сигналы **U225K0** и **U123K0** равны «1».

- 2) **DOF: <условие>**
 <1-ый блок уравнений >
END F
DOE: <условие>
 <2-ой блок уравнений>
END E
DOE: [условие]
 <n-й блок уравнений>
END E

Данная инструкция обеспечивает обработку блока уравнений, соответствующего инструкции **DOF...ENDF**, при истинном условии, в противном случае - обработку последующих блоков **DOE...ENDE** также при проверке соответствующего условия.

Эти блоки уравнений также исключаются между собой, т.е. они учитываются последовательным способом, и после отработки первого блока (с истинным условием) выполняется переход к первому блоку программы после инструкции **END E**, соответствующей последнему блоку инструкций. В этой инструкции можно пропустить «условие». В этом случае при проверке инструкция всегда выполняется (всегда учитывается как истинное условие).

Пример

```

008 0   DOF: U100K1
009 1       U100K3=U100K2
010 1       U100K4=U100K3*U00T0
011 1   END F
012 0   DOE: U100K2
013 1       W28K1=0FH
014 1       U100K1=[W28K1=0FH] * IOA4
015 1   END E
016 0   DOE: U100K3
017 1       W101K0=MUX(W28K1), (IOA4)
018 1   END E
019 0   DOE:
020 1       U104K0=U100K0
021 1   END E
022 0       W102K0=BIN(W102K1)
    
```

Если при выполнении программы подтверждается условие **U100K10=1**, выполняется цикл от блока 008 до блока 011, а затем происходит переход к блоку 022. При ложном условии немедленно осуществляется переход к блоку 012 и рассматривается новое условие с теми же ограничениями упомянутой выше инструкции.

Если в течение последовательной проверки не подтверждается никакого условия, будет обрабатываться последний блок уравнений, относящийся к инструкции **DOE**, поскольку условия нет. Необходимо, чтобы в указанном выше цикле инструкций первая инструкция была **DOF**.

5.1.4. Операторы

Операторы представляют собой элементы языка, которые указывают способы взаимосвязи операндов, метаоперандов и функций.

5.1.4.1. Присвоение

Представлено символом «=». Этим символом выполняется равенство между оператором или метаоперандом, находящимся слева от символа, с выражением, находящимся справа. При этом значение выражения передается в элемент, предшествующий символу «=».

5.1.4.2. Логические операторы

Логические операторы - это операторы, осуществляющие логическую связь операндов, между которыми они находятся.

- **NOT** – выражается символом «/» перед отрицаемым операндом;
- **AND** – выражается символом «*» между операндами, устанавливаемыми на **AND**;
- **OR** – выражается символом «+» между операндами, устанавливаемыми на **OR**;
- **XOR** – выражается символом «&» между операндами, устанавливаемыми на **XOR**.

Примечание – Оператор **AND** имеет высший приоритет по отношению к **OR** и **XOR**.

5.1.4.3. Применение скобок для выполнения логических схем

Выражение типа $U04A5 = (I1A1 + I1A2) * U04A6$ необходимо применять, если соответствующая схема имеет вид, представленный на рисунке 5.6.

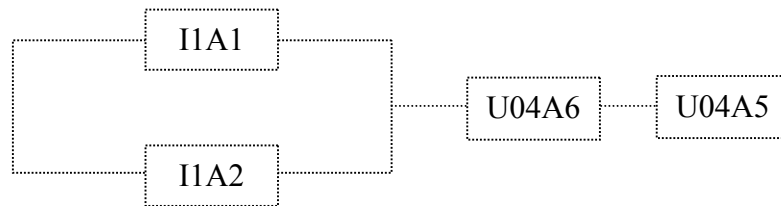


Рисунок 5.6 – Применение скобок в логических схемах. Пример 1

Поскольку при выполнении операций соблюдается приоритет операций, предусмотренных в арифметических выражениях, оператор **AND** имеет высший приоритет по отношению к **OR** и **XOR**.

Если схема имеет вид, представленный на рисунке 5.7, правильное выражение было бы $U04A5 = I1A1 + (I1A2 * U102K6)$. Его можно написать также $U04A5 = I1A1 + I1A2 * U102K6$ с экономией времени и памяти, благодаря устранению скобок.



Рисунок 5.7 – Применение скобок в логических схемах. Пример 2

5.1.4.4. Арифметические операторы

Арифметические операторы – это операторы, которые активизируют арифметическую операцию операндов, среди которых они находятся.

Сложение – представляется символом «+» между слагаемыми операндами.

Вычитание - представляется символом «-» между вычитаемыми операндами.

Примечание - Нет никакого приоритета в пределах данного класса операторов. Невозможно перепутать между оператором «+», определяемым как «ИЛИ» и «+», определяемым как сложение, поскольку они выполняются по-разному.

Синтаксис следующий:

Wxxyz = [АРИФМЕТИЧЕСКАЯ ОПЕРАЦИЯ] ,

где:

АРИФМЕТИЧЕСКАЯ ОПЕРАЦИЯ = ОПЕРАНД - ОПЕРАТОР - ОПЕРАНД

ОПЕРАНД - ABS (<Арифметическая операция>) модуль операции

DEC (Wxxyz) - дешифратор слова

ENC (Wxxyz) - шифратор слова

BIN (WXXYZ) - двоичное представление формата десятичного числа в формате BCD на слове.

Пример

W100K0=[ABS (W0A0-W0A1)]

W100K1=[DEC (W0A0) -ENC (W0A1)]

W100K2=[W0A0-W0A1) - (W0A2-W0A3)]

W100K3=[W0A0-W0A1]

Числа **X** могут быть представлены следующим образом:

- 128 < X < 128 .

Все отрицательные числа имеют старший бит (значение 2000), равный «1».

Все положительные числа имеют старший бит, равный «0».

Пример

-1 представляется 8 битами 11111111;

+1 представляется 8 битами 00000001.

Для преобразования слова, представленного как отрицательное значение, в слово в абсолютном значении необходимо располагать в обратном порядке, начиная с младшего бита, все биты после первого, находящегося на «1».

6. СТРУКТУРА ЯЗЫКА

Главным элементом структуры языка является **ОПЕРАТОР** или инструкция, посредством которой даётся команда.

Существует два типа операторов:

- 1) оператор описания;
- 2) исполнительный оператор.

6.1. Операторы описания

Операторы описания представляют собой класс инструкций, не требующих никакой операции выполнения, но предоставляющих компилятору некоторую информацию.

Операторы описания внутри языка **PLC**:

- 1) **СООБЩЕНИЯ;**
- 2) **ЦИКЛ;**
- 3) **КОММЕНТАРИЙ.**

6.1.1. Сообщения для оператора

Разработчик ПЛ может создать файл из 255 сообщений по 42 знака каждое. Данные сообщения должны информировать оператора об аварийной ситуации на станке, текущем цикле, причине остановки станка, возможных командах для продолжения обработки или переналадки станка и т.д.

6.1.2. Цикл

Определяется в формате: **\$**.

При этом подтверждается, что последующие инструкции будут принадлежать «медленному» циклу программы логики.

Примечание - После знака «**\$**» может быть записан комментарий.

6.1.3. Комментарии в программе

Очень важно иметь возможность включения в текст ПЛ некоторых комментариев, чтобы лучше понимать документацию программы. При записи в программу комментариев необходимо перед ним ставить символ «**;**». Количество комментариев в исходном тексте ПЛ не влияет на размер скомпилированной программы.

6.1.4. Символьное представление ПЛ

Прямое обращение к сигналам и словам ПЛ (**U10K0**, **I00K1**, **W12K1**, **U235K0** и т.д.) может быть заменено программистом на любые символьные («**MUSPE**», «**RESET**», «**STRMAS**», «**QQQQ**»). Для этого программист должен создать файл соответствия. Первым символом этого файла должен быть символ «*****». Количество присвоений определяется значением, которое записано в инструкции **SPL** (секция 1 файл **IOCFIL**).

Пример

★

MUSPE = U10K0
COMU = U10K24
CEFA = U10K25
ASTOOL= U10K21

Данный файл должен компилироваться первым (до компиляции ПЛ).

В ПЛ символьные имена сигналов должны использоваться со знаком «!».

Пример

!STRMAS=10
!ASTOOL=I00A20*[C00W=6]+!ASTOOL*/I00K26*/I00K27.

6.2. Исполнительные операторы

Исполнительные операторы - это операторы, определяющие электрические сети станка. Подразделяются на две группы:

- 1) для сигналов;
- 2) для слов.

6.2.1. Операторы для сигналов

Операторы для сигналов определяют логические ветви программы, относящиеся к сигналам, и существуют в формате:

SIGN = выражение по сигналу,

где:

SIGN - является любым операндом или метаоперандом, принятым как выходной сигнал;

«выражение по сигналу» - представляет собой набор операндов, метаоперандов или функций (имеющих сигнал в качестве выхода), связанных между собой посредством логических операторов (возможно с помощью круглых скобок).

6.2.2. Операторы для слов

Операторы для слов определяют логические ветви программы, относящиеся к группам сигналов (слов), и существуют в формате:

слово = выражение по слову,

где:

«слово» - представляет собой любой операнд или метаоперанд, принятый как выходное слово;

«выражение по слову» - может быть:
- логическое;

- функция **MUX**.

Если «**Выражение по слову**» - логическое, оно состоит из набора операндов, метаоперандов или функций (имеющих как выход слово, за исключением **MUX**), соединённых между собой логическими операторами (возможно с помощью круглых скобок).

Функция **MUX** должна записываться сразу после символа «**=**»

7. ПОРЯДОК ПРИМЕНЕНИЯ ЯЗЫКА PLC

Внутри программы исполнительные операторы могут отличаться между собой: имеются быстрые и медленные исполнительные операторы. Быстрые операторы включены между началом программы и первой инструкцией **ЦИКЛ \$**; данная часть называется «быстрая логика». Все остальные операторы являются медленными и составляют «медленную логику». Сама система подразделяет «медленную логику» на блоки в зависимости от времени исполнения программы, объявленного в файле характеристики **IOCFIL**.

8. КРАТКОЕ ИЗЛОЖЕНИЕ ЯЗЫКА PLC

8.1. Элементарные операции

- 1) / - НЕ;
- 2) * - И;
- 3) + - ИЛИ;
- 4) & - исключение ИЛИ;
- 5) = - равенство или выход.

8.2. Таймеры (постоянные или параметрические)

- 1) 48 на 255 десятых сек. (0-47);
- 2) 16 на 255 сотых сек. (48-63);
- 3) **TnnI** (время) = вход таймера;
- 4) **TnnA** = остановка счёта;
- 5) **TnnU** = выход таймера;
- 6) **TnnD** = сигнал производный.

8.3. Счетчики (постоянные или параметрические)

- 1) 100 счётчиков с модулем счёта от 2 до 255;
- 2) **CnnI** (модуль) = вход счётчика;
- 3) **CnnR** = выход переноса;
- 4) **CnnW** = текущее значение;
- 5) **CnnZ** = вход сброса;
- 6) **CnnA** = счет вперед, назад.

8.4. Компараторы с 8 битами

- | | |
|-----------------|------------------------------------|
| [..... > | : больший; |
| [..... = | : равный; |
| [..... < | : меньший; |
| /[..... = | : не равный; |
| /[..... < | : больший или равный (не меньший); |
| /[..... > | : меньший или равный (не больший). |

8.5. Декодирование 8 выходов

- DEC (Wxxxхуz)** = декодирование слова **xxxхуz**;
DEC (CXXW) = декодирование счетчика **XX**.

8.6. Кодирование в двоичном коде 8 сигналов

- ENC (Wxxxхуz)** : значение старшего бита слова (0-7).

8.7. Отдельные сигналы

I xxxuzz : вход;
U xxxuzz : выход.

8.8. Сложные сигналы из 8 бит: слово

W xxxuz : группа из 8 входных или выходных сигналов.

8.9. Сложные сигналы из 4 бит: полуслово

HIG (Wxxxuz) : выделяет 4 старших бита;
LOW (Wxxxuz) : выделяет 4 младших бита;
XCH (Wxxxuz) : обменивает 4 старших бита на 4 младших.

8.10. Условные переход

DOF : выполняет с действующим условием;
DOE : выполняет с действующим условием.

8.11. Обозначение сигналов

y = пакет A/K/N/T;
xxx = номер разъёма в пакете от 0 до 255;
zz = номер сигнала (бит) в разъёме от 0 до 31;
z = номер слова (группа из 8 сигналов - байт) в разъёме от 0 до 3.

8.12. Сообщения для оператора

Максимально 255 сообщений из 42 знаков каждое.

8.13. Условное присвоение или мультиплексор

MUX (СЛОВА), (Сигналы) выбирает слово, соответствующее первому активному сигналу.

8.14. Одновибраторы

Rxx = Ixxxuzz - импульс производной переднего фронта сигнала Ixxxuzz.

8.15. Арифметические операции в двоичном коде

[Wxxxuz + Wxxxuz] - сложение или вычитание 2-х или больше слов.

8.16. Преобразование числа в формате BCD в двоичный код

BIN (Wxxxxyz) - преобразует число в формате BCD, содержащееся в слове, в двоичный код.

8.17. Преобразование двоичного кода в формат BCD

BCD (Wxxxxyz) - преобразует двоичное значение слова в значение в формате BCD.

8.18. Модуль или абсолютное значение

[ABS (Wxxxxyz)] - представляет модуль числа или операции между скобками.

8.19. Знак числа

SGN (Wxxxxyz) - предоставляет бит знака числа или операции между скобками.

8.20. Компаратор ASCII (сообщение оператора)

«Текст» - сообщение, заданное оператором из 32 знаков максимально.

9. ПРОЦЕДУРЫ PLC. ДИРЕКТИВЫ

Директивы для отладки и отработки ПЛ задаются в режиме «КОМАНДА» с помощью опции меню «PLC» («F3»).

9.1. Ввод программы в память

Файл-источник ПЛ при вводе воспринимается системой как общий файл. Поэтому его ввод в память может быть выполнен оператором с заданием директивы с клавиатуры:

EDI, имя ПЛ /МРх.

В случае если файл уже существует в другом логическом носителе **МРх**, ввод его в память может быть выполнен посредством директивы:

СОР, Имя ПЛ1/МРх, Имя ПЛ2/МРу,

где:

Имя ПЛ1 - наименование существующего источника ПЛ;

МРх - логический носитель, на котором он находится;

Имя ПЛ2 - наименование нового источника ПЛ;

МРу - логический носитель, на котором он будет находиться.

Источник ПЛ можно набирать в любом текстовом редакторе и сохранять как файл **MSDOS** в текстовом формате. Имя должно состоять из 6 символов (первой должна быть буква) без расширения.

9.2. Директивы компиляции

Файл-источник ПЛ, загруженный на логический носитель предыдущей операцией, чтобы стать активным, должен быть переведён в объектную программу посредством директивы компиляции. Для этого необходимо вызвать программу **PLC**. Директива следующая:

RUN, PLC

Это же возможно сделать через соответствующую опцию меню режима «КОМАНДА» по клавише «F3»:

1 ДИСПЛ | 2 МОДИФ | 3 PLC | 4 КОПИЯ

После ввода этой директивы на дисплее появится «Среда», если ещё не имеется конфигурация параметров «Среды», или **МЕНЮ 1**, первое из списка меню. Для удобства идентификации каждое **МЕНЮ** в этом документе имеет порядковый номер: **МЕНЮ 1 - МЕНЮ 5**.

Пример

МЕНЮ 1

1 Компиляция | 2 Быстрая компиляция | 3 Отладчик | 4 Среда | 5 Выход

Для того чтобы выбрать одну из выше указанных возможных операций **МЕНЮ 1**, необходимо начать одну из функциональных клавиш **«F1»–«F5»**:

- **«F1»** – осуществляется компиляция источника ПЛ с возможными опциями (по умолчанию, отладка и т.д.), которые будут запрошены в ходе выполнения данной операции;
- **«F2»** – выполняется компиляция в соответствии с данными, записанными в **«Среде»**;
- **«F3»** – вызывает меню отладки;
- **«F4»** – обеспечивает вход в **«Среду» PLC** для характеристики параметров быстрой компиляции файлов-источников ПЛ;
- **«F5»** – осуществляет возврат в режим **«КОМАНДА»**.


Перед выполнением быстрой компиляции необходимо подготовить **«Среду» PLC** в соответствии с требованиями пользователя.

9.2.1. «Среда»


PLC обеспечивает определение источников ПЛ и резервирование части ОЗУ для загрузки объектного кода посредством задания параметров **«Среды»**. Параметры **«Среды»** запоминаются в невидимом файле **SIPCON/MPO** в момент выхода из **«Среды»**: по клавише **«ESC»** перейти в **МЕНЮ «функции среды»** и нажать клавишу **«F1» («Исправить»)**.

Данный файл должен присутствовать при активизации **PLC**, поскольку он содержит основные данные для компилятора и отладчика.

«Среда» PLC состоит из двух страниц. Топология изображения страниц **«Среды»** на экране представлена на рисунке 9.1.

Обращение ко второй странице выполняется клавишей . Для ввода или замены параметра **«Среды»** необходимо сначала позиционировать маркер на этом параметре, а затем нажать клавишу **«ENTER»**. При этом наименование этого параметра появится во второй строке дисплея. Ввод набранного значения выполняется нажатием клавиши **«ENTER»**. Движение маркера осуществляется клавишами **«СДВИГ НА СТРОКУ ВПЕРЕД»** или **«СДВИГ НА СТРОКУ НАЗАД»**.

```


* NC-100 КОМПИЛЯТОР/ОТЛАДЧИК * ver.X.X * опции среды
Ввод параметра:
↓ ↑ и ENTER - выбор,  - страница, ESC -выход

Полное имя прог.: JJJJJ /XXX| Опции компиляции : FFFFF

Имя источника 0: JJJJJJ/XXX| Имя источника 1: JJJJJJ/XXX
Имя источника 2: JJJJJJ/XXX| Имя источника 3: JJJJJJ/XXX
Имя источника 4: JJJJJJ/XXX| Имя источника 5: JJJJJJ/XXX
Имя источника 6: JJJJJJ/XXX| Имя источника 7: JJJJJJ/XXX
Имя источника 8: JJJJJJ/XXX| Имя источника 9: JJJJJJ/XXX
    
```

а) 1-ая страница «Среды» PLC

```

* NC-100 КОМПИЛЯТОР/ОТЛАДЧИК * ver.X.X * опции среды
Ввод параметра:
↓ ↑ и ENTER - выбор,  - страница, ESC -выход

Полное имя прог.: JJJJJ /XXX | Опции компиляции: FFFFF
t быстр. лог. (мкс) : | t медл.лог. (мс):
Адрес загрузки : | Адрес загр. отладки:
Адрес исполнения : | Адрес исполн. отлад.:
Объем прогр (Кб) : XXXXX | Програм.отлад. (Kb) : XXXXX
Адрес загр вх/вых : | Лог. адрес загрузки :
Адрес исполн. вх/вых : | Лог. адр. исполнения:
    
```

б) 2-ая страница «Среды» PLC

Рисунок 9.1 – Топология изображения страниц «Среды» PLC

9.2.1.1. Описание параметров, вводимых в «Среде»

Параметры, определяемые в «Среде»:

Полное имя прог.: наименование объектной программы (5 знаков макс., первая буква) и логического носителя, на котором она находится. Объектная программа будет загружена в ОЗУ и активизирована только при объявлении логического носителя под именем **/МЕМ**. Операция компиляции всегда должна выполняться при выключенном станке.

Опции компиляции: данный параметр устанавливает все возможные опции, которые будут автоматически приняты после задания команды «**Быстрая компиляция**».

Опции задаются 1-ой буквой их названия и могут быть следующие:

V-Дисплей – компиляция с визуализацией на дисплее программы-источника;

B-Бит – создание объектной программы в коде, обеспечивающем доступ к биту во время выполнения ПЛ, увеличивая ее скорость. Доступ к биту возможен только на физи-

ческом пакете «А», а не по соответствующему адресу в ОЗУ, где они размещаются.

- D-Отладчик** – отладка: при компиляции автоматически генерируется объектная программа «**Полное имя программы + D**» для отладки, которая упрощает аттестацию ПЛ, активизируемой посредством директив, описанных в соответствующем разделе;
- S-Сохранить** – сохранение предыдущей объектной программы. Без этой опции до начала новой компиляции стирается старая объектная программа, если уже имеется в памяти (с тем же наименованием); может оказаться целесообразным не сохранять ее, когда место в объявленном носителе недостаточно для размещения новой и старой объектных программ. С этой опцией старая объектная программа заменяется на новую программу только по окончании компиляции, если не появляются ошибки;
- E-Выход по ошибке** – выход по ошибке, если при компиляции выявляется ошибка, данная компиляция прерывается на строке с ошибкой;
- I-Прервать** – компилируемая с данной опцией ПЛ получается с запрещением прерывания системы во время ее выполнения. На самом деле, если в ходе выполнения какого-нибудь оператора программы происходит прерывание системы, немедленно прерывается оператор логики. Без этой опции для каждого оператора имеются 5 байтов для управления названным выше прерыванием.

Например, при задании **DBISVE** в параметре «**Опции компиляции**» активизируются все вышеуказанные опции компиляции.

Имя источника: данный параметр с записью типа «**Имя ПЛ/логический носитель**» указывает наименование и носитель одного из десяти компилируемых файлов-источников; программу-источник поэтому можно разбить на десять файлов и с помощью компиляции получить единую исполняемую скомпилированную программу. Такое разбиение удобно при создании ПЛ на серию одноплатных станков, отличающихся друг от друга в управлении одним или несколькими узлами, например, количеством позиций в револьверной головке и прочее.

t быстр.лог. (мкс): этот параметр определяет максимальное время для выполнения быстрой части ПЛ (текст ПЛ выше символа «\$»);

t медл.лог. (мс): этот параметр определяет максимальное время для выполнения медленной части ПЛ (текст ПЛ ниже символа «\$»);

Адрес загр., Адрес исполн., Адрес загр.отлад., Адрес исполн.отлад.: эти четыре адреса определяются автоматически программой PLC;

Объем progr. , Програм.отлад. - этими двумя параметрами определяются максимальная длина (КБ) объектной программы и объектной программы для отладки. При компиляции программа оценивается по двум критериям:

- 1) наличие свободной памяти в ОЗУ для размещения скомпилированной программы. Если памяти не хватает для размещения программы, на экране индицируется ошибка:

**«Неиспр. Ош. - нажмите ENTER
недостаточно места для файла
ОСТАНОВ КОМПИЛЯТОРА»;**

- 2) резервирование недостаточного объема памяти для скомпилированной программы. Если памяти отведено недостаточно, на экране индицируется ошибка:

«ОСТАНОВ КОМПИЛЯЦИИ - ПЕРЕПОЛНЕНИЕ КОДА»

Примечание - Максимальная длина объектной программы и объектной программы для отладки может быть установлена не более **FFFF (HEX)** и физически ограничена объемом ОЗУ, установленным в УЧПУ.

При переопределении объема программы логики станка для новой компиляции необходимо предварительно остановить активную логику станка в меню **«PLC»** -> опция **«Отладчик»** -> опция **«Стоп»**.

При не выполнении данной процедуры действий и попытке компилирования на экран будет выведена следующая информация:

**«Неиспр. Ош. - нажмите «ENTER»
Логика не остановлена
ОСТАНОВ КОМПИЛЯТОРА»**

Адрес загр вх/вых , Лог. адрес загрузки , Адрес исполн. вх/вых, Лог. адр. исполнения: эти 4 параметра представляют собой адреса памяти, по которым вводятся и читаются во время выполнения программы логики входы/выходы физического пакета **«А»** и логического пакета **«К»**. Данные адреса заполняются автоматически при компиляции программой **PLC**.

После заполнения второго кадра **«Среды»** надо нажать клавишу **«ESC»**, на дисплей будет выведено **МЕНЮ**:

Функции среды:

1 Исправить | 2 Прекратить | 3 Изменить

Для выбора требуемой команды достаточно нажать одну из клавиш **«F1»-«F3»**:

- **«F1»:** **«Исправить»** - запоминание информации, объявленной в **«Среде»**;
- **«F2»:** **«Прекратить»** - запоминание **«Среды»** без последних изменений;
- **«F3»:** **«Изменить»** - возврат в **«Среду»**

9.2.1.2. Значения параметров «Среды» для компилирования и выполнения отладки ПЛ

Алгоритм выполнения ПЛ изображен на рисунке 9.2.



Рисунок 9.2 – Алгоритм выполнения ПЛ

Значения параметров «Среды» для компилирования и выполнения отладки ПЛ:

- интервал времени для отработки «медленной» части ПЛ и периодичность её отработки задаются при конфигурации системы, в файле **IOCFIL** (параметр **CLO**). Рекомендуемыми значениями будут соответственно 10 мс и 2 мс;
- время, выделяемое для отработки «быстрой» части ПЛ, задаётся в параметре **t быстр.лог.** (мкс), записанном в области «Среда» **PLC**.

Вышеуказанные параметры надлежит задавать перед компилированием в области «Среда» **PLC**, поскольку они определяют длительность окончательной ПЛ, т.е. периодичность, с которой будут обновляться сигналы, обрабатываемые «медленной» и «быстрой» частями ПЛ.

Действительно, после компилирования **PLC** выдает результат компилирования.

Пример

При установке задаются:

параметр CLO в IOCFIL 10,2
 область PLC «t быстр.лог (мкс)» 1000
 область PLC «t медл.лог. (мс)» 10.

Результат компилирования:
быстрая задача 500 мкс,
медленная задача 3856 мкс

Из этого примера можно сделать вывод, что «медленная» логика будет иметь длительность, равную двум циклам по 10 мс каждый, т.е. 20 мс.

Кроме того, на основе полученного результата для интервала «быстрой» логики можно оптимизировать соответствующий параметр, задаваемый при характеристике, уменьшая вдвое время, резервируемое «быстрой» логике, ввиду его не использования (тарировка 500).

9.3. Компиляция с запросом действий

9.3.1. Процедура компиляции

После задания «F1» в **МЕНЮ 1** система автоматически переходит на конфигурацию:

Имя источника 0:
[Имя] [/устр.]

Следует задать наименование первой программы-источника и ее место размещения «/логическое устройство памяти».

После нажатия клавиши «ENTER» запрашиваются наименования и носители, других программ-источников, если такие имеются. Если следующее имя программы-источника не задается, а нажимается клавиша «ENTER», на дисплей выводится конфигурация **МЕНЮ 2**:

МЕНЮ 2

D-Отладчик | B-Бит | I-Прервать | S-Сохранить | V-Дисплей | E-Выход по ошибке

Примечание - Если данные опции компиляции не нужны, по клавише «ENTER» выбираются опции компиляции **по умолчанию**.

При задании начальной буквы (**D, B, I, S, V, E**) различных позиций вышеуказанного меню получается компиляция с выбранными опциями (см. описание опций в разделе п.п.9.2.1.1). При нажатии клавиши «ENTER» (после выбора опций) выводится следующая информация:

Полное имя progr.:
[Имя] [/устр.]

В данном случае следует объявлять имя генерированной объектной программы и логический носитель, куда надо ее вводить. Если параметр пропускается, будет принято по умолчанию наименование программы-источника. Наименование логического носителя **[/устр.]** рекомендуется установить: «MEM», при этом объектная программа будет скомпилирована и загружена в ОЗУ по адресу, определенному программой автоматически.

В конце компиляции результат её выводится на экран:

- количество строк программ;
- количество предупреждений;
- количество ошибок;
- время и объём памяти в байтах для каждой части ПЛ.

В верхней части дисплея будет выводиться начальная конфигурация
МЕНЮ 1:

МЕНЮ 1

1 Компиляция|2 Быстрая компиляция|3 Отладчик|4 Среда|5 Выход

9.4. Быстрая компиляция с опциями, объявленными в «Среде»

Процедура быстрой компиляции выполняется при задании «F2» в **МЕНЮ 1**, при этом автоматически выполняется компиляция в соответствии с данными, записанными в «Среде», т.е. генерируется объектный файл с учётом всех опций (отладка, выход с ошибкой и т.д.) и данных, предварительно присвоенных соответствующим параметрам «Среды». Если в «Среде» не определено имя программы-источника «Имя источника» и/или логический носитель, где она находится, то до начала компиляции запрашиваются эти отсутствующие данные.

Можно прерывать компиляцию при нажатии клавиши «ESC», а затем сбросить это прерывание или восстановить его, нажимая соответственно «N» или «Y». Активизация новой скомпилированной программы происходит сразу после процедуры компиляции.

9.5. Аттестация программы логики

9.5.1. Создание файла для отладки

Для аттестации ПЛ на станке следует получить изменяемую и исполнительную скомпилированную программу с возможностью визуализации и записи в реальном масштабе времени переменных, используемых ПЛ, а также включения и отключения строк ПЛ с тем, чтобы оптимально выполнять предусмотренные функции и циклы.

Система **PLC** предоставляет в распоряжение пользователя опцию, называемую «Отладчик», которую можно вызвать двумя способами:

- 1) записать «D» в «Среде», в параметре «опции компиляции»;
- 2) с помощью директивы «Компиляция», задать «D» в конфигурации **МЕНЮ 2**.

В результате этих операций генерируется объектный файл **ИМЯ.D**.

ВНИМАНИЕ! ПОСЛЕ КОМПИЛЯЦИИ НЕОБХОДИМО ВЕРНУТЬСЯ В СОСТОЯНИЕ ОТЛАДЧИК, ЧТОБЫ ЗАГРУЖАТЬ, ПОДКЛЮЧАТЬ И АКТИВИЗИРОВАТЬ ПОЛУЧЕННЫЙ ФАЙЛ.

9.5.2. Загрузка объектного файла для отладки

При задании «D» в конфигурации **МЕНЮ 2** читается:

МЕНЮ 3

1 Монитор | 2 Монитор | 3 Назнач. | 4 Пуск | 5 Стоп | 6 Загруз.
реал.врем. | графики | перемен. | | | програм

Нажатием «F6» запрашивается наименование файла:

Имя отладочной программы:

[Имя] [/устр.]

Затем необходимо задать наименование отладочной программы и логический носитель (**/устр.**), на котором он находится.

ВНИМАНИЕ! ЛОГИЧЕСКИЙ НОСИТЕЛЬ В ПАРАМЕТРЕ «ПОЛНОЕ ИМЯ ПРОГРАМ.» ВСЕГДА ДОЛЖЕН ИМЕТЬ ИМЯ **/МЕМ.**

Имя отладочной программы должно состоять максимально из пяти знаков и буквы «D» в конце имени. Если программа отладки получена при быстрой компиляции, то имя отладочной программы должно повторять имя установленное в «Среде» в параметре «**Полное имя прогр.**» с буквой «D» в конце имени. В этом случае при запросе ввода имени загружаемой программы отладки по клавише «F6» достаточно нажать клавишу «**ENTER**».

9.5.3. Подключение программы отладки

В конце загрузки появляется конфигурация **МЕНЮ 4**:

МЕНЮ 4

1Монитор | 2Монитор | 3Назнач. | 4Подкдю- | 5Откдю- | 6Пуск | 7Стоп | 8Загруз.
реал.врем | графики | перемен. | чить | чить | | | програм

При нажатии клавиши «F4» индицируется:

подключить :

строка 1 {, строка 2}

Вводом параметров «**строка 1**» и «**строка 2**» можно подключить (активизировать) любые строки файла.

Пример

1,7 - с нажатием клавиши «**ENTER**»; подключаются уравнения с первого по седьмое.

Для активизации всех строк файла надо задать «*».

9.5.4. Выполнение объектного файла для отладки

После подключения файла снова появляется конфигурация **МЕНЮ 4**. При задании «**F6**» начинается выполнение объектного файла **ИМЯD**, и снова появляется конфигурация **МЕНЮ 4** с визуализацией:

Программа: имяD/устр.⏏

Символ ⏏ определяет активизацию (запуск на отработку) отладочной программы.

9.5.5. Отключение и разъединение объектной программы для отладки

Можно отключать файл или разъединять определенные строки программы, задавая соответственно команды «**Стоп**» и «**Отключить**» (конфигурация **МЕНЮ 4**).

9.5.6. Цифровая и графическая визуализация переменных в состоянии отладки

Визуализация переменных осуществляется одновременно с выполнением объектного файла **ИМЯD** с конфигурацией **МЕНЮ 3** или объектного файла для отладки **ИМЯD** в конфигурации **МЕНЮ 4**.

9.5.6.1. Визуализация цифрового значения переменной

Для визуализации в реальном масштабе времени цифрового значения переменных (16 максимум), таких как разъемы, слова, биты, необходимо задать «**F1**».

После чего на дисплее появляется **МЕНЮ 5**:

МЕНЮ 5

функции монитора реал. вр.:

1 Добавить | 2 Удалить | 3 Пуск | 4 Стоп

Значения клавиш **МЕНЮ 5**:

«**F1**» – запрос визуализации переменной в реальном масштабе времени;

«**F2**» – запрос стирания переменной;

«**F3**» – включение программы-монитора для визуализации переменных;

«**F4**» – остановка программы-монитора.

При нажатии клавиши «**F1**» **МЕНЮ 5** на дисплее появляется:

Добавить переменную:

Имя

При этом надо ввести наименование переменной и нажать клавишу **«ENTER»**. После выполнения этих действий появляется **МЕНЮ 5**:

сигнал: «имя сигнала» = значение

Эту операцию можно повторять максимально для 16 переменных.

Для индикации данных переменных в реальном масштабе времени необходимо включать программу-монитор командой **«F3» («ПУСК»)**. При необходимости ее отключения, следует задать **«F4» («СТОП»)**.

При вводе более 16 переменных на дисплее сразу появляется сообщение: **«Окно монитора заполнено»**. При этом необходимо стирать переменные, которые для наблюдения больше не нужны. Для этого надо задать в конфигурацию **МЕНЮ 5** опцию **«Удалить»**, при этом на дисплее воспроизводится:

Ввод параметра:

Вначале маркер (определяемый реверсивным изображением строки на дисплее) позиционирован на первой переменной; нажатием клавиши **«ENTER»** стирается переменная, на которой установлен маркер. Нажатием клавиш **«СДВИГ НА СТРОКУ ВПЕРЕД»** или **«СДВИГ НА СТРОКУ НАЗАД»** маркер перемещают на переменную, которую необходимо стереть.

9.5.6.2. Графическая визуализация переменных

В конфигурации **МЕНЮ 3** или **МЕНЮ 4** можно по клавише **«F2»** войти в режим графической визуализации.

функции монитора графики:

1 Добавить | 2 Удалить | 3 Пуск

Для выполнения режима визуализации необходимо:

- 1) нажать **«F1»** для введения переменных, при этом на дисплее появляется:

Добавить переменную:

Имя

- 2) ввести наименование переменной, которую требуется визуализировать, и нажать клавишу **«ENTER»**;
- 3) повторить операции 1) и 2) при необходимости ввода других переменных;
- 4) задать **«F3»**, при этом на дисплее появится:

базовое время/задержка:

Параметр **«базовое время»** определяет интервал времени, т.е. частоту, при которой переменная должна выводиться на дисплей.

Параметр «**задержка**» представляет собой время задержки после начала «**базового времени**» (значение, выраженное в единицах базового времени).

Пример

При задании 2/3, частота визуализации сигналов будет составлять $2 \cdot x$, и пауза будет составлять $3 \cdot x$, где x – значение **ТИКа** из инструкции **ТИМ** (секция 1 файла **АХСFIL**).

5) нажать клавишу «**ENTER**». При этом на дисплее появится:

сигнал триггера:

{имя { ↑ | ↓ }}

Здесь необходимо задать наименование сигнала запуска синхронизации для визуализируемых переменных. При изменении состояния этого сигнала начинается графическая визуализация. Нажатие клавиш «**СДВИГ НА СТРОКУ ВПЕРЕД**» и «**СДВИГ НА СТРОКУ НАЗАД**» указывают на фронт срабатывания сигнала запуска (передний или задний соответственно).

6) нажать клавишу «**ENTER**».

После выполнения вышеуказанных действий осуществляется графическая визуализация переменной.

9.5.7. Присвоение значений переменным

В конфигурации **МЕНЮ 3** или **МЕНЮ 4** можно присваивать требуемое значение переменным выхода по клавише «**F3**», при этом на дисплее появится:

назначить:
переменная = значение

Необходимо задать наименование переменной и присваиваемое ей значение в десятичном коде.

После нажатия клавиши «**ENTER**» в зависимости от того, загружена или нет объектная программа для отладки, на дисплее снова появляется конфигурация **МЕНЮ 3** или **МЕНЮ 4**, при этом на дисплее можно наблюдать:

байт: W00T0 = 0000 1010 (010 D).

Этот способ присвоения значения переменной имеет высший приоритет возможного присвоения тому же слову в ПЛ.

9.6. Перечень директив, используемых PLC

Перечень всех директив, используемых в **PLC**, сведён в таблицу 9.1.

Таблица 9.1 - Перечень директив, используемых в PLC

ДИРЕКТИВА	ФУНКЦИИ ДИРЕКТИВЫ
Компиляция	Компиляция с опциями
Быстрая компиляция	Быстрая компиляция с опциями, определенными в «Среде»
Отладчик	Отладка программы логики
Среда	Среда для характеристики параметров PLC
Выход	Выход из PLC
/устр	Объявление логического носителя файла
Бит	Генерация программы с доступом к биту
Сохранить	Не сохраняется старая скомпилированная программа
Дисплей	Визуализация скомпилированной программы
Прервать	Запрещение прерываний
Выход по ошибке	Компиляция с выходом по ошибке
Монитор реал. врем.	Визуализация значения переменной
Монитор графики	Графическая визуализация переменной
Назначить	Присвоение значений переменным
Загруз. програм	Загрузка объектного файла для отладки
Подключить	Подключение объектного файла для отладки
RUN	Вызов программы-монитора или файла
Отключить	Отключение объектного файла для отладки
Стоп	Останов программы-монитора или файла
Добавить	Загрузка визуализируемых переменных
Удалить	Стирание переменных
Исправить	Обновление среды PLC «СРЕДА»
Прекратить	Не обновляется среда PLC
Изменить	Возврат в среду PLC

10. ДИАГНОСТИКА ПРИ РАБОТЕ С PLC

Диагностика **PLC** выявляет ошибки, возникающие по различным причинам, сообщения о которых выдаются на дисплей.

10.1. Ошибки языка PLC

Ошибки использования языка **PLC** представлены в таблице 10.1. Эти сообщения, написанные строчными буквами, относятся к ошибкам пользователя.

Таблица 10.1 – Ошибки использования языка PLC

СООБЩЕНИЯ	ПРИЧИНА ОШИБКИ	ИСТОЧНИКИ
Ошибка формата	Неверный формат	К О
Неправильная операция	Неверная инструкция	К О
Файл не существует имя/устр.	Вызванный файл не существует в объявленном носителе	К О
Неправильное имя	Слишком длинное имя файла или первый знак – не буква	О
Неиспр. Ош. – нажми <ENTER> Ошибка вв/выв – 1	На выходе из состояния «Среда» нельзя запомнить файл SIPCON . Надо инициализировать память PLC и повторить операцию. Проверить наличие MPO	О
Неиспр. Ош. – нажми <ENTER> Недостаточно места для файла «имя файла»/MEM	В верхней памяти (UMB) недостаточно места для размещения объектного файла	О
Неиспр. Ош. – нажми <ENTER> Логика не остановлена	Перед началом компиляции в отладчике PLC не остановлена логика клавишей «F5» («СТОП»)	О
Примечание – «О» – ошибки отладчика, «К» – ошибки компилятора		

10.2. Ошибки программы, выявляемые при компиляции

При выявлении компилятором ошибки на дисплей выводится:

***ошибка компиляции * nnn**

где:

nnn – код ошибки.

Ошибки компиляции сведены в таблицу 10.2.

Ошибки, отмеченные символом «*», не останавливают компиляцию. Вышеназванные ошибки не останавливают компиляцию; для продолжения компиляции (если не запрашивается опция) необходимо нажать букву «Y», а для ее прекращения – букву «N».

Таблица 10.2 - Ошибки программы, выявляемые при компиляции

КОД	ПРИЧИНА ОШИБКИ
1	Ошибка в определении переменной
2	Синтаксическая ошибка
3	Неподходящий операнд
4	Ошибка в идентификаторе <MSG>
5	Не хватает номера сообщения
6	Неправильный номер сообщения
7	Слишком длинное сообщение
8	Сообщение уже определено
9	Ошибка в идентификаторе <MUX>
10	Неизвестный разделительный знак
11	< () Запрошено перед параметром MUX
12	<) > Запрошено после параметров MUX
13	Номер слова отличается от номера сигналов
14	Неизвестные элементы в конце MUX
15	Слишком сложный оператор
16	Слишком много уровней скобок
17	Ошибка в идентификаторе <SGN>
18	Ошибка в идентификаторе <ABS>
19	Неизвестный тестовый символ
20	<] > запрошено в конце теста
21	Нулевой компаратор ASCII
22	Слишком длинный компаратор ASCII
23	< " > Запрошено в конце компаратора ASCII
24	Ошибка в идентификаторе <DEC>
25	Ошибка в идентификаторе <ENC>
26	Ошибка в идентификаторе <BIN>
27	Ошибка в идентификаторе <BCD>
28	Ошибка в идентификаторе <LOW>
29	Ошибка в идентификаторе <HIG>
30	Ошибка в идентификаторе <XCH>
31	<) > запрошено в конце функции DEC
32	<) > запрошено в конце функции ENC
33	<) > запрошено в конце функции BIN
34	<) > запрошено в конце функции BCD
35	<) > запрошено в конце функции LOW
36	<) > запрошено в конце функции HIG
37	<) > запрошено в конце функции XCH
39	Превышение максимального времени исполнения задачи медленной логики (указанного в «СРЕДА»)
40	Слишком большая вложенность в операторах в условных переходах
41	Неверное имя оператора условного перехода
42	Несоответствие типа DO/END в операторах условных переходов
43	Отсутствие END в операторе условного перехода
44	Отсутствие DO в операторе условного перехода
45	Оператор DOE без предшествующего DOF
46	Синтаксическая ошибка в символьном имени оператора
47	Вторичное объявление символьного имени оператора
48	Переполнение таблицы символьных имен операторов
49	Символьный оператор не найден

10.3. Ошибки компиляции

Ошибки компиляции сведены в таблицу 10.3. Ошибки, указанные в таблице 10.3, прекращают компиляцию.

Таблица 10.3 - Ошибки компиляции

ОШИБКА	ПРИЧИНА
50	Переполнение таймера
51	Переполнение длины программы-источника
52	Слишком много строк-источников для отладчика
60	Ошибка ВХ/ВЫХ в записи объектной программы
62	Ошибка ВХ/ВЫХ в считывании программы-источника
63	Прекращение запроса от пользователя
ОСТАНОВ КОМПИЛЯЦИИ - ПЕРЕПОЛНЕНИЕ КОДА	Слишком большая программа-источник, либо отведено недостаточно места в «Среде» в параметрах «Объем прогр.» и «Прогр. отлад» эти параметры вводятся 16-ми числами.
ОСТАНОВ КОМПИЛЯЦИИ - Логика не остановлена	После изменения значений в параметрах «Объем прогр.» и «Прогр. отлад» Среда не была остановлена ПЛ.
ОСТАНОВ КОМПИЛЯЦИИ - недостаточно места для файла	Недостаточно места в ОЗУ для размещения скомпилированной ПЛ с объемом, который задан в Среде в параметрах «Объем прогр.» и «Прогр. отлад».

10.4. Ошибки отладчика

Ошибки отладчика представлены в таблице 10.4.

Таблица 10.4 - Ошибки отладчика

СООБЩЕНИЯ	УКАЗАНИЯ ПО ОШИБКАМ
Неправильная переменная	Уточнить переменную
Неправильное значение	Уточнить значение
Строки вне диапазона	Уточнить диапазон

11. ГЕНЕРАЦИЯ ФАЙЛА ОБЪЕКТНОЙ ПРОГРАММЫ ЛОГИКИ СТАНКА

Порядок выполнения действий для генерации файла объектной программы логики станка:

- 1) записать символ «;» перед инструкцией **ALM** (секция 1 файла IOCFIL) : **;ALM = 0**
- 2) выключить, а затем снова включить УЧПУ;
- 3) нажать клавишу «**F4**» для перехода в окно «**Среда**»;
- 4) ввести значения параметров «**Среды**»:

Полное имя программы: **Имя** (не более пяти знаков) / **МЕМ**

Опции компиляции: **по умолч.** (для чего нажмите клавишу «**ENTER**»)

Имя источника **0-9**: **имя ПЛ/устройство памяти размещения ПЛ**;

- 5) нажать клавишу для перехода на 2 страницу «**Среды**» и ввести значения параметров «**Среды**»:

- t быстр.лог. (мкс)	00000
- t медл.лог. (мс)	0000
- Объём програм. (КБ)	(max. FFFF)
- Програм.отлад. (КБ)	(max. FFFF)

- 6) нажать клавишу «**ESC**», а затем клавишу «**F1**» («**ИСПРАВИТЬ**»);
- 7) нажать клавишу «**F2**» («**БЫСТРАЯ КОМПИЛЯЦИЯ**»), после чего контролировать результат выполнения компиляции по сообщению: «**0000 ошибок**»;
- 8) нажать клавишу «**F5**» («**ВЫХОД**») для выхода из **PLC**;
- 9) удалить символ «;» перед инструкцией **ALM** (1 секция файла IOCFIL) : **ALM = 0;**
- 10) выключить, а затем снова включить УЧПУ.

Контролировать активизацию ПЛ при загрузке УЧПУ.

При необходимости удалить исходную программу логики из памяти УЧПУ (с FLASH), необходимо открыть ее в редакторе и удалить текст, оставив в программе один символ, например: «;». Записать файл и выйти из редактора. Параметры «**Среды**» в PLC необходимо оставить без изменения.

12. ИНТЕРФЕЙС PLC

12.1. Общие сведения

Основные задачи интерфейса PLC:

- 1) инициализация сигналов для включения управляемого оборудования;
- 2) обработка сигналов протокола, который определяет выполнение различных режимов работы УЧПУ;
- 3) обеспечение работы устройств безопасности станка;
- 4) выполнение вспомогательных функций **S**, **T**, **M**, индексной оси, оси «от точки к точке»;
- 5) выполнение протоколов:

БАЗОВОЕ ПрО ↔ ПЛ ↔ УПРАВЛЯЕМОЕ ОБОРУДОВАНИЕ

В данном документе представлено описание алгоритмов всех задач, реализуемых программным интерфейсом, а также перечень и описание назначения сигналов интерфейса с рекомендациями по их использованию.

В документе можно встретить описание сигналов, слов и операций, отмеченных кодом **ECDF**. Применять эти сигналы, слова и операции допускается в двух случаях:

- 1) в версиях ПрО УЧПУ с порядковым номером 74 и выше (например: Z.74P), если во втором параметре инструкции **NBP** записан код **ECDF**;
- 2) в версиях ПрО УЧПУ, в обозначении расширения которых присутствует буквенный код «РИВ», например: Z.33РИВ.

Примечание – Кодирование версий ПрО УЧПУ указано в документе «Руководство по характеристикам».

Блок – схемы алгоритмов взаимодействия программного интерфейса **PLC** с ПЛ приведены в приложении **A**.

Интерфейс **PLC** обеспечивает взаимодействие базового ПрО УЧПУ с ПЛ, причём ПЛ является персональной для каждого объекта управления. Протокол связи реализован посредством сигналов логических пакетов «**K**» и «**N**». Условно, пакеты разделены на разъёмы. Общее число разъёмов составляет **256** для пакета «**K**» (**OK-255K**) и **256** для пакета «**N**» (**ON-255N**). Определённая часть разъёмов, сигналы которых представлены в таблицах 24.1, 24.2, 25.1-25.5, представляет собой базовые сигналы, остальные разъёмы – это память пользователя.

Выполнение запросов от базового ПрО к ПЛ осуществляется посредством базовых сигналов следующих разъёмов пакета «**K**»:

Процесс 1	разъёмы: 0K	-	9K
Процесс 2	разъёмы: 26K	-	35K
Процесс 3	разъёмы: 52K	-	61K
Процесс 4	разъёмы: 78K	-	87K
Процесс 5	разъёмы: 104K	-	113K

Выполнение запросов от ПЛ к базовому ПрО осуществляется посредством базовых сигналов следующих разъёмов пакета «К»:

Процесс 1	разъёмы: 10К - 22К
Процесс 2	разъёмы: 36К - 51К
Процесс 3	разъёмы: 62К - 77К
Процесс 4	разъёмы: 88К - 103К
Процесс 5	разъёмы: 114К - 129К

Кроме сигналов пакета «К» имеются еще 512 внутренних сигналов, зарезервированных для пользователя для записи информации, сохраняющейся при отключении УЧПУ. Это сигналы пакета «Т» (**0Т-15Т**). Всего 64 записи. Данные сигналы входят в состав секции 4 файла **IOCFIL**. Каждая запись имеет следующий формат:

N записи = ЗНАЧЕНИЕ,

где:

N записи - номер записи от 1 до 64;

ЗНАЧЕНИЕ - выражается в шестнадцатеричном коде от 0 до FF.

Пример

```

T01 = (W00T0)
T02 = (W00T1)           разъём 0Т
T03 = (W00T2)
T04 = (W00T3)
.
.
.
T61 = (W15T0)
T62 = (W15T1)           разъём 15Т
T63 = (W15T2)
T64 = (W15T3)
    
```

Для правильного использования пакета «Т» необходимо помнить:

- 1) если какие-то слова пакета «Т» используются в ПЛ в качестве таймеров или счётчиков, в последующем их нельзя использовать для другого назначения;
- 2) задание параметров счётчиков и таймеров неявным способом производится двумя соседними словами; например, **T05I(W00T0)** - время для таймера определено в словах: **W00T0** и **W00T1**. Слово **W00T1** не должно использоваться и изменяться из ПЛ для других целей, даже если оно равно нулю. Изменение слова **W00T1** из ПЛ приведёт к изменению времени работы таймера **T05I**. То же относится и к заданию параметров счётчиков и таймеров через слова пакета «К»;
- 3) значения пакета «Т» сохраняются в случае, если УЧПУ загружено с уже установленной инструкцией **OLD** в файле **PGCFIL**, и перед выключением УЧПУ сигнал **U10K0 (MUSP)** равен «1» (см. «Руководство по характеристикам»).

13. ИНИЦИАЛИЗАЦИЯ УЧПУ

Под инициализацией понимается выполнение определённых операций, служащих для приведения УЧПУ в предпусковое состояние (состояние **STAND-BY**). Процедура инициализации УЧПУ может быть произведена различными способами:

- 1) процедурой **ВКЛЮЧЕНИЕ**;
- 2) включением станка после аварии;
- 3) процедурой **«СБРОС» («RESET»)**.

13.1. Процедура ВКЛЮЧЕНИЕ

Эта процедура инициализации начинается при повороте ключа включения УЧПУ. В этом случае УЧПУ после фазы сброса и самодиагностики, если не обнаруживаются какие-либо аномалии в работе (ошибки памяти, входов/выходов и т.д.), устанавливает в «1» сигнал **RSPEPN (I06K21)**, выдаваемый в базовый пакет «К». Для включения управляющего напряжения на станке через контакты реле «**SPEPN**» сигнал **U10K20 (ASPEPN)** необходимо установить в «1» (для выключения - в «0»). При включении цепи управляющего напряжения на станке, когда сигнал **MUSPE** будет равным «0», в пакет «К» выдается сигнал **CONP («Готовность УЧПУ»)**.

Примечание - Реле «**SPEPN**» (реле готовности УЧПУ к включению станка) имеет нормально-разомкнутые контакты, которые должны быть задействованы в цепи аварийного отключения станка.

13.2. Включение станка после аварии

Под аварией понимается какая-либо особая, экстренная ситуация, сбой привода, выключение станка и т.п. и установка сигнала **MUSPE** в «1». После аварии система ожидает перехода в «0» сигнала **MUSPE**, после чего она сбрасывает диагностическое сообщение об аварии и устанавливает в «1» сигнал **CONP («Готовность УЧПУ»)**.

13.3. Процедура «СБРОС» («RESET»)

При процедуре инициализации «**СБРОС**» прекращаются текущие операции, и осуществляется переход УЧПУ в состояние **STAND-BY**. Это происходит следующим образом:

- 1) останавливаются перемещения по осям;
- 2) аннулируются все предварительно запрограммированные вспомогательные функции.

PLC сообщает в пакет «К» о состоянии «**СБРОС**» посредством сигнала **RESET (I00K1)** импульсного типа, который длится два цикла «медленной» части ПЛ.

14. РЕЖИМЫ РАБОТЫ УЧПУ

УЧПУ может работать в различных режимах. Режим, как правило, выбирается самим оператором переключателем режимов или клавишами на станочном пульте или же из ПЛ посредством соответствующего запроса.

Интерфейс PLC выдаёт в пакет «К» набор сигналов в соответствии с определённым протоколом, однозначно определяющим выбранный режим.

Имеются следующие режимы:

- 1) отсоединение осей;
- 2) исключение контроля управления приводом;
- 3) переключение осей;
- 4) ручные перемещения («**БЕЗРАЗМЕРНЫЕ РУЧНЫЕ ПЕРЕМЕЩЕНИЯ**», «**ФИКСИРОВАННЫЕ РУЧНЫЕ ПЕРЕМЕЩЕНИЯ**», «**ВОЗВРАТ НА ПРОФИЛЬ**», «**ВЫХОД В НОЛЬ**»);
- 5) автоматический режим («**АВТОМАТИЧЕСКИЙ**», «**КАДР**», «**РУЧНОЙ ВВОД КАДРА**»).

14.1. Отсоединение осей

Этот режим позволяет визуализировать координаты, соответствующие вычисленной позиции осей, но без отработки перемещений по осям.

В этом режиме оси обслуживаются в той позиции, в которой они находятся, и контролируются только на сбой привода. Режим вызывается посредством команды: **UAS=1** (**UAS=0** – подсоединение осей).

14.2. Исключение контроля управления приводом

Этот режим позволяет управлять осями и визуализировать координаты осей, соответствующие перемещению, но без обычного контроля по скорости и положению.

Этот режим задаётся установкой в «0» тех разрядов (бит) слова **RABn (W10K1)**, которые соответствуют осям, по которым исключён контроль управления приводом.

14.3. Переключение осей

В УЧПУ имеется возможность управления двумя взаимно переключаемыми осями (их имена объявляются при конфигурации) посредством одного цифрового аналогового канала.

Для этого в слове **RCOMn (W17K1)** устанавливается в «1» тот бит, который соответствует временно «отключаемой» оси (система выполняет только контроль сбоя привода неподвижной оси) в пользу той оси, которой требуется управлять. После выполнения коммутации осей система передаёт в пакет «К» посредством слова **W06K0** информацию об отключённой оси.

14.4. Ручные перемещения

Этот режим задаётся установкой переключателя режимов станочного пульта в одно из положений: **«БЕЗРАЗМЕРНЫЕ РУЧНЫЕ ПЕРЕМЕЩЕНИЯ»** (**«MANU»**), **«ФИКСИРОВАННЫЕ РУЧНЫЕ ПЕРЕМЕЩЕНИЯ»** (**«MANJ»**), **«ВОЗВРАТ НА ПРОФИЛЬ»** (**«PROF»**), **«ВЫХОД В НОЛЬ»** (**«НОМЕ»**). В этих режимах единственными разрешёнными операциями являются перемещения по осям, запрашиваемые оператором с пульта, или же от ПЛ посредством определённого запроса. Перемещения выполняются поочередно для каждой оси. Кроме того, может выполняться поиск микронуля для абсолютного (нулевого) отсчётного значения оси и возврат на профиль.

В зависимости от выбранного оператором ручного режима система информирует ПЛ об установленном режиме посредством сигналов: **MANUC (I08K27)**, **MANUJ (I08K28)**, **RIPRO (I08K29)**, **RIMZE (I08K30)**.

В тот момент, когда запрашивается перемещение оси, интерфейс устанавливает в «1» сигнал, соответствующий выбранной оси **MOV1 (I00K16)** – для оси 1, **MOV2 (I00K17)** – для оси 2, **MOV3 (I00K18)** – для оси 3 и т.д.). Перемещение выполняется сразу же при получении сигнала разрешения перемещения **COMU=1 (U10K24)**.

Скорость и направление перемещения оси задаётся посредством корректора ручной подачи **«JOG»**. После окончания перемещения сигнал сбрасывается. УЧПУ информирует ПЛ о направлении перемещения выбранной оси посредством сигнала **DIRMN (I00K31)** (**DIRMN=1** означает отрицательное перемещение).

14.5. Автоматический режим

Режим задаётся установкой переключателя режимов станочного пульта в одно из положений **«АВТОМАТИЧЕСКИЙ»** (**«АУТО»**), **«КАДР»** (**«STEP»**) или **«РУЧНОЙ ВВОД КАДРА»** (**«MDI»**). В каждом из этих режимов УЧПУ выполняет, в принципе, однотипную работу. Работа УЧПУ состоит в обработке данных и отработке кадра программы (вспомогательные функции, перемещение по осям).

В зависимости от режима изменяется только тип отработки кадра программы:

- 1) в режиме **«АВТОМАТИЧЕСКИЙ»** система автоматически обрабатывает технологическую программу кадр за кадром;
- 2) в режиме **«КАДР»** система обрабатывает программу по кадру, после чего переходит в состояние **STAND-BY**, и только после нажатия оператором клавиши **«ПУСК»** начинает отработку очередного кадра программы;
- 3) в режиме **«РУЧНОЙ ВВОД КАДРА»** при нажатии клавиши **«ПУСК»** система начинает отработку кадра программы, предварительно введенного с клавиатуры УЧПУ.

Отработка кадра состоит из следующих фаз:

- устанавливается в «1» сигнал **AUTO (I08K25)**, если режим – **«АВТОМАТИЧЕСКИЙ»** (**«АУТО»**); сигнал **SEMI (I08K26)** – если

режим «КАДР» («STEP»); сигнал **EMDI (I08K24)** – если режим «РУЧНОЙ ВВОД КАДРА» («MDI»);

- устанавливается в «1» сигнал «Признак отработки кадра» **CYCLE (I00K3)**;
- исполняются вспомогательные функции начала перемещения «S», «T», «M», индексная ось. Обработка функций производится в указанном порядке, если на фазе конфигурации (файл **IOCFIL**) была задана последовательная обработка. При задании параллельной обработки вспомогательные функции «S», «T», «M» выдаются одновременно, а индексная ось – в течение двух последовательных периодов ПЛ;
- сбрасывается сигнал состояния **STAND-BY** (сигнал **STABY (I00K4)**), затем устанавливаются сигналы: «Признак кадра G00» (**FG00 (I05K0)**) и «Признак кадра стандартного цикла» в соответствии с характеристиками обрабатываемого кадра;
- выполняется, если задано, перемещение по осям (с их переключением, если необходимо);
- выполняются функции «M» немедленного действия;
- сбрасывается сигнал «CYCLE»;
- выполняются вспомогательные функции конца перемещения «M»;
- если имеется смена коррекций, то выполняется обновление таблицы инструмента и корректоров;
- устанавливается в «1» сигнал **STABY**, чем и заканчивается обработка.

15. БЛОКИРОВКИ БЕЗОПАСНОСТИ СТАНКА

В УЧПУ предусмотрены следующие блокировки:

- 1) ограничения перемещений;
- 2) ограничения перемещений (ECDF);
- 3) объёмная защищённая зона «куб» (ECDF);
- 4) программно-контролируемые зоны (ECDF);
- 5) контроль скорости осей (ECDF);
- 6) авария;
- 7) приостанов;
- 8) останов перемещения.

15.1. Ограничения перемещений

УЧПУ обеспечивает контроль за тем, чтобы все оси находились в пределах безопасных зон. При характеристике системы (файл **AXCFIL**) устанавливается способ управления конечными выключателями: аппаратный и/или программный.

При аппаратном способе управления конечными выключателями для каждой оси предусматривается два сигнала (положительный и отрицательный) конечных выключателей.

При программном способе - два предела (положительный и отрицательный) для каждой координаты, вводимые на фазе конфигурации системы от позиции начала отсчёта оси.

При программном способе управления контроль за ограничением перемещения действует только после выхода в «0» для данной оси.

Следует отметить, что один и тот же станок может иметь как аппаратный, так и программный контроль ограничения перемещения. В случае если УЧПУ обнаруживает, что при перемещении какая-либо ось вышла в зону ограничения перемещения, УЧПУ обнуляет задания на перемещение для всех осей и визуализирует сообщение об ошибке на дисплее. Для выхода из зоны ограничения перемещения необходимо в режиме ручных перемещений выбрать требуемую ось и выполнить перемещения оси в таком направлении, чтобы вернуть её в рабочую зону. При этом если направление перемещения выбрано неправильно, УЧПУ прекращает перемещение, не заканчивая его.

15.2. Ограничения перемещений (ECDF)

При активизации функций расширения возможно использование двух зон оперативных пределов. Выбор одной из 2-х зон оперативных пределов выполняется в любой момент работы установкой сигнала **LOX (U10N1)**.

Зоны ограничения оперативных пределов, определённые в инструкции **LOn**, действительны в режимах «**АВТОМАТИЧЕСКИЙ**» («**AUTO**»), «**РУЧНОЙ ВВОД КАДРА**» («**MDI**») и «**БЕЗРАЗМЕРНЫЕ РУЧНЫЕ ПЕРЕМЕЩЕНИЯ**» («**MANU**») и «**ФИКСИРОВАННЫЕ РУЧНЫЕ ПЕРЕМЕЩЕНИЯ**» («**MANJ**»). Так как подготовка кадра для выполнения и ПЛ не синхронизированы между собой, то для более оперативной реакции проверка на вход в активную зону **LOn** проверяется каждый цикл привода, указанный в инструкции **CAS** (секция 1 файла **AXCFIL**).

В режиме ручных перемещений сообщение «ВНЕ ОПЕРАТИВНЫХ ПРЕДЕЛОВ» (**Сообщение_4 03**) и запрет на дальнейшее движение происходят непосредственно при въезде в активную на данный момент зону **LOn**.

В режимах «**РУЧНОЙ ВВОД КАДРА**» или «**АВТОМАТИЧЕСКИЙ**» сообщение «ВНЕ ОПЕРАТИВНЫХ ПРЕДЕЛОВ» (**Сообщение_4 03**) и запрет на дальнейшее движение происходят при анализе обрабатываемого кадра перед перемещением, заданным в нём, в том случае, если окончание заданного перемещения будет в активной на данный момент зоне **LOn**.

Примечание - Если переключение зон **LO1** и **LO2** происходит в момент перемещения, и при этом положение оси оказывается внутри новой активной зоны, то перемещение будет немедленно прервано без контролируемого замедления, и на экране появится сообщение: «Вход на ограничение перемещения:» (**Сообщение_4 04**). При этом могут быть превышены пределы, установленные в инструкции **SRV**.

15.3. Объёмная защищённая зона «куб» (**ECDF**)

При активизации функций расширения возможно использование объёмной защищённой зоны. Объёмная зона позволяет определить часть пространства, где три оси не могут находиться одновременно.

Объёмная защищённая зона действует в режимах «**АВТОМАТИЧЕСКИЙ**» («**AUTO**»), «**РУЧНОЙ ВВОД КАДРА**» («**MDI**»), «**БЕЗРАЗМЕРНЫЕ РУЧНЫЕ ПЕРЕМЕЩЕНИЯ**» («**MANU**») и «**ФИКСИРОВАННЫЕ РУЧНЫЕ ПЕРЕМЕЩЕНИЯ**» («**MANJ**»). Так как подготовка кадра для выполнения и ПЛ не синхронизированы между собой, то для более оперативной реакции проверка на вход в объёмную защищённую зону проверяется каждый цикл привода, указанный в инструкции **CAS** (секция 1 файла **AXCFIL**).

В режимах «**БЕЗРАЗМЕРНЫЕ РУЧНЫЕ ПЕРЕМЕЩЕНИЯ**» («**MANU**») и «**ФИКСИРОВАННЫЕ РУЧНЫЕ ПЕРЕМЕЩЕНИЯ**» («**MANJ**») сообщение «ВНЕ ОПЕРАТИВНЫХ ПРЕДЕЛОВ» (**Сообщение_4 03**) и запрет на дальнейшее движение в данном направлении происходят непосредственно при въезде в активизированную объёмную защищённую зону.

В режимах «**РУЧНОЙ ВВОД КАДРА**» или «**АВТОМАТИЧЕСКИЙ**» сообщение «ВНЕ ОПЕРАТИВНЫХ ПРЕДЕЛОВ» (**Сообщение_4 03**) и запрет на дальнейшее движение происходят при анализе обрабатываемого кадра перед перемещением, заданным в нём, в том случае, если окончание заданного перемещения будет в активной на данный момент зоне.

Примечания

1. Вход в защищённую зону **CUB**, и сообщение об этом происходит только в тот момент времени, когда в эту зону входят все три оси, имеющие инструкцию **CUB**, при **U10N0 (CUB)**, равным «1». Таким образом, если только одна или две оси из 3-х, объявленных в инструкции **CAX**, находятся в защищённой зоне, то этого недостаточно для определения входа в объёмную защищённую зону и останковки движения осей.
2. Если активизация зоны происходит в момент перемещения, и при этом положение всех трёх осей оказывается внутри этой зоны, то перемещение будет немедленно прервано без контролируемого замедления, и на экран выводится сообщение: «Вход на ограничение перемещения:» (**Сообщение_4 04**). В этом случае могут быть превышены пределы, установленные в инструкции **SRV**. Во избежание отключения станка правильно сформируйте значение **области** в третьем параметре инструкции **CUB** секция 2 файла **AXCFIL**.
3. Контроль входа осей в защищённую зону происходит каждый тик привода, определённый в инструкции **CAS** (секция 1 файла **AXCFIL**).
4. Объёмная защищённая зона и запрещённая зона, определённая в трёхбуквенном коде **ASC**, могут быть активны и контролироваться одновременно.

15.4. Программно-контролируемые зоны (ECDF)

При активизации функций расширения возможно использование программного контроля местоположения оси максимум в восьми зонах. Данные зоны должны быть определены при характеристике оси в инструкции **SWn**. Сигнал, закреплённый за зоной **n**, устанавливается программным обеспечением в состояние «1», если ось находится в пределах заданной контролируемой зоны, и сбрасывается в «0», если ось выходит за пределы этой контролируемой зоны.

Примечание - Сигналы **PLC**, устанавливаемые для контролирования зон, не должны быть внутри базовой области интерфейсных сигналов пакета «К» для всех определённых процессов и не должны быть в пакете «А».

Допустимые действия оператора при входе в ту или иную контролируемую зону **SWn**, а также вывод соответствующих этому сообщений определяет ПЛ.

15.5. Контроль скорости осей (ECDF)

15.5.1. Контроль нулевой скорости оси

При активизации функций расширения возможно использование контроля нулевой скорости оси по его датчику обратной связи. Зона контроля скорости, равной нулю, и сигнал PLC назначаются пользователем при характеристике оси в инструкции **GM0**. Сигнал, закреплённый за контролируемой зоной скорости оси, устанавливается программным обеспечением в состояние «1», если реальная скорость оси находится в пределах заданной контролируемой зоны, и сбрасывается в «0», если ось выходит за пределы этой контролируемой зоны.

Примечание - сигналы PLC, устанавливаемые для контролирования зон, не должны быть внутри базовой области интерфейсных сигналов пакета «К» для всех определённых процессов и не должны быть в пакете «А».

Допустимые действия оператора, если скорость оси не достигла заданной, и вывод сообщений об этом определяет ПЛ.

15.5.2. Контроль нулевой скорости вращения шпинделя

При активизации функций расширения возможно использование контроля скорости вращения шпинделя по его датчику обратной связи. Значение допуска нулевой скорости вращения шпинделя и сигнал **PLC**, контролирующий этот допуск, назначаются пользователем при характеристике оси шпинделя в инструкции **GM1**. Сигнал, закреплённый за контролируемой зоной скорости вращения шпинделя, устанавливается программным обеспечением в состояние «1», если реальная скорость вращения находится в пределах заданной контролируемой зоны, и сбрасывается в «0», если ось выходит за пределы этой контролируемой зоны.

Примечание - Сигналы **PLC**, устанавливаемые для контролирования зон, не должны быть внутри базовой области интерфейсных сигналов пакета «К» для всех определённых процессов и не должны быть в пакете «А».

Допустимые действия оператора, если скорость вращения шпинделя после команды **S0M3** или **S0M4** не достигла заданной, и вывод сообщений об этом определяет ПЛ.

15.5.3. Контроль скорости вращения шпинделя при значениях S больше нуля

При активизации функций расширения возможно использование контроля скорости вращения шпинделя по его датчику обратной связи. Зона контроля скорости и сигнал **PLC** назначаются пользователем при характеристике оси шпинделя в инструкции **POM**. Сигнал, закреплённый за контролируемой зоной скорости вращения шпинделя, устанавливается программным обеспечением в состояние «1», если реальная скорость вращения находится в пределах заданной контролируемой зоны, и сбрасывается в «0», если ось выходит за пределы этой контролируемой зоны.

Для отмены контроля скорости вращения шпинделя, если активна функция поддержания скорости резания **G96**, можно использовать сигнал **FG96 (I9K26)**.

Примечание - Сигналы **PLC**, устанавливаемые для контролирования зон, не должны быть внутри базовой области интерфейсных сигналов пакета «К» для всех определённых процессов и не должны быть в пакете «А».

Допустимые действия оператора, если скорость шпинделя не достигла заданной с адресом S, и вывод сообщений об этом определяет ПЛ.

15.6. Авария

Под аварией понимается какая-либо аномальная или опасная ситуация, при которой возможно нарушение работоспособности станка, и станок подлежит выключению. При этом УЧПУ визуализирует сообщение об аварии в зависимости от вызвавшей её причины.

Имеется три типа аварийной ситуации:

- 1) аварийный останов по запросу от системы;
- 2) аварийный останов по запросу от ПЛ;
- 3) аварийный останов по запросу оператора.

В случае аварийного останова по запросу от системы сама система обнаруживает какую-либо аномалию в работе памяти, модулях вх./вых. или же потерю шага оси. При этом в пакет «К» выдаётся сигнал **EMERG (I00K0)**.

В случае аварийного останова по запросу от ПЛ аномалия обнаруживается ПЛ. Вследствие этого ПЛ должна запросить выключение станка посредством установки в УЧПУ сигнала **RISPE (U10K7)** и одновременно выдать диагностическое сообщение.

В случае аварийного останова по запросу оператора, как только сигнал **MUSPE=1**, система сбрасывает готовность **CONP=0 (I00K2=0)** и ви-

зуализирует сообщение о причине выключения станка, т.е. третий случай аварийной ситуации аналогичен второму случаю.

15.7. Приостанов

Под аварийной ситуацией типа **ПРИОСТАНОВ** понимается останов без изменения ситуации на станке. **ПРИОСТАНОВ** может запрашиваться асинхронным способом:

- 1) от УЧПУ при отработке кадра в режиме «**КАДР**»;
- 2) по запросу оператора нажатием клавиши «**СТОП**» на пульте;
- 3) по запросу ПЛ посредством установки сигнала **HLDR** или **RHOE**.

При выборе оператором процесса системы **ПРИОСТАНОВ** действует для осей всех процессов, а в противоположном случае – для осей выбранного процесса. Процедура управления ситуацией **ПРИОСТАНОВ** может быть подразделена на три отдельные фазы:

- 1) вход в **HOLD**;
- 2) разрешение на выход из **HOLD**;
- 3) выход из **HOLD**.

В течение первой фазы перемещения по осям останавливаются с управляемым замедлением, сбрасываются соответствующие сигналы перемещений (**MOVn**), затем устанавливается в «1» сигнал **HOLDA (I00K28)**, сообщающий ПЛ о достижении состояния **HOLD**.

Во время второй фазы проверяются сигналы **HLDR (U10K2)** и **RHOE (U10K3)**. Если они находятся в активном состоянии, то визуализируется диагностическое сообщение.

Для выхода из ситуации **HOLD** необходимо, чтобы оператор нажал на клавишу «**СТОП**». В любом случае, выход из **HOLD** разрешается только, если ПЛ даёт разрешение на выход посредством установки сигналов **HLDR=0** и **RHOE=0**.

В течение третьей фазы восстанавливается активное состояние тех сигналов перемещения (**MOVn**), которые были активны в момент входа в **HOLD**, сбрасывается сигнал **HOLDA** и выполняется продолжение (возможное) прерванного перемещения, если сигнал **COMU=1**.

Следует заметить, что некоторые операции разрешены даже при нахождении УЧПУ в состоянии **HOLD**, а именно:

- 1) ручное перемещение по одной из осей (для выхода из **HOLD** ось должна быть возвращена на профиль, если она была уведена с него);
- 2) ввод кадра с клавиатуры для активизации вспомогательных функций, которые могут быть приняты в состоянии **HOLD** (см. файл **IOSFIL**).

15.8. Останов перемещения

При запросе **FOLD=1 (U10K5)** выполняется останов текущих перемещений по осям как при автоматическом, так и при ручных перемещениях.

При исчезновении этого запроса **FOLD=0** прерванное перемещение возобновляется автоматически.

16. ВЫПОЛНЕНИЕ ВСПОМОГАТЕЛЬНЫХ ФУНКЦИЙ

Существует четыре различных типа вспомогательных функций, которые различаются по способу их выполнения.

К этим функциям относятся:

- 1) вспомогательные функции стандартного цикла;
- 2) вспомогательные функции начала перемещения;
- 3) вспомогательные функции конца перемещения;
- 4) функции немедленного действия.

Кроме этого, вспомогательные функции разделены на четыре семейства, различающихся по типу выполняемых операций. К ним относятся: «S», «T», «M», «индексная ось». Вспомогательные функции могут управляться обычным способом или с запоминанием.

- 1) В первом случае функции выполняются синхронизированно внутри кадра программы.
- 2) Во втором случае одна функция выполняется за другой так, чтобы можно было возобновить обработку детали именно с того места программы, где она была прервана.

Посредством запроса **RCM** управляющая программа выполняется системой без активизации каких-либо перемещений и без передачи в пакет «K» какой-либо вспомогательной функции. После окончания поиска с запоминанием посредством активизации оператором команды **ERM** в пакет «K» передаются вспомогательные функции, прошедшие буферизацию, причём в том порядке, в котором они были запрограммированы в кадре. Затем необходимо выполнить возврат осей на профиль, после которого можно будет возобновить работу с того кадра программы, где ранее произошло ее прерывание.

Примечания

1. Если при конфигурации системы индексная ось была задана как инкрементальная ось, то после окончания поиска с запоминанием УЧПУ передаёт в пакет «K» информацию о перемещении, эквивалентную разности между начальной точкой (нуль оси) и точкой, вычисленной при поиске.
2. Если же индексная ось была задана как абсолютная, то после окончания поиска с запоминанием в пакет «K» будет передана последняя информация о перемещении, запрограммированная для индексной оси.
3. Если в технологической программе имеются переходы, то на оператора возлагается задача по возобновлению условий, предшествующих проверкам параметров.

16.1. Вспомогательные функции стандартного цикла

К вспомогательным функциям стандартного типа относятся функции, генерируемые системой при выполнении стандартных циклов обработки. Данные функции служат для выдачи информации в ПЛ, когда следует выполнить реверс, ориентацию или останов вращения шпинделя во время циклов сверления, нарезания резьбы метчиком и фрезерования. К ним относятся нижеследующие функции:

- 1) останов шпинделя;
- 2) реверс шпинделя;
- 3) ориентация шпинделя;
- 4) восстановление исходного направления вращения шпинделя.

Эти данные необходимы, когда шпиндель управляется двигателем переменного тока.

Вспомогательные функции стандартного цикла выполняются в начале перемещения по одной на кадр и поотдельности, т.е. не может быть кадра программы, содержащего одновременно вспомогательные функции пользователя «S», «T», «M» и «индексная ось». Каждой из этих функций присваивается в пакете «K» один сигнал, служащий для идентификации типа операции, которую должен выполнить интерфейс.

16.2. Вспомогательные функции начала перемещения

Вспомогательные функции начала перемещения выдаются перед каким-либо возможным перемещением по осям. К ним относятся:

- 1) функции стандартного цикла;
- 2) функции «индексной оси»;
- 3) функции «S» и «T»;
- 4) функции «M» начала перемещения, задаваемые при конфигурации.

Существуют два способа управления вспомогательными функциями, (способ выбирается при конфигурации системы):

- 1) параллельное управление;
- 2) последовательное управление.

Если используется параллельный способ управления, и если в кадре программируются несколько вспомогательных функций начала перемещения, принадлежащие к различным семействам (за исключением «индексной оси», которая всегда выполняется последней), то все эти функции выполняются вместе за одну «посылку».

Если же, кроме этого, имеется также несколько вспомогательных функций из одного и того же семейства, то они включаются во вторую «посылку» снова все вместе и по одной на семейство и т.д. в порядке их программирования. Выдача каждой «посылки» вспомогательных функций синхронизируется с состоянием **SEFA=1 (U10K25)**.

16.3. Вспомогательные функции конца перемещения

Вспомогательные функции конца перемещения выполняются после перемещения по осям в том кадре программы, где они были запрограммированы. К вспомогательным функциям конца перемещения относятся функции «M», но только если они были заданы таковыми на фазе конфигурации.

Существуют два типа вспомогательных функций конца перемещения:

- 1) обычные вспомогательные функции;
- 2) вспомогательные функции блокировки вычислений.

К первым относятся те функции конца перемещения, для которых выдача информации в пакет «К» синхронизирована с состоянием **CEFA=1**.

Ко вторым относятся вспомогательные функции конца перемещения, которые могут запрашивать перемещение по осям от системы (**MAS, W12K1**). Для разрешения отработки команд **MAS** ПЛ должна синхронизироваться с интерфейсом посредством двух сигналов: **CEFA** и **CEFAB**.

При этом используется следующий протокол связи:

- Выдача каждой «посылки» (передачи информации в пакет «К») функций конца перемещения связана с условием **CEFA=1 (U10K25)** и с состоянием **CEFAB=1 (U10K26)**, а, если функция предусматривала запрос на смену коррекций, то редактируются и значения коррекций. Для запроса на смену коррекций обязательно, чтобы вспомогательная функция запрашивала блокировку вычислений. Для исполнения команд **MAS** сигнал **CEFAB** должен удерживаться в состоянии «0» до тех пор, пока не будет закончено все перемещение по осям, заданное от системы. После этого, устанавливая в «1» вышеуказанный сигнал **CEFAB** (а при наличии функций, требующих смены коррекций, также активизируя новый корректор), можно продолжить просмотр всех последующих вспомогательных функций конца перемещения.
- Сигнал **CEFAB** позволяет программировать вспомогательные функции начала и/или конца перемещения внутри какого-либо перемещения осей системы. Единственным условием, которое необходимо соблюдать, является следующее условие: функции, введённые в **MAS**, не должны требовать блокировки вычислений.

16.4. Вспомогательные функции немедленного действия

Эти функции передаются в пакет «К» в виде двух цифр в коде BCD после вспомогательных функций начала перемещения и действуют во время всего интервала перемещения осей, даже с активными функциями **G27** или **G28**. Однако они передаются в зону, отличную от той, где записаны функции «М» начала и конца перемещения. Эти функции не зависят от состояния сигнала **CEFA**.

17. ФУНКЦИЯ «S»

Функция «S» – это функция начала перемещения, принимаемая даже в состоянии **HOLD**. В одном кадре программы может программироваться только одна функция «S». Функция «S» определяет скорость вращения шпинделя. Она задается в виде пяти цифр, передаваемых в пакет «K» в формате: «код BCD + строб FUAS» (для шпинделя без преобразователя значение функции «S» записывается также в первые три слова разъёма **I08K**, но в зависимости от процентов, установленных корректором скорости вращения шпинделя «S»). Цифры BCD остаются зафиксированными до смены.

Дополнительно к функции «S» могут быть заданы четыре сигнала в пакете «K»: **SGAM1-SGAM4 (I06K16-I06K19)**, которые определяют диапазон запрограммированной функции «S».

Пределы режимов вращения для каждого диапазона задаются в файле характеристики **AXCFIL**. Тип управления шпинделем определяется в инструкции **TRA** файла **AXCFIL**:

- 1) шпиндель с двигателем постоянного тока – внутреннее управление;
- 2) шпиндель с двигателем переменного тока – внешнее управление.

Внутреннее управление шпинделем:

- шпиндель управляется непосредственно от аналогового сигнала, пропорционально запрограммированной функции «S». Сигнал на выходе ЦАП управляется от ПЛ посредством сигналов **ROMAO (U11K3)** (вращение по часовой стрелке) и **ROMAA (U11K4)** (вращение против часовой стрелки). Обновление выходного сигнала ЦАП выполняется только после установки в «1» сигнала выбранного диапазона, принимаемого от ПЛ **GAMx (U11K16-U11K19)**.

Внешнее управление шпинделем:

- в пакет «K» передаются только пять цифр в коде BCD, соответствующие запрограммированной функции «S», и строб **FUAS** – импульсный, длительностью 2 цикла «медленной» части ПЛ. ПЛ также информируется сигналами **INVER (I09K24)** и **STOPR (I09K25)**, когда вращение шпинделя должно быть реверсировано или остановлено.

18. ФУНКЦИЯ «Т»

Функция «Т» выполняется в начале перемещения и может приниматься в состоянии **HOLD**. Программируется в формате: **T1234**, т.е. посредством не более четырёх цифр. Функция может иметь два типа управления:

- 1) обычное управление;
- 2) произвольное управление (**RANDOM**).

Обычное управление:

- интерфейс не выполняет никакого контроля для запрограммированного инструмента. Выполняется только передача в пакет «К» четырёх цифр в формате: «BCD + строб **FUAT(I04K17)**». Цифры фиксируются до смены, а строб **FUAT** – импульсный, длительностью 2 цикла ПЛ.

Произвольное управление (**RANDOM**):

- оператору необходимо создать файл командой **EDI**, например: **EDI,RANDOM/MP2**, а затем заполнить его, т.е. объявить для каждой строки этого файла те номера инструментов, которыми надо управлять (номер строки в файле соответствует гнезду магазина инструментов), и указать это имя в инструкции **FIL** файла **PGCFIL**. После этого интерфейс при смене инструмента выполнит присвоение соответствующего гнезда (позиции магазина) смененному инструменту, т.е. выполнит обновление таблицы инструментов в файле **RANDOM**. При произвольном способе управления (**RANDOM**) интерфейс производит и ряд проверок для запрограммированной функции «Т». Проверки должны установить достоверность запрограммированной функции «Т» и необходимость поиска в инструментальном магазине. Возможны случаи, когда функция «Т» запрограммирована, но она не будет вызывать поиска в инструментальном магазине, поскольку очередной инструмент будет загружаться в шпиндель извне (не из магазина, а путём ручной смены инструмента). Если поиск в магазине необходим, то в пакет «К» передаются цифры в коде: «BCD + строб **FUAT**», точно так же, как для первого способа управления. В пакет «К» выдаются как цифры, соответствующие запрограммированной функции «Т», так и цифры, соответствующие тому гнезду магазина, где находится запрограммированный инструмент.

18.1. Способы управления функцией «Т»

При произвольном способе управления функцией «Т» нужно выделить следующие операции:

- 1) Смена коррекций.

Программирование какой-либо функции «Тх», где «Тх» – инструмент, стоящий в шпинделе (аналогично запросу на смену коррекций без какого-либо физического поиска). В этом случае интерфейс передаёт в пакет «К» номер гнезда магазина, равный 0. Таким образом, ПЛ (пакет «К») не получает сообщения о гнезде магазина, которое надо искать, а получает только сообщение о запрограммированной функции: «Т + строб FUAT (или FUTKO)».

2) Ручная смена инструмента.

Можно запрограммировать ручную смену инструмента, программируемого с функцией «Т», который не содержится в файле присвоения **Гнездо = Инструмент** (файл **RANDOM**). В этом случае устанавливается в «1» сигнал **CUMAN (I04K26)**. ПЛ информируется о запрограммированной функции «Т + строб FUAT», но не о гнезде, которое следует искать. Инструмент, стоявший в шпинделе, должен быть выгружен вручную оператором, а стирание его из файла выполняется автоматически.

3) Обычная смена инструмента.

Если программируется какая-либо функция «Т» инструмента, содержащегося в файле **RANDOM**, то в пакет «К» передаётся:

- запрограммированная функция «Т»;
- гнездо, где находится запрограммированный инструмент;
- строб функции «Т»: **FUAT**.

Программируя какую-либо функцию «Т» с номером инструмента, находящегося в таблице, после ручной смены инструмента УЧПУ выполняет выгрузку в магазин загруженного вручную инструмента и запись его в таблицу инструментов.

4) Загрузка инструментов в магазин.

Загрузка инструментов в магазин может выполняться двумя способами.

Первый способ предусматривает, что оператор загружает инструменты вручную или при помощи определённых средств (обычно станочный пульт) и затем «объявляет» загруженные инструменты посредством записи в файле **RANDOM**.

При втором способе можно выполнить загрузку инструмента из шпинделя. Последовательность выполняемых операций такова:

- программирование **ТхМ6** для загрузки инструмента;
- система принудительно устанавливает режим ручной смены инструмента, поскольку **Тх** не содержится в таблице;
- ручная загрузка инструмента оператором в шпиндель;
- программирование **ТОМ6**; оно активизирует освобождение шпинделя, и таким образом в пакет «К» выдаётся функция «Т», соответствующая одному свободному

- гнезду (если есть), в противном случае - сигнализируется о заполненности таблицы;
- занесение в файл **RANDOM** инструмента, загруженного ранее. Тоже выполняется для других инструментов, подлежащих загрузке.

5) Специальные инструменты.

Предусмотрено управление «специальными» инструментами, т.е. инструментами, занимающими в магазине три гнезда (в файле **RANDOM** специальный инструмент занимает также три строки), например:

```

.....
0005
0010
9050
9050
9050
0000
0000
0000
.....

```

Специальный инструмент выявляется системой, если программируются четыре цифры, из которых первая цифра равна 9 (можно иметь 1000 специальных инструментов: 9000-9999).

Интерфейс аннулирует смену инструмента, если оператор хочет выполнить смену обычного инструмента на специальный. В этом случае необходимо выполнять загрузку (и разгрузку тоже) через шпиндель (посредством программирования **ТОМ6**).

19. ФУНКЦИИ «М»

Функции «М» передаются в пакет «К» в формате:

«2 цифры в коде BCD + строб FUAM(I04K18)»

Характеристики каждой функции «М» определяются в файле характеристики **IOSFIL** посредством трёх байтов в шестнадцатиричном формате, имеющих следующие значения, начиная с младшего бита:

1-й байт:

- 1) функция начала перемещения;
- 2) функция конца перемещения;
- 3) функция, принимаемая в состоянии **HOLD**;
- 4) функция не визуализируемая;
- 5) функции немедленного действия;
- 6) резерв;
- 7) функция модальная;
- 8) функция, подлежащая визуализации после сброса.

2-й байт:

- 1) свободный бит;
- 2) переход в режим «КАДР» по условию;
- 3) блокировка вычислений;
- 4) принудительная установка режима «КАДР»;
- 5) запрос на смену коррекции инструмента;
- 6) запрос на сброс в конце выполнения;
- 7) запрос на смену оси шпинделя;
- 8) свободный (бит).

3-й байт:

- | | | |
|--|---|-----------------------------|
| <ol style="list-style-type: none"> 1. 2. 3. 4. | } | класс визуализации; |
| <ol style="list-style-type: none"> 5. 6. 7. 8. | } | класс поиска с запоминанием |

На фазе инициализации все используемые функции «М» должны быть объявлены в файле **IOSFIL**. В случае если программируется какая-либо функция «М», не определённая в файле характеристики, сигнализируется ошибка. В зависимости от того, как эти функции определены в файле характеристики, они выполняются в начале или в конце перемещения (синхронизируясь с сигналом **SEFA (U10K25)**) или же, как функции немедленного действия.

Функции «М» передаются в пакет «К» в формате «2 цифры в коде BCD + строб FUAM» (строб импульсный, длительностью 2 цикла ПЛ), если это функции начала/конца перемещения, или «2 цифры в коде BCD», которые сохраняются в течение всего времени отработки перемещения осей, если эти функции – немедленного действия.

В каждом кадре программы может быть запрограммировано не более четырёх функций «М» начала или конца перемещения плюс одна функция «М» немедленного действия.

20. ФУНКЦИЯ «ИНДЕКСНАЯ ОСЬ»

Функция «**Индексная ось**» – вспомогательная функция начала перемещения, не принимаемая в состоянии **HOLD**. Система может управлять движением 3-х «**индексных осей**», заданных в кадре. Имена осей, используемые при программировании, объявляются на фазе конфигурации в файле **IOCFIL**. Запрограммированная величина перемещения сообщается в пакет «**К**» посредством восьми цифр **BCD** в формате 5.3. Эта информация сохраняется в разъёме до её смены или сброса. С помощью импульсных строб – сигналов **TASCn (I04K20–I04K22)**, длительностью 2 цикла ПЛ, система сообщает, по какой из координат задано движение.

Сигнал **SESC (I04K24)**, передаваемый в пакет «**К**», является общим для трёх осей и определяет запрограммированный знак направления вращения оси.

Каждая индексная ось может быть задана на фазе характеристики двумя способами:

- 1) как абсолютная, при этом интерфейс всегда сообщает позицию, в которую следует переместить ось; таким способом можно определить 99999.999 различных позиций;
- 2) как инкрементальная, при этом интерфейс сообщает число делений (позиций) для перемещения оси (макс. 99999.999 делений).

ВНИМАНИЕ! ПОСЛЕ ВКЛЮЧЕНИЯ УЧПУ ПЕРВОНАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ ДОЛЖНО ВКЛЮЧАТЬ В СЕБЯ «ОБНУЛЕНИЕ» «ИНДЕКСНЫХ ОСЕЙ», Т.Е. НЕОБХОДИМО ЗАПРОГРАММИРОВАТЬ ПЕРЕМЕЩЕНИЕ В ПОЗИЦИЮ **0000** ИЛИ СИГНАЛИЗИРОВАТЬ СИСТЕМЕ О ВЫПОЛНЕННОМ «ОБНУЛЕНИИ» СИГНАЛАМИ **MIZE1 (U10K27)**, **MIZE2 (U10K28)**, **MIZE3 (U10K29)**, КОТОРЫЕ ДОЛЖНЫ ВСЕГДА ОСТАВАТЬСЯ АКТИВНЫМИ (ИНАЧЕ БУДЕТ СИГНАЛИЗИРОВАТЬСЯ ОШИБКА).

21. УПРАВЛЕНИЕ ЗАПРОСАМИ ОТ ПЛ

Интерфейс предоставляет ПЛ определенное число циклов в целях расширения оперативных возможностей программиста.

Данные циклы могут активизироваться сигналами запроса. Сигналы запроса можно подразделить на две категории:

- 1) асинхронные запросы (принимаемые в любой момент);
- 2) синхронные запросы (принимаемые в особых условиях).

21.1. Асинхронные запросы

К возможным асинхронным запросам относятся:

- 1) обновление сигнала ЦАП шпинделя;
- 2) сигналы управления шпинделем;
- 3) позиционирование осей «от точки к точке» от ПЛ;
- 4) обновление инструмента шпинделя и коррекций;
- 5) визуализация сообщения;
- 6) принудительная установка сигнала в канале ЦАП;
- 7) управление пультом от ПЛ;
- 8) управление штурвалами;
- 9) режим обучения (**TEACHING**).

21.1.1. Обновление сигнала ЦАП шпинделя

Система выдает аналоговый сигнал, который может использоваться для управления электродвигателем постоянного тока.

ПЛ путём определенных установок и соответствующих запросов информации может выдавать последовательно с течением времени определенный уровень этого аналогового сигнала.

Для расчёта коэффициента ЦАП используются четыре сигнала включения диапазонов вращения шпинделя, а именно: **GAMM1-GAMM4 (U11K16-U11K19)**.

21.1.2. Сигналы управления шпинделем

Кроме указанных выше сигналов имеются еще пять сигналов, которые по-разному активизируют выход аналогового сигнала. К ним относятся:

- 1) сигнал **ANGOM (U11K0)** - служит для ориентации шпинделя. Шпиндель удерживается в позиции до тех пор, пока этот запрос активен. Используется для шпинделя:
 - с датчиком положения типа энкодер (фотоимпульсный, с отношением 1 оборот шпинделя - 1 оборот датчика);
 - с датчиком положения типа энкодер (фотоимпульсный, с отношением 1 оборот шпинделя - **N** оборотов датчика).

В любом случае шпиндель позиционируется посредством таких параметров как скорость, ускорение, коэффициент усиления, допуск позиционирования. Эти параметры задаются на фазе конфигурации системы.

- 2) сигналы **F0MA0** и **F0MAA** (**U11K1**, **U11K2**) – служат для запуска вращения шпинделя со скоростью, не запрограммированной в кадре, а принудительной, задаваемой от ПЛ двумя цифрами ВСD в слове **W11K1**. Эти две цифры выражают значение напряжения в вольтах, требуемое на выходе 0,1–9,9. Сигнал **FORID** (**U11K5**) служит для деления на 10 заданного принудительного напряжения (если это необходимо).
- 3) сигналы **ROMA0** и **ROMAA** (**U11K3**, **U11K4**) – служат для выдачи на выходе сигнала ЦАП, пропорционального запрограммированной функции «S» и диапазону, заданному от ПЛ. Интерфейс информирует ПЛ посредством сигналов **SGAM1-SGAM4** (**I06K16-I06K19**) о том, какому диапазону принадлежит функция «S».

В случае одновременности запросов сигналы, указанные выше, действуют с приоритетом в соответствии с порядком их упоминания (т.е. **ANGOM** – **F0MA0** – **F0MAA** – **ROMA0** – **ROMAA**).
- 4) сигналы **SMRIF** и **SMCZ** (**U16N0**, **U16N1**). Сигнал **U16N0** устанавливает режим шпинделя, когда 1 оборот шпинделя соответствует **N** оборотов датчика. Сигнал **U16N1** – запрос поиска ближайшей «ноль-метки» шпинделя (обычно устанавливается при наезде на концевик и сбрасывается сразу при съезде с концевика шпинделя, при активном сигнале **SMRIF**). Данные сигналы устанавливаются для обеспечения ориентации шпинделя и резьбонарезания.

ВНИМАНИЕ! ПЕРВОЕ НАПРАВЛЕНИЕ ВРАЩЕНИЯ ШПИНДЕЛЯ ОПРЕДЕЛЯЕТ ОДНУ ИЗ ДВУХ (ЛЕВУЮ ИЛИ ПРАВУЮ) «НОЛЬ-МЕТКУ» ОТНОСИТЕЛЬНО КОНЦЕВИКА. ПОЭТОМУ ПОСЛЕ ВКЛЮЧЕНИЯ УЧПУ ПЕРВОЕ НАПРАВЛЕНИЕ ВРАЩЕНИЯ ШПИНДЕЛЯ В ПРЕДЕЛАХ ОДНОГО ЕГО ОБОРОТА ДОЛЖНО БЫТЬ ОБЕСПЕЧЕНО ВСЕГДА В ОДНОМ НАПРАВЛЕНИИ.

- 5) сигнал **RIFOR** (**U16N2**). Состояние сигнала **RIFOR** определяет два алгоритма ориентации шпинделя по запросу сигналом **ANGOM**:
 - алгоритм ориентации при состоянии сигнала **RIFOR=0** выполняется с использованием накопителя импульсов датчика шпинделя (стандартный алгоритм). «Ноль-метка» датчика определяется при первом обороте вращения или первой ориентации шпинделя и далее ориентация выполняется как позиционирование в координату учитывающую значение смещения (первый параметр инструкции **POM** и разъем «14K»).

Если шпиндель после включения станка не вращался, то первая ориентация выполняется на скорости, определенной вторым параметром **POM**. Направление ориентации зависит от состояния сигналов **ORIMO**

(U11K6) и ORIMA (U11K7). Если сигналы **U11K6=0** и **U11K7=0**, то направление ориентации шпинделя будет соответствовать направлению вращения по сигналу **ROMOO**, последующие ориентации выполняются с разгоном и торможением по значениям параметров инструкции **TSM** по кратчайшему пути.

Если состояние одного из сигналов **U11K6** или **U11K7** не равно «0», то направление ориентации шпинделя будет соответствовать направлению вращения, которое задано одним из них.

- алгоритм ориентации при состоянии сигнала **RIFOR=1** выполняется всегда с поиском «ноль-метки» датчика шпинделя (новый алгоритм). «Ноль-метка» датчика определяется при каждой ориентации. Ориентация всегда выполняется на скорости, определённой вторым параметром **РОМ**, и всегда в одном направлении. Накопитель импульсов датчика шпинделя при нахождении «ноль-метки» датчика обнуляется, после чего отрабатывается смещение, определённое первым параметром **РОМ** и значением разъёма «14К», по кратчайшему пути с разгоном и торможением по значениям параметров инструкции **TSM**.

Значение скорости ориентации должно быть в пределах **V_{РОМmin}** и **V_{РОМmax}**, которые рассчитываются по формулам:

$$V_{РОМmin} = 60000 / (\text{электрический шаг} / \text{механический шаг}) / \text{ТИК из CAS}$$

$$V_{РОМmax} = \pm 32768 / (\text{электрический шаг} / \text{механический шаг})$$

Кроме того, что значение **V_{РОМ}** должно быть в промежутке между значениями **V_{РОМmin}** и **V_{РОМmax}**, оно должно быть больше дрейфа на шпинделе.

ВНИМАНИЕ! СИГНАЛ **RIFOR (U16N2)** ДОЛЖЕН БЫТЬ УСТАНОВЛЕН В СОСТОЯНИЕ «0» ИЛИ «1» ДО ЗАПРОСА ОРИЕНТАЦИИ СИГНАЛОМ **U11K0 (ANGOM)** И УДЕРЖИВАТЬСЯ В ЭТОМ СОСТОЯНИИ НА ВСЁ ВРЕМЯ ОРИЕНТАЦИИ. ЕСЛИ ВЫБРАННЫЙ АЛГОРИТМ ОРИЕНТАЦИИ ДОЛЖЕН БЫТЬ ЕДИНСТВЕННЫМ, ТО СИГНАЛ **U16N2** МОЖНО УСТАНОВИТЬ В СООТВЕТСТВУЮЩЕЕ СОСТОЯНИЕ «1» ИЛИ «0» НА ВСЁ ВРЕМЯ РАБОТЫ УЧПУ, НАПРИМЕР: **U16N2=I0K2**.

21.1.3. Позиционирование осей «от точки к точке» от ПЛ

Система предоставляет ПЛ два одинаковых и отдельных цикла для позиционирования восьми осей «от точки к точке».

Оси можно позиционировать на конечное число эквидистантных позиций (гнезд), максимум – 99999.999. Запрос на позиционирование оси производится посредством двух слов, которые однозначно определяют пару осей, подлежащих перемещению.

Когда в одном из двух программных каналов активизируется запрос на движение, система во время всего интервала времени перемещения осей сигнализирует ПЛ о занятости канала сигналами **BUSY1**, **BUSY2 (I05K26, I05K27)** соответственно для первого и второго канала.

Если во время позиционирования одной оси в том же канале обнаруживается запрос на позиционирование другой оси, система аннулирует этот запрос и сигнализирует ошибку посредством сигналов **KOSI1**, **KOSI2** (**I05K28**, **I05K29**).

При позиционировании оси «от точки к точке» система использует кроме запросов от ПЛ также некоторые тарировки, установленные при характеристизации (см. «Руководство по характеристизации»).

Позиционирование оси «от точки к точке» выполняется по следующему алгоритму:

- 1) активизация сигнала **BUSY**, соответствующего занятому каналу;
- 2) проверка выполнения выхода оси на микровыключатель абсолютного нуля после включения УЧПУ; если он не выполнен, начинается цикл поиска микронуля, причём скорость, направление и его адрес должны быть заданы при характеристизации в инструкции **MCZ** файла **ACXFIL**;
- 3) если нет принудительной установки направления движения от ПЛ сигналами **FOPA** (**U12K4**, **U12K5**) и **FONA** (**U12K6**, **U12K7**), система выбирает его путём считывания датчика положения, после чего сообщает о направлении перемещения ПЛ сигналами **ROPO** (**I05K16**, **I05K17**) и **RONE** (**I05K18**, **I05K19**). Сигналы **FOPA** и **FONE** имеют значение только в том случае, если ось является циклической;
- 4) ожидание факта достижения осью координаты первого замедления, определенной при конфигурации;
- 5) активизация сигнала **ROLE** (**I05K20**, **I05K21**) – первое замедление;
- 6) ожидание факта достижения осью координаты второго замедления, определенной при конфигурации;
- 7) активизация сигнала **ROLLE** (**I05K22**, **I05K23**) – второе замедление;
- 8) ожидание факта входа оси в зону допуска, тарированного при характеристизации;
- 9) активизация сигнала «ось в позиции» **POSI** (**I05K24**, **I05K25**);
- 10) проверка сброса запроса;
- 11) сброс сигналов **POSI** и **BUSY**.

Примечание – Если двигатель оси определён при конфигурации не как двигатель постоянного тока, т.е. когда не используется аналоговый сигнал, задачей ПЛ будет являться выполнение конкретного позиционирования оси сигналами направления, замедления, позиционирования.

21.1.4. Позиционирование осей «от точки к точке» от ПЛ (ЕСDF)

21.1.4.1. Включение осей

Управление осями (максимум 8) возможно по двум независимым параллельным каналам. Для запроса включения осей необходимо установить их номера (номер каждой оси определяется индексом «n» в инструкции **TAn** файла **IOCFIL**) в байтах:

- **W20K0** – для управления одной осью в первом канале;
- **W20K1** – для управления другой осью во втором канале.

Про по запросам включения займёт соответствующий канал для их управления и вернёт признаки захвата канала сигналом **BUSY1 (I5K26)** и «Ось в допуске позиционирования» сигналом **POS11 (I5K24)**, если номер оси был установлен в первом канале, и сигналами **BUSY2 (I5K27)** и **POS12 (I5K25)**, если номер оси был установлен во втором канале. При этом оси станут контролироваться по датчику обратной связи.

Каждая ось «от точки к точке» должна быть выведена в ноль до первого её перемещения по программе. Выход оси в ноль выполняется в режиме работы «**ВЫХОД В НОЛЬ**» («**НОМЕ**»). Поведение оси до и после выхода в ноль отличаются.

21.1.4.2. Работа оси до её выхода в ноль

До первого выхода в ноль по оси, открытой для управления на любом канале, Про позволяет в системе по кнопке «**ПУСК**» выполнять следующие действия:

- движение оси в режимах работы «**БЕЗРАЗМЕРНЫЕ РУЧНЫЕ ПЕРЕМЕЩЕНИЯ**» («**MANU**») и «**ФИКСИРОВАННЫЕ РУЧНЫЕ ПЕРЕМЕЩЕНИЯ**» («**MANJ**») без «дотягивания» до целой позиции;
- движение оси в режиме «**ВЫХОД В НОЛЬ**» («**НОМЕ**»).

Примечания

1. Система игнорирует программное задание позиции для перемещения, записанное в разъёмах пакета «**К**» (18К или 19К), не выдавая на экран никакого сообщения. В данном случае сообщение необходимо формировать от ПЛ на основании сигналов-признаков выхода осей в ноль **W3K0**.
2. Система не обнуляет на экране индикацию значения подачи **F** для оси после общего сброса.
3. Система не посылает текущую позицию оси в разъем **4N**, если ось открыта в первом канале, и в разъем **5N**, если ось открыта во втором канале.

Скорость ручных перемещений ограничена значением максимальной скорости ручных перемещений, определённой для оси в инструкции **MAN** файла **AXCFIL** с учётом коррекции, устанавливаемой корректором подач «**JOG**».

21.1.4.3. Выход оси в ноль

Процедура выхода оси, имеющей ЦАП, в ноль выполняется аналогично процедуре выхода в ноль непрерывных координатных осей.

Примечание – Физическое смещение позиции нуля оси может быть выполнено с помощью значения первого параметра инструкции **ZNO** для данной оси в файле характеристики **AXCFIL**.

После выхода в ноль на индикацию в поле «**ФАКТ**» будет выведено значение смещения фаз между механическим и электрическим нулем (**величина 3** инструкции **TAn** файла **IOSFIL**), а для оси «**ТТ**» в режиме «**МАГАЗИН**

ИНСТРУМЕНТОВ» будет выведено значение, установленное в **парамetre 4** инструкции **PAS** файла **АХСF1L**.

В видеокдрах **«УПРАВЛЕНИЕ СТАНКОМ»** в строке **«Сообщение 4_XX»** индицируется сообщение: **«Ось выведена в ноль»**. В отсутствии текстового файла или соответствующей строки в нём - код: **«Сообщение 4_74»**.

Примечание - Процедура **«Выход в ноль»** может быть выполнена неограниченное количество раз.

21.1.4.4. Работа оси «ТТ» после выхода в ноль

После первого выхода в ноль по оси **«ТТ»** Пр0:

- устанавливает сигналы **POS11 (I5K24)** и/или **POS12 (I5K25)** в «1» и управляет их состоянием:
 - сбрасывает в «0» по началу движения;
 - устанавливает в «1» по концу перемещения с учётом допуска позиционирования;
- разрешает движение оси **«ТТ»** во всех режимах;
- выполняет перемещение с «дотягиванием» до целой позиции во всех режимах;
- выполняет программные перемещения, заданные от ПЛ через разъёмы **18K, 19K**;
- посылает её текущую позицию в разъём **4N**, если ось открыта в первом канале, и **5N**, если - во втором канале;
- выводит для оси **«ТТ»** в режиме **«Магазин Инструментов»** на индикацию в поле **«ФАКТ»** номер позиции (**параметр 4** инструкции **PAS** файла **АХСF1L**);
- обнуляет для оси **«ТТ»** величину подачи на экране после общего сброса.

Скорость ручных перемещений ограничена значением максимальной скорости ручных перемещений, определённой в инструкции **MAN** файла **АХСF1L** для соответствующей оси **«ТТ»** с учётом коррекции, устанавливаемой корректором подачи **«JOG»**.

Перемещение по оси **«ТТ»**, заданное от ПЛ через разъёмы **18K** и/или **19K**, выполняется на скорости, определённой в инструкции **RAP** файла **АХСF1L** для соответствующей оси **«ТТ»** с учётом коррекции, устанавливаемой в байтах **W20K2** и **W20K3**. Коррекция подачи в байтах **W20K2** и **W20K3** должна быть установлена до задания перемещения оси **«ТТ»**. Если изменение коррекции подачи было выполнено в момент движения оси **«ТТ»**, то оно игнорируется до завершения данного движения и активизируется в начале следующего.

Примечание - Величина подачи оси **«ТТ»** индицируется на экране справа от её фактической позиции сразу после символа **«F»**.

21.1.5. Общий сброс

При общем сбросе системы в момент движения оси «ТТ» будет произведено:

- до вывода оси в ноль:
 - контролируемое замедление скорости до нуля без последующего «дотягивания» до ближайшей позиции.
- после вывода оси в ноль:
 - контролируемое замедление скорости до нуля с последующим «дотягиванием» до ближайшей позиции;
 - обнуление подачи, если ось «ТТ» была выведена в ноль.

21.1.6. Ручные перемещения

Для выполнения ручных перемещений необходимо:

- установить режим «**MANU**» или «**MANJ**» (в режиме «**MANJ**» необходимо задать шаг, например: JOG=4);

Примечание - После вывода оси в ноль шаг, заданный дробным числом, для выполнения перемещения оси «ТТ» округляется до целого.

- установить клавишами «**СДВИГ НА СТРОКУ ВПЕРЕД**» или «**СДВИГ НА СТРОКУ НАЗАД**» маркер в поле «**ФАКТ**» на ось «ТТ», например: Т;
- установить корректором подач «**JOG**» нужное направление и скорость;
- выполнять движение при нажатии клавиши «**ПУСК**»;
- при отпускании клавиши «**ПУСК**» ось «ТТ» будет позиционирована в одну из ближайших позиций с учётом установленного ускорения (второй параметр в инструкции **MAN** файла **AXCFIL**).

Примечания

1. Если позиции оси «ТТ» расположены так близко друг от друга, что ось с заданным параметром ускорения не может быть позиционирована в ближайшую следующую позицию, то ось будет позиционирована в одну из следующих позиций с учётом установленного ускорения.
2. При одновременном задании движения от ПЛ и в ручных режимах будет индигировано сообщение: «**Ошибка системы**».

21.1.7. Программные перемещения оси

Программное задание позиции для перемещения оси «ТТ» должно быть записано в разъём **18К**, активизированный в первом канале, и в разъём **19К** - во втором. Номер позиции может быть переписан в эти разъёмы ПЛ с помощью:

- индексной оси (инструкция **ASn** секция 3 файла **IOCFIL**) через разъем **№1** пакета «**К**», например: дискетный поворотный стол.

Пример

DOF:I04K20


```

W18K0=W01K0
W18K1=W01K1
W18K2=W01K2
W18K3=W01K3
ENDF

```

- заданного номера инструмента под адресом «Т» (не **RANDOM**) через слова **W02K2** и **W02K3**.

Пример

```

; активизация оси «ТТ» №1 на канале1
W20K0=MUX(1,0), (I00K2,U10K0)
; управление скоростью оси корректором подачи «F»
W20K2=W1N1
; строб функции «Т», «Т» не «0»
DOF:I04K17*/([W2K2=0]*[W2K3=0])
W18K1=LOW(W2K2)
W18K1=XCH(W18K1)
W253K0=LOW(W2K3)
W18K2=XCH(W253K0)+HIG(W2K2)
W18K3=HIG(W2K3)
ENDF

```

- слов **W02K0** и **W02K1**, в которых записан номер ячейки магазина, где находится запрограммированный инструмент под адресом «Т» (**RANDOM**);
- входных сигналов от станка и т.д.

21.1.8. Обновление инструмента шпинделя и коррекций

ПЛ может направить запрос к системе посредством сигнала **AGTOL** (**U10K21**) на обновление и визуализацию инструмента, находящегося в шпинделе, и присвоенного ему корректора.

В случае, если обновление инструмента запрошено от какой-либо функции «М», система остается в состоянии ожидания (**WAIT**) до тех пор пока не выполнится запрос от ПЛ (сигнал **AGTOL=1**).

В зависимости от результата операции УЧПУ отвечает либо сигналом **АСКТО** (**I00K26**) (обновление инструмента выполнено), либо **НАСКТО** (**I00K27**) (не конгруэнтный запрос на обновление инструмента).

21.1.9. Запрос на визуализацию сообщения

Имеется возможность запроса на визуализацию сообщений, содержащихся в файле **FILMS5**. Команда на визуализацию может быть дана двумя различными способами:

- 1) сигналами разъемов **21K** и **22K**; этот способ позволяет визуализировать первые 64 сообщения, созданные в файле **ФАЙЛ_5**, одно за другим с убывающим приоритетом.

Пример

U22K3 = I0A1 - Когда I0A1 будет установлено в «1», будет визуализироваться сообщение 36 записи из файла **FILMS5**.

- 2) сигналами слова **W17K3**; этот способ позволяет визуализировать одно из 255 сообщений из ранее записанного файла **FILMS5**. Номер сообщения задаётся двоичным кодом в этом слове.

Пример

W17K3 = MUX (1, 13, 54, 253), (I0A1, I0A2, I0A3, I0A4)

21.1.10. Принудительная установка сигнала в канале ЦАП

Система предусматривает возможность принудительной установки сигнала в канале ЦАП, объявленного при конфигурации. ПЛ должна сделать запрос к системе, задавая значение аналогового сигнала (в вольтах), выраженного в формате BCD, в пакет «К» сигналами **FORZN=0** (положительное направление) и **FORZN=1** (отрицательное направление). Номер канала ЦАП выбирается путём задания сигналов **FORZ1**, **FORZ2** для осей, заявленных в инструкции **UCDA** файла **IOCFIL**.

Система выдает ступенчатый аналоговый сигнал, а заданное значение может изменяться в реальном масштабе времени.

21.1.11. Запрос управления пультом от ПЛ

Система позволяет управление пультом со стороны ПЛ посредством соответствующих запросов в пакете «К». Другими словами, имеется возможность запрета управления пультом для оператора и, в то же время, возможность присвоения этих функций ПЛ.

Для выполнения этого используют следующие функции:

- 1) выбор режима работы;
- 2) выбор осей;
- 3) присвоение величины «**JOG**» и направления ручных перемещений;
- 4) запрос на «**ПУСК**»;
- 5) присвоение значения корректору подачи.

Запрещение управления пультом для оператора может быть как частичным, путём установки в «1» отдельных сигналов слова **W15K0**, так и полным, путём установки в «1» всех сигналов вышеуказанного слова.

При получении запроса на перемещение осей, если этот запрос является конгруэнтным и, следовательно, может приниматься системой, о положительном результате операции ПЛ сообщается сигналом **АСКСО**, который остаётся в «1» до исчезновения запроса.

Перед отработкой запроса на «**ПУСК**» целесообразно подготовить пульт, т.е. необходимо, чтобы режимы, установленные с пульта, были конгруэнтны с запросом, подлежащим исполнению.

21.1.12. Управление штурвалами. Версии Про Z.33P (РИВ)

Внешнее управление штурвалом (штурвалами) выполняется Про и активизируется ПЛ в любом режиме работы. Управление осями от штурвалов может выполняться по двум независимым каналам. Штурвал, объявленный в инструкции **ADV** в первом параметре, закрепляется за управлением осью,

определённой на первом канале (**W15N0**). Штурвал, объявленный в инструкции **ADV** во втором параметре, закрепляется за управлением осью, определённой на втором канале (**W15N1**).

Движение оси (осей) выполняется при следующих условиях:

- 1) ось (оси) не должна быть отключена (**UAS=0**);
- 2) клавиши «ПУСК» и «СТОП» не должны быть активизированы;
- 3) состояние УЧПУ не должно быть в «HOLD»;
- 4) сигнал **COMU** должен быть активен (**U10K24=1**);
- 5) выбранная ось (оси) должна (ы) быть определена (ы) в инструкции **MAS** (секция 6 файла **PGCFIL**) и не должна (ы) являться виртуальной осью или осью «от точки к точке»;
- 6) ось не должна быть шпинделем, например, **NAS=S**;
- 7) сигнал **FOLD** не должен быть активным (**U10K05=0**);
- 8) штурвалу (штурвалам) должна быть назначена шкала в байте **W15N2**;
- 9) ось (оси) должна быть выбрана для движения в байтах **W15N0** для управления на первом канале и **W15N1** для управления на втором канале;
- 10) каналы управления штурвалами должны быть активизированы сигналами **U15N24** или **U15N25**

21.1.13. Режим обучения (TEACHING). Версия Z.70.10

Система позволяет посредством соответствующих запросов в пакете «К» сохранить перемещения, кадры со вспомогательными («M», «S», «T») или подготовительными («G») функциями в виде кадров управляющей программы, выполненные в режиме «MDI», «MANU» и «MANJ». Кадры УП в режиме обучения записываются в файле **TEACH**. Файл **TEACH** создаётся автоматически при активизации режима обучения. Местоположение файла **TEACH** определено инструкцией **NDD** (секция 4 файла **PGCFIL**).

Для работы в режиме обучения предназначены следующие сигналы:

- **TEACH (U13K1) – TACKN (I9K19)**
- **PSTOR (U13K2) – TPOSS (I9K20)**
- **BSTOR (U13K3) – TBLOS (I9K21)**
- **TPROB (I9K22)**

21.1.14. Компенсация смещения нуля привода

Чтение и компенсация «смещения нуля» приводов всех осей выполняется посредством двух запросов **RZD (U11K22)** и **WZD (U11K23)** соответственно. Устанавливать запрос **RZD** для чтения с последующим запросом **WZD** для компенсации можно, если компенсируемые оси включены и находятся в покое. Запрос **WZD** для постоянной компенсации должен оставаться в состоянии «1» до тех пор, пока не появятся условия для повторения цикла чтения и компенсации. Приоритет запроса **WZD** выше, чем у запроса **RZD**, поэтому, пока он находится в состоянии «1», состояние запроса **RZD** программным обеспечением не рассматривается.

Пример PLC

§

```
T0I(10) = "START" * I0K4 * I0K8 + T0D
U220K2 = T0U + U220K2*/"START"
U11K22=T0D
U11K23=/T0D*U220K2
```

Инициирование цикла чтения и компенсации «**смещения нуля**» всех приводов в данном примере выполняет слово ASCII-компаратора «**START**».

21.1.15. Компенсация смещения нуля привода (ECDF)

При активизации функций расширения возможно использование компенсации «**смещения нуля**» привода отдельно для каждой оси. Чтение и компенсация «**смещения нуля**» привода выполняется посредством запросов «**чтение смещения нуля привода**» (третий параметр инструкции **GAS**) и «**компенсации смещения нуля привода**» (четвертый параметр инструкции **GAS**) соответственно. Устанавливать запрос для чтения с последующим запросом компенсации по выбранной оси можно, если она включена и находится в покое. Запрос компенсации по выбранной оси должен оставаться в состоянии «1» до тех пор, пока не появятся условия для повторения цикла чтения и компенсации по этой же оси. Приоритет запроса компенсации выше, чем у запроса чтения, поэтому, пока он находится в состоянии «1», состояние сигнала чтения программным обеспечением не рассматривается.

Пример АХСF1L

```
NAS=X
GAS=, , U220K0, U220K1
```

Пример PLC

```
$
T0I(10) = "START" * I0K4 * I0K8 + T0D
U220K2 = T0U + U220K2*/"START"
U220K0=T0D
U220K1=/T0D*U220K2
```

Инициирование цикла чтения и компенсации «**смещения нуля**» привода оси X в данном примере выполняет слово ASCII-компаратора «**START**».

Процедура чтения-компенсации «**смещения нуля**» привода для шпинделя без датчика:

- 1) включить станок и ввести команду **S0M03**;
- 2) контролировать величину скорости вращения шпинделя по индикации;
- 3) задать **S<значение>**, равное скорости вращения шпинделя и обратное его направлению (направление выбрать функциями M03 или M04), и контролировать останов вращения шпинделя или вращение в допустимых пределах;
- 4) установить сигнал чтения, равным «1» (третий параметр в инструкции **GAS** для оси шпинделя);
- 5) установить сигнал компенсации, равным «1» (четвертый параметр в инструкции **GAS** для оси шпинделя), через одну секунду после выполнения предыдущего действия.

Процедура чтения-компенсации «**смещения нуля**» привода для шпинделя с датчиком:

- 1) включить станок, ввести команду ориентации шпинделя и удерживать его в этом состоянии (U11K0=1) до окончания процедуры чтения-компенсации;
- 2) установить сигнал чтения, равным «1» (третий параметр в инструкции **GAS** для оси шпинделя);
- 3) установить сигнал компенсации, равным «1» (четвертый параметр в инструкции **GAS** для оси шпинделя), через одну секунду после выполнения предыдущего действия.

21.1.16. Компенсация трения покоя оси на круговом контуре (ECDF)

При активизации функций расширения возможно использование компенсации трения покоя оси при её участии в выполнении кругового контура. Для включения или выключения этой функции с параметрами, установленными для неё в инструкции **FRC** в файле **AXCFIL**, необходимо установить сигнал **PLC** (параметр 9 в инструкции **FRC**) в состояние «1» или «0» соответственно. Сигнал **PLC** может быть установлен из ПЛ или из управляющей программы (см. переменные **SK** в документе «Руководство программиста МС/ТС»).

21.2. Синхронные запросы

К возможным синхронным запросам относятся:

- 1) выполнение записи файла перемещение осей от системы;
- 2) выбор программы или управления с клавиатуры;
- 3) запрос на «**СТОП**» программы;
- 4) запрос на «**ПУСК**» программы;
- 5) коммутация шпинделя.

21.2.1. Выполнение записи файла перемещений осей от системы

ПЛ может запросить выполнение перемещения осей в соответствии с подлежащей выполнению одной из 255 возможных записей (длина одной записи - 128 символов), имеющих в файле **FILMAS**, путём присвоения двоичного номера строки в этом файле слову **W12K1**. Файл **FILMAS** создаётся командой **EDI** и должен быть объявлен в секции 2 файла **FCRSYS**.

Запрос принимается системой в двух случаях:

- 1) во время отработки какой-либо функции «**М**» конца движения, вызывающей блокировку вычислений (см. характеристику функций «**М**» в документе «Руководство по характеристике»), при состоянии сигналов **CEFA=1** и **CEFAB=0**;
- 2) когда УЧПУ находится в состоянии **STAND-BY**, и переключатель режимов - в положении «**РУЧНОЙ ВВОД**».

В файле **FILMAS** могут быть записаны функции перемещения осей, подготовительные функции «G», функции «M», «S», «T», функции «индексной оси» и т.д., как в обычной управляющей программе, за исключением функции «M» с характеристикой «Блокировка вычисления».

Координаты позиционирования осей в файле **FILMAS** всегда задаются в миллиметрах. После выполнения записи система устанавливает в «1» сигнал **POSIA (I00K24)**, который остается активным до исчезновения запроса.

21.2.2. Запрос на выбор программы или управления с клавиатуры

ПЛ может запросить выбор какой-либо управляющей программы или же выполнение какой-либо команды, заданной с клавиатуры. Причём, имя программы или команды может находиться:

- 1) в файле **FILCMD**;
- 2) в одном из разъёмов пакета «K» в коде **ASCII**.

В случае, когда имя программы или команды находится в файле **FILCMD**, необходимо открыть файл с логическим именем **FILCMD** (имя файлу присваивается при характеристике в секции 2 файла **FCRSYS**). Этот файл создаётся и заполняется командой **EDI**, и имеет не более 255 строк с максимальной длиной записи, равной 128 символам.

Посредством директивы **EDI** необходимо ввести в строки имена возможных управляющих программ и устройство их хранения (по одному для каждой строки), или же команду, которую потребуется выбирать.

Пример

URL, 1
E8, 40
PROG17/MP3 и т.д.

Для запуска программы/команды, заданной с клавиатуры, необходимо выполнить последовательно одну за другой следующие операции:

- 1) активизировать сигнал **FILCMD (U10K23)**;
- 2) присвоить слову **W17K0** номер записи, содержащей имя программ/команды, задаваемой с клавиатуры, которую надо активизировать;
- 3) присвоить **CMDLOG1=1 (U10K19)**, если требуется выполнить команду, задаваемую с клавиатуры, нажатием клавиши «ENTER»;
- 4) активизировать сигнал **SPGCOM (U10K18)**.

Примечание - Команды могут быть выполнены списком, если они указаны в косвенном файле (см. описание сигнала **FILCMD (U10K23)**).

В случае, когда имя программы или команды находится в одном из разъёмов пакета «K» в коде **ASCII**, последовательность операций будет следующей:

- 1) присвоить слову **W17K0** номер первого разъёма пакета «К», в котором записаны в коде **ASCII** имя файла, который надо выбрать, и устройство его хранения;
- 2) присвоить **CMDLOG=1**, если надо запросить выполнение команды, задаваемой с клавиатуры;
- 3) записать в пакет «К» (начиная с первого из 10 имеющихся разъёмов, предварительно объявленных) имя программы и соответствующее устройство ее хранения;
- 4) активизировать сигнала **SPGCOM**.

В обоих случаях, если запрос принимается, система сообщает об этом ПЛ сигналом **ACKSPG (I05K30)** (положительный результат операции) или же сигналом **NKSPG (I05K31)** (неконгруэнтный запрос на выбор программы).

После выполнения выбора программы система выполняет «СБРОС» («**RESET**»).

21.2.3. Запрос на «СТОП» программы

От ПЛ можно осуществить запрос на «СТОП» двумя способами:

- 1) запросом на **HOLD** с выходом из состояния «СТОП» по разрешению оператора, осуществляемому установкой сигнала **HLDR (U10K2)**;
- 2) запросом на **HOLD** с автоматическим выходом из состояния «СТОП», осуществляемым установкой сигнала **RHOE (U10K3)**.

21.2.4. Запрос на «ПУСК» программы

ПЛ может запросить выполнение предварительно выбранной программы (посредством **SPG**) при помощи сигнала **CYST (U10K4)**.

Запрос принимается, когда УЧПУ находится в состоянии **STAND-BY**.

21.2.5. Коммутация шпинделя

Из ПЛ можно управлять параллельно несколькими осями шпинделей, имена и параметры перемещения которых были предварительно определены в файле **AXCFIL** при конфигурации системы.

Для запроса коммутации шпинделя необходимо объявить в слове **W11K3** имя нового шпинделя (в коде **ASCII**), которым надо управлять. Если этот запрос принимается, система переходит в состояние **WAIT** («Ожидание») на 2 цикла ПЛ и устанавливает сигнал **ACKCM=0 (I00K29)**. Заменяемый шпиндель остаётся обслуживаемым согласно тому предписанию, которое он имел ранее. Заменяющий шпиндель обслуживается запросами на обновление аналогового сигнала шпинделя (сигналы **ROMAO, ROMAА, FOMAO** и т.д.) только тогда, когда сигнал **ACKCM** установится в состояние «1» (новый шпиндель обслуживается).

Если запрос на коммутацию шпинделя не был принят, интерфейс отвечает сигналом **NCKCM (I00K30)**.

При включении, если слово **W11K3=0**, будет обслуживаться первый из объявленных при конфигурации системы шпиндель.

В случае сброса (**RESET**) все коммутируемые шпиндели отключаются от управления.

22. АКТИВИЗАЦИЯ И ИСПОЛЬЗОВАНИЕ КОМАНДЫ «СТОП»

Команда «СТОП» выполняется особым образом. Действие клавиши «СТОП» на пульте не может быть запрещено посредством слова **W15K0**.

Установка **HLDR=1** позволяет системе войти в состояние **HOLD**. Для выхода из этого состояния необходимо сбросить сигнал **HLDR**, и только после этого оператор нажатием клавиши «СТОП» может ввести систему в нормальное рабочее состояние. До тех пор пока **HLDR=1** действие клавиши «СТОП» запрещено во избежание возможного непредусмотренного выхода из состояния блокировки.

Установка **RHOE=1** позволяет системе войти в состояние «СТОП». Выход из этого состояния осуществляется автоматически после сброса этого сигнала. Работа не возобновится, если при установке **RHOE=1** система уже находится в состоянии **HOLD**, или же, если в то время, когда система находится в состоянии **HOLD**, был осуществлен запрос установкой **HLDR=1**. Вышесказанное означает, что возобновление работы будет выполняться только тогда, когда наряду с активизацией сигнала **RHOE** не возникает запроса на **HOLD** с высшим приоритетом – посредством **HLDR**.

В состоянии **HOLD** сигнал **HOLDA (I00K28)** будет установлен в «1».

23. АКТИВИЗАЦИЯ И ИСПОЛЬЗОВАНИЕ КОМАНДЫ «ПУСК»

Команда «**ПУСК**» имеет особенности с точки зрения её использования. Сигнал **CYST (U10K4)** (запрос на «**ПУСК**») проверяется системой даже в том случае, когда нулевой бит **W15K0** не является активным. Этот бит, таким образом, имеет только одно назначение – запрещение действия клавиши «**ПУСК**». Что касается управления сигналом **CYST**, устанавливаемого ПЛ, то система выполняет активизацию этого сигнала, исполняющего роль синхронизатора с системой, и сообщает об этом установкой сигнала **АСКСУ (I04K30)**. Т.е. запрос на «**ПУСК**» (**CYST**) может быть активизирован только тогда, когда **АСКСУ=0**, и должен быть сброшен, когда система устанавливает **АСКСУ=1**.

24. СИГНАЛЫ ИНТЕРФЕЙСА PLC

24.1. Сигналы пакета «К»

24.1.1. Разъём I00K (26K-52K-78K-104K)

- I0K0 (EMERG)** – при аварии (если есть какая-либо причина для выключения осей) УЧПУ устанавливает сигнал **EMERG=1**, устанавливает аппаратным способом сигнал **SPEPN=0** для выключения вспомогательных цепей, сбрасывает сигнал **CONP=0** и сообщает ПЛ в пакет «К» посредством слова **W06K3** о типе аварии (первые 4 бита) и об оси, для которой выявлена авария (последние 4 бита). Сигнал **EMERG** сбрасывается, когда ПЛ выдаёт системе сигнал **MUSPE=0** (авария отработана).
- I0K1 (RESE)** – сигнал, являющийся импульсным, с длительностью, равной двум циклам ПЛ, реализует сброс для ПЛ. Сигнал появляется вследствие сброса:
- 1) запроса от оператора;
 - 2) по запросу от ПЛ;
 - 3) после выключения станка (**MUSPE=1**);
 - 4) после включения станка.
- I0K2 (CONP)** – сигнал устанавливается в «1» в момент включения станка, когда сигнал **MUSPE** из состояния «1» переходит в состояние «0», для указания на правильную инициализацию УЧПУ. После установки сигнала **CONP=1** УЧПУ начинает обслуживать оси, активизируя все защиты станка. Сигнал сбрасывается в «0» после какой-либо аварии или выключения станка, когда будет активизирован сигнал **MUSPE=1** («Станок выключен»).
- I0K3 (CYCLE)** – сигнал информирует ПЛ о том, что система обрабатывает кадр программы или кадр, введенный с клавиатуры, или находится в состоянии обработки вспомогательных функций начала перемещения, или же выполняется перемещение осей (за исключением ручных перемещений).
- I0K4 (STABY)** – сигнал выдает на интерфейс информацию о том, что оси неподвижны в позиции. Сигнал сбрасывается в «0» каждый раз, когда оси перемещаются (также и в ручном режиме). Во время перемещений при функциях **G00** и **G29** этот сигнал меняет свое значение в начале и в конце перемещения. При **G28** и **G27** этот сигнал постоянно находится на логическом уровне «0».
- Диаграмма изменения сигналов **CYCLE** и **STABY** представлена на рисунке 24.1.

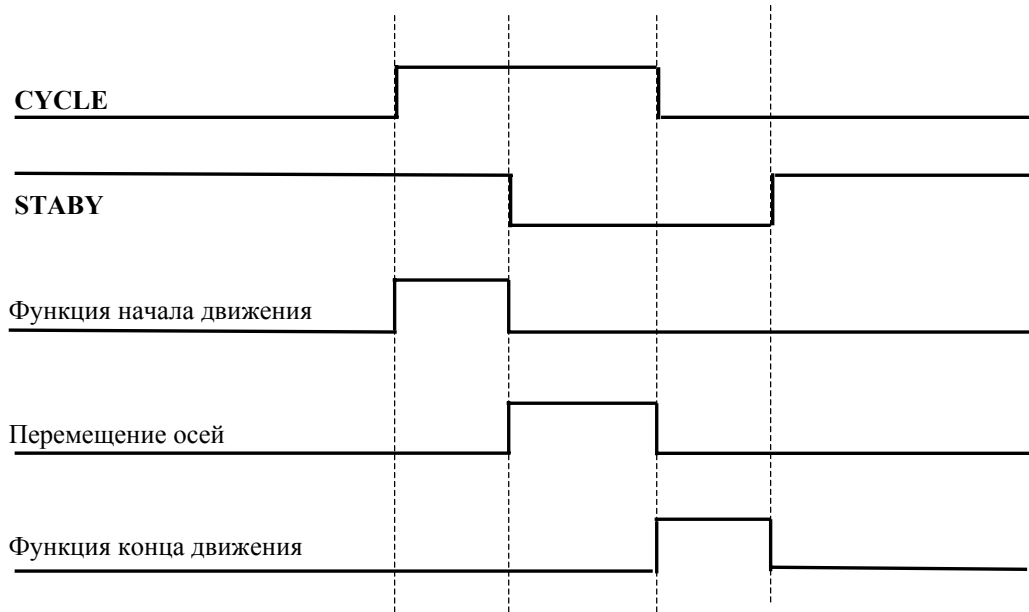


Рисунок 24.1 - Диаграмма изменения сигналов CYCLE и STABY

Согласно выше приведённой диаграмме следует, что для того, чтобы знать, когда происходит перемещение осей, ПЛ должна учитывать комбинацию сигналов **CYCLE** и **STABY**. Точнее, перемещение осей осуществляется, когда **CYCLE=1** и **STABY=0**. Во всех остальных случаях система будет находиться в фазе отработки функций начала и конца движения.

I0K6 (RCM) - сигнал устанавливается интерфейсом, когда производится запомненный поиск, и сбрасывается в «0» после того, как оси вновь возвращены на профиль; для возобновления выполнения программы следует нажать клавишу «ПУСК» (подробнее см. «Руководство оператора»).

I0K7 (CYON) - сигнал сообщает о состоянии включенного режима «ПУСК». Длительность сигнала соответствует времени включения светодиода кнопки «ПУСК».

W0K1 (ABIj, j=1-6) - в этом слове устанавливаются в «1» все биты, соответствующие осям, которым разрешены движения по запросу от ПЛ посредством **RABIj** ($j=1-6$). Существует четкое соответствие битов слова **W0K1** порядку записи имен осей в интерполяторе в файле **AXCFIL**. Например, первой оси в интерполяторе соответствует бит **I00K8**, второй - бит **I00K9** и т.д.

W0K2 (MOVj, j=1-6) - сигналы сообщают, если они находятся на логическом уровне «1», о том, что УЧПУ собирается выполнить перемещение осей; они остаются на этом уровне все время при перемещении осей, пока оси не будут позиционированы. Устанавливаются в «1» сигналы **MOVj** ($j=1-6$) после

отработки вспомогательных функций начала перемещения, а в кадре с перемещением по **G27, G28** – в начале кадра.

I0K24 (POSIA) – сигнал информирует ПЛ о том, что запись файла **FILMAS** была выполнена. После установки **POSIA** УЧПУ готово выполнять другие запросы ПЛ на перемещения. Данный сигнал остается в состоянии «1» до исчезновения запроса или же до появления другого запроса.

I0K25 (POSIM) – сигнал, если он находится на логическом уровне «1», информирует ПЛ о факте выполнения позиционирования шпинделя. Сигнал активизируется тогда, когда с датчика положения получен сигнал о том, что шпиндель вошел в допуск позиционирования, заданный в файле **AXCFIL** (для выполнения позиционирования необходим сигнал **ANGOM=1**). Сигнал **POSIM** сбрасывается после исчезновения сигнала **ANGOM**. Если перед сбросом запроса шпиндель выходит за пределы допуска, система стремится вернуть его в позицию, если это не удастся сделать, даётся сигнал об аварии.

I0K26 (NACKTO) – сигнал информирует ПЛ о том, что запрос на «обновление» инструмента не был принят. Такая ситуация может обуславливаться следующими причинами:

- 1) не существует файл корректоров инструмента;
- 2) не существует корректор;
- 3) корректор за пределами допуска;
- 4) ошибка логического вх./вых.;
- 5) запрограммирован инструмент, срок службы которого превышен;
- 6) инструмент в шпинделе не подтвержден.

После обнаружения такой ситуации интерфейс выдает ПЛ сигнал на «СБРОС».

I0K27 (ACKTO) – сигнал информирует ПЛ о том, что было выполнено обновление инструмента в шпинделе и присвоенного ему корректора.

I0K28 (HOLDA) – сигнал (если находится на логическом уровне «1»), информирует ПЛ о состоянии **HOLD** системы УЧПУ. Этот сигнал появляется после запроса на «СТОП» в конце возможного торможения перемещений по осям и длится до выхода системы из состояния **HOLD**, согласно запросу оператора. Запрос на выход из **HOLD** может выполняться путём нажатия клавиши «СТОП» на пульте управления или от ПЛ путем сброса сигналов **HLDR** или **RHOE**. Для выхода из состояния **HOLD** необходимо условие **COMU=1**.

I0K29 (ACKCM) – сигнал информирует ПЛ о том, что запрос на переключение оси шпинделя на другой шпиндель был выполнен.

I0K30 (NCKCM) - сигнал информирует ПЛ о том, что запрос на переключение оси шпинделя на другой шпиндель не был принят, т.к. запрос является не конгруэнтным (несовместимым) с текущим состоянием системы.

I0K31 (DIRMN) - сигнал информирует ПЛ о том, что выполняется перемещение оси в ручном режиме с отрицательным направлением.

24.1.2. Разъём I01K (27K-53K-79K-105K)

W01K0-W01K3 - информация этого разъёма передает в ПЛ позицию индексной оси, задаваемую в двоично-десятичном коде (BCD). Формат записи: **5.3** (5 цифр перед десятичной точкой и 3 цифры - после неё), т.е. можно запрограммировать перемещение оси от тысячных долей градусов до десятков тысяч. Например: B58345.637.

24.1.3. Разъём I02K (28K-54K-80K-106K)

W02K0, W02K1 - функция «Т» (номер инструмента, который требуется найти), заданная в двоично-десятичном коде BCD (из таблицы **RANDOM**). В этих двух словах записывается номер (единицы, десятки, сотни и тысячи) того гнезда, где находится запрограммированный или альтернативный ему инструмент. В случае смены инструмента **RANDOM** ПЛ передаётся номер гнезда, где находится запрограммированный инструмент, но можно узнать (для дальнейшей обработки) действительно запрограммированную функцию «Т» в слове **W04K0-W04K1**.

Примечание - Способ управления сменой инструмента **RANDOM** и возможность контроля срока службы инструмента устанавливается на фазе инициализации, если они были объявлены в файле **PGCFIL** в инструкции **FIL**.

W02K2, W02K3 - в этих двух словах системой передается в ПЛ функция «Т» инструмента, который действительно пойдет в шпиндель. В действительности же, при управлении сроком жизни инструмента не всегда запрограммированный инструмент пойдет в шпиндель, он может быть заменён (в соответствии с таблицей) на возможный альтернативный инструмент. Когда группа инструментов одинакового назначения закончена (износ инструментов), система выдает сигнал **FUTKO**.

24.1.4. Разъём I03K (29K-55K-81K-107K)

W03K0 - функции «М» импульсные. Это слово используется для передачи в ПЛ функций «М» в двоично-десятичном коде. Код BCD записывается в слово при условии, если сигнал **CEFA** на логическом уровне «1». Код сохраняется в слове в течение двух циклов «медленной» логики (минимально). Функции этого типа могут выдаваться в начале или в конце перемещения в зависимости от характеристики их в файле **I0CFIL**.

W03K1 - функции «**M**» немедленного действия. Данное слово используется для передачи в ПЛ функций «**M**» немедленного действия, в двоично-десятичном коде (**BCD**). Код **BCD** записывается в слово в начале перемещения осей (если в кадре запрограммирована такая функция «**M**») и держится весь интервал времени, когда **MOVn=1** и **COMU=1**. Обработка функции «**M**» данного типа не зависит от состояния сигнала **CEFA**. Такие функции «**M**» могут программироваться при функциях **G28**, **G27**. Тип функции «**M**» немедленного действия задается при характеристизации в файле **IOSFIL**.

Примечание - функции «**M**» немедленного действия задаются только в кадрах с перемещением.

24.1.5. Разъём I04K (30K-56K-82K-108K)

W04K0, W04K1 - эти слова используются для передачи в ПЛ запрограммированной функции «**T**» в коде **BCD**. Код **BCD** записывается в этих двух словах при наличии сигнала **CEFA** и запоминается до программирования очередной функции «**T**» или сброса. Функция «**T**» всегда является функцией начала перемещения.

I04K16 (FUAS) - строб функции «**S**», устанавливается в «1» каждый раз, когда задаётся новая «**S**»-функция, и имеет длительность в два цикла «медленной» логики.

I04K17 (FUAT) - строб функции «**T**», устанавливается в «1» каждый раз, когда задается новая «**T**»-функция и имеет длительность в два цикла «медленной» логики.

I04K18 (FUAM) - строб функции «**M**», устанавливается в «1» каждый раз, когда задается новая «**M**»-функция, и имеет длительность в два цикла «медленной» логики.

I04K20 (TASC1) - строб функции «**Индексная ось 1**», устанавливается в «1» каждый раз, когда задается новая функция первой индексной оси, и имеет длительность в два цикла «медленной» логики.

I04K21 (TASC2) - строб функции «**Индексная ось 2**», устанавливается в «1» каждый раз, когда задается новая функция второй индексной оси, и имеет длительность в два цикла «медленной» логики.

I04K22 (TASC3) - строб функции «**Индексная ось 3**», устанавливается в «1» каждый раз, когда задаётся новая функция третьей индексной оси, и имеет длительность в два цикла «медленной» логики.

I04K23 (FUTKO) - строб функции «**T**» для инструмента с истекшим сроком службы. Этот сигнал активизируется, только если в ха-

рактеризации заявлен файл срока службы инструмента, и означает, что запрограммированный инструмент принадлежит к группе инструментов, все члены которой сломаны (**A**) или не пригодны к использованию (**D**) (см. файл срока службы инструментов). В этом случае система информирует ПЛ о запрограммированной **T**-функции не установкой сигнала **FUAT**, а установкой в «1» строб-сигнала **FUTKO** на два цикла «медленной» логики.

ВНИМАНИЕ! СИСТЕМА АКТИВИЗИРУЕТ СТРОБ **FUTKO** ТОЛЬКО ТОГДА, КОГДА РАВЕН НУЛЮ АЛЬТЕРНАТИВНЫЙ ИНСТРУМЕНТ В СТРОКЕ ФАЙЛА СРОКА СЛУЖБЫ ИНСТРУМЕНТОВ, КОТОРАЯ СООТВЕТСТВУЕТ ЗАПРОГРАММИРОВАННОМУ ИЛИ ВЫБРАННОМУ ИЗ ЕГО ГРУППЫ ИНСТРУМЕНТУ.

Если группа альтернативных инструментов к запрограммированному инструменту замкнута сама на себя, то система не устанавливает ни **FUTKO**, ни **FUAT**, и выдаёт сообщение о блокирующей ошибке «Сообщение_4 72».

- I04K24 (SESC)** – сигнал активизируется при программировании функции «индексной оси», если запрограммированный знак направления движения – отрицательный. В этом активном состоянии сигнал остаётся до программирования другого перемещения или до сброса.
- I04K25 (MPROFI)** – сигнал устанавливается, когда оси перемещаются по профилю при **G1, G2, G3**.
- I04K26 (CUMAN)** – сигнал устанавливается в «1» при управлении сменой инструмента **RANDOM**, когда программируется функция «**T**» инструмента, не существующего в таблице **RANDOM**, либо запрограммирована функция «**T**» при заполненной таблице **RANDOM**. Этот сигнал остаётся в таком состоянии до программирования очередной функции «**T**».
- I04K27 (PWMAN)** – сигнал устанавливается в «1» на всё время продолжительности выполнения запроса вращения шпинделя. Устанавливается для всех сигналов, которые запрашивают вращение шпинделя без условия, что шпиндель действительно вращается (пример: **ROMAO/ROMAA, FOMAO/FOMAA, ANGOM**). Этот сигнал действителен только для шпинделя с датчиком и с двигателем постоянного тока.
- I04K28 (SOFIT)** – сигнал появляется во время цикла измерения: контроль диаметра отверстия, пересчёт начальной точки при перемещении быстрого хода во время подвода к детали.
- I04K29 (GOMAN)** – сигнал устанавливается в «1» в тот момент, когда выполнен запрос на вращение шпинделя, а количество оборотов шпинделя превышает значение, объявленное в инструкции **POM** файла **AXCFIL**, и сохраняется до момента остановки вращения шпинделя с количеством оборотов, меньше числа, указанного в инструкции **POM**. Этот сигнал

действителен только для шпинделя с датчиком и с двигателем постоянного тока.

I04K30 (АСКСУ) - сигнал устанавливается в «1» интерфейсом **PLC** для того, чтобы сообщить ПЛ о том, что запрос на перемещение осей, задаваемого оператором с пульта, управляемого от ПЛ, был принят и выполнен. Сигнал устанавливается в «1» по окончанию перемещения.

I04K31 (НСКСУ) - сигнал сообщает ПЛ о том, что запрос на перемещение осей, задаваемое оператором с пульта, управляемого от ПЛ, не был принят из-за несоответствия состоянию системы.

24.1.6. Разъём I05K (31K-57K-83K-109K)

I05K0 (GOO) - система установкой этого сигнала сообщает в ПЛ, что УЧПУ находится в состоянии отработки **GO**. Этот сигнал устанавливается и по выключению УЧПУ. Сбрасывается он наличием перемещения при **G1, G2, G3** и заново устанавливается, если программируется быстрое перемещение или же сброс.

I05K1 (PROFI) - система посредством этого сигнала сообщает в ПЛ, что оси находятся на профиле. Этот сигнал устанавливается в «1», когда УЧПУ находится в состоянии отработки **G1, G2** или **G3**, и остается в этом состоянии до тех пор, пока не выполняется движение при **G0**.

I05K2 (MISU) - система посредством этого сигнала сообщает в ПЛ, что УЧПУ находится в отработке стандартного цикла измерения (**G72, G73**). Этот сигнал устанавливается в «1», когда системе поступает запрос на отработку цикла измерения щупом.

I05K3 (INTUT) - система посредством этого сигнала сообщает в ПЛ, что УЧПУ выполняет цикл проверки целостности инструмента (**G74**). Этот сигнал устанавливается в «1», когда системе поступает запрос на проверку целостности инструмента.

I05K4 (FG8.1)
I05K5 (FG8.2)
I05K6 (FG8.4)
I05K7 (FG8.8) } - эти четыре бита идентифицируют текущий стандартный цикл обработки отверстия (**G81-G89**).

W05K1 (ASRI_j, j=1-6) - эти 6 сигналов устанавливаются в «1» после выполнения цикла поиска микропуля соответствующей оси.

- I05K16 (ROPO1)**
I05K17 (ROPO2)
I05K18 (RONE1)
I05K19 (RONE2) } - эти сигналы устанавливаются в «1» при появлении запроса на перемещение осей «от точки к точке». Эти сигналы используются в том случае, если для управления осями не используются двигатели постоянного тока. Для позиционирования необходимо знать направление движения.
- I05K20 (ROLE1)**
I05K21 (ROLE2)
I05K22 (ROLLE1)
I05K23 (ROLLE2) } - сигналы устанавливаются в «1» во время перемещения соответствующей оси «от точки к точке» (первой или второй) при достижении осью координат точки первого и второго замедления соответственно, установленных при характеристизации в файле **IOCFIL**. Данные сигналы используются в случае, если ось управляется двигателем не постоянного тока.
- I05K24 (POSI1)**
I05K25 (POSI2) } - сигналы устанавливаются в «1», когда соответствующая ось «от точки к точке» входит в пределы допуска позиционирования, и остаются в этом состоянии до тех пор, пока действует запрос. Если при активном запросе ось выходит из допуска, система пытается вернуть ее в пределы допуска, и, если попытка не удастся, выдаёт **Сообщение_4_64 «Ось вне допуска»**. Сигнал **POSI** всё это время остаётся в состоянии логической «1».
- I05K26 (BUSY1)**
I05K27 (BUSY2) } - сигналы устанавливаются в «1», когда активизируется запрос на перемещение оси «от точки к точке» в канале 1 (**BISY1**) или 2 (**BISY2**). Они остаются в этом состоянии до тех пор, пока не будет сброшен запрос.
- I05K28 (KOSI1)**
I05K29 (KOSI2) } - сигналы **KOSI1** или **KOSI2** активизируются, когда канал 1 или 2 занят выполнением текущего запроса на позиционирование оси «от точки к точке» в канале 1 (**KOSI1**) или 2 (**KOSI2**), и в этот же канал поступает другой такой же запрос.
- I05K30 (ACKSPG)** - сигнал (когда находится в активном состоянии) информирует ПЛ о том, что предыдущий запрос на выбор программы или команды, введённой с клавиатуры, был принят. Сигнал остается в состоянии «1» до тех пор, пока не будет сброшен запрос **SPG**.
- I05K31 (NACKSPG)** - сигнал (когда находится в активном состоянии) информирует ПЛ о том, что предыдущий запрос на выбор программы или команды, введённой с клавиатуры, не был принят, т.к. был сделан неправильно или его невозможно выполнить. Сигнал остается в активном состоянии до тех пор, пока не будет сброшен запрос **SPG**.

24.1.7. Разъём I06K (32K-58K-84K-110K)

- W06K0 (COMj, j=1-6)** - в этом слове устанавливаются в «1» разряды (биты), соответствующие осям, для которых было запрошено переключение посредством ПЛ. На фазе характеристизации (в

файле **AXCFIL**) необходимо выполнить тарировки для взаимно переключаемых осей. После запроса на переключение оси для отключенной оси выполняется только контроль на сбой привода.

- I06K16-I06K19 (SGAMj, j=1-4)** } - сигналы служат для передачи в ПЛ информации о диапазоне вращения шпинделя, устанавливаются в «1» в зависимости от запрограммированной функции и инструкции **GMn**, установленной при характеристизации в файле **AXCFIL (GM1-GM4)**. ПЛ должна сообщить системе посредством сигналов **GAMj (j=1-4)** о том, какой диапазон подключён на станке. Обновление аналогового сигнала шпинделя, функции «S», переданной в коде BCD в слова **W08K0, W08K1, W08K2**, и строба **FUAS** при смене диапазона выполняется в течение 2-х циклов ПЛ после обновления сигналов **SGAMj**. Если имеется только один диапазон вращения шпинделя, ПЛ должна устанавливать бит соответственно диапазону 1 (**GAMMA1**). Сигналы **SGAMj** и значение байта **W6N1** обновляются параллельно и одновременно
- I06K20 (RETRA)** - бит устанавливается в «1» при первом перемещении осей в состоянии **MBR=1** и остается в положении «1» до тех пор, пока (в **MBR=0**) не возвратится к кадру программы, в котором было активизировано **MBR=1**. Точнее сказать, бит **I06K20** остаётся в состоянии «1» для всех кадров программы, которые выполняются повторно.
- I06K21 (RSPEPN)** - сигнал сообщает в ПЛ о готовности УЧПУ к подаче управляющего напряжения на станок («1» - УЧПУ готово; «0» - УЧПУ не готово). ПЛ должна ответить о готовности станка к подаче напряжения установкой в «1» сигнала **U10K20**;
- I06K22 (MODEC)** - сигнал сообщает в ПЛ об активности режима «КОМАНДА» («0» - режим «КОМАНДА» активен; «1» - не активен). Сигнал может анализироваться ПЛ для активизации/деактивизации кнопок «F11»-«F18» в режиме «КОМАНДА», когда мнемоника клавиш не видна. Сигнал устанавливается только в УЧПУ NC-200/201/201M/202/210/220/230/310.
- I06K23 (ABIM)** - сигнал служит для обеспечения синхронизации при выполнении коммутации шпинделей, когда для управления используется только один аналоговый канал. Если **ABIM=1**, это говорит о том, что канал занят. Устанавливается на логический уровень «1», когда появляются запросы на вращение шпинделя (сигналы **ROMAO, ROMAА, FOMAO, FOMAА, ANGOM**). Данный сигнал сбрасывается в «0», когда запросы сняты или отработаны.
- W06K3** - в слове закодирован в двоичном коде тип выявляемой аварии и номер оси, для которой была выявлена авария. В первые четыре бита записывается номер аварийной оси (от 1 до 8). Эти 4 бита передаются ПЛ, когда происхо-

дит авария, выявленная системой или ПЛ. Они сбрасываются, когда сигнал **MUSPE** устанавливается в «0».

В последние четыре бита заносится код аварии. Существует 5 типов аварий и соответствующих им кодов:

- 1) (0001)**1** - выключение, запрошенное от ПЛ;
- 2) (0010)**2** - выключение по сервоошибке (**SERVOERROR**);
- 3) (0011)**3** - ошибка скорости (**ERROR**);
- 4) (0101)**5** - аномалия датчика положения;
- 5) (0110)**6** - ось вне допуска.

24.1.8. Разъём I07K (33K-59K-85K-111K)

W07K2, W07K3 - в этих словах ПЛ передается номер корректора (4 цифры), присвоенный запрограммированному или альтернативному ему инструменту в случае, если первый изношен или сломан. Эти цифры сохраняются до осуществления обновления инструмента.

24.1.9. Разъём I08K (34K-60K-86K-112K)

W08K0, W08K1, W08K2 - в этих трёх словах в ПЛ передаётся запрограммированная функция «S» в двоично-десятичном коде BCD. Эти цифры фиксируются до программирования следующей функции «S» или до сброса (**RESET**). Для шпинделя без преобразователя значение функции «S» будет зависеть от процента, установленного **КОРРЕКТОРОМ СКОРОСТИ ВРАЩЕНИЯ ШПИНДЕЛЯ «S»**. Максимальное программируемое значение равно 9999.

W08K3 - в этом слове в ПЛ сообщается информация о режиме работы УЧПУ (сигналы **EMDI, AUTO, SEMI, MANUC, MANUJ, RIPRO, RIMZE, RESET**).

24.1.10. Разъём I09K (35K-61K-87K-113K)

I09K19 (TACKN) - этим сигналом ПрО сообщает в ПЛ о состоянии запроса включения/выключения режима обучения сигналом **TEACH (U13K1)**:

- **I09K19 (TACKN)=1** - режим обучения включён, и файл **TEACH** открыт;
- **I09K19 (TACKN)=0** - режим обучения выключен, и файл **TEACH** закрыт.

I09K20 (TPOSS) - этим сигналом ПрО сообщает в ПЛ о состоянии запроса сохранения позиций перемещённых осей **PSTOR (U13K2)** в файле **TEACH** в режиме обучения, когда **I09K19 (TACKN)=1**:

- **I09K20 (TPOSS)=1** - позиция сохранена;
- **I09K20 (TPOSS)=0** - позиция не сохранена.

- I09K21 (TBLOS)** - этим сигналом ПрО сообщает в ПЛ о состоянии запроса сохранения выполненного кадра **BSTOR (U13K3)** в файле **TEACH** в режиме обучения, когда **I09K19 (TACKN)=1**:
- **I09K21 (TBLOS)=1** - кадр сохранён;
 - **I09K21 (TBLOS)=0** - кадр не сохранён.
- I09K22 (TPROB)** - этим сигналом ПрО сообщает в ПЛ состояние выполнения запросов. Сигнал **TPROB** устанавливается в состояние «1», если:
- произошла ошибка открытия файла по запросу **TEACH (U13K1)**;
 - произошла ошибка записи кадра в файле **TEACH** по запросу **PSTOR (U13K2)**;
 - произошла ошибка записи кадра в файле **TEACH** по запросу **BSTOR (U13K3)**;
 - очередной номер кадра **N**, формируемый в файле **TEACH**, больше **9999**;
 - с переменной **CBT** введён недопустимый номер кадра, например, больше **9999**.
- W09K1 (BILA_j, j=1-6)** - в этом слове в ПЛ передаётся информация о направлении перемещения по всем осям. Эта информация предоставляется в реальном масштабе времени для того, чтобы можно было сбалансировать с помощью гидравлики одну или несколько из возможных 8 осей.
- I09K24 (INVER)** - сигнал активен только тогда, когда система выполняет функцию **G84** (только для шпинделя с двигателем переменного тока). Устанавливается в «1» до достижения конечной точки нарезания резьбы метчиком для того, чтобы ПЛ имела возможность изменить направление вращения шпинделя. Это опережение зависит от значения, которое табулируется в параметре «**время изменения направления вращения шпинделя**» с помощью кода **TSM**.
- I09K25 (STOPR)** - сигнал активен только тогда, когда система выполняет функцию **G86** (только для шпинделя с двигателем переменного тока). Устанавливается в «1» при достижении конечной точки резьбы для того, чтобы ПЛ имела возможность остановить вращение шпинделя.
- I09K26 (FG96)** - этот сигнал устанавливается от ПО:
- 1) в состояние «1», если при активной функции **G96** задана скорость резания командой **S<значение>**;
 - 2) в состояние «0», если при активной функции **G97** задана скорость вращения шпинделя командой **S<значение>**.

Пример

Кадр	Состояние I09K26
G97S400M3	0
SSL=500	0
GX100	0
G96	0
S40M3	1
G1X0F60	1
GX100	1
G97	1
S400	0

24.1.11. Разъём U10K (36К-62К-88К-114К)

- U10K0 (MUSPE)** – сигнал устанавливается ПЛ после обнаружения какой-либо аварийной ситуации для того, чтобы интерфейс сбросил сигнал **CONP** и, по возможности, **EMERG**. Вышеуказанный сигнал может быть установлен равным нулю только после включения питания вспомогательных механизмов станка.
- U10K1 (REAZ)** – сигнал служит для запроса выполнения сброса, запрос асинхронного типа и длится один цикл ПЛ (см. п.13.3).
- U10K2 (HLDR)** – запрос на «**ПРИОСТАНОВ**» с перезапуском по разрешению. При активизации этого запроса система останавливает перемещение осей, после чего сообщает об этом ПЛ сигналом **HOLDA**. Для выхода из состояния **HOLD** оператору следует нажать клавишу «**СТОП**», дать разрешение на перемещение осей установкой **COMU=1**, сбросить запрос и запросить «**ПУСК**».
- U10K3 (RHOE)** – запрос на «**ПРИОСТАНОВ**» с автоматическим перезапуском. При активизации этого запроса система останавливает перемещение осей, после чего сообщает об этом ПЛ сигналом **HOLDA**. Для выхода из состояния **HOLD** необходимо, чтобы **COMU=1**, и был сброшен запрос **RHOE**, при этом перемещение осей будет автоматически продолжено.
- U10K4 (CYST)** – при помощи сигнала **CYST** (**U10K4=1**) ПЛ может запрашивать запуск выполнения программы, выбранной ранее посредством **SPG**, или движение в ручных перемещениях по выбранной оси. Запрос принимается системой в состоянии **STAND-BY**. Выполненное перемещение заканчивается сигналом **ACKCY=1** (**I04K30**). Сигнал **CYST** должен устанавливаться в «1», когда **I04K30=0**.
- U10K5 (FOLD)** – сигнал прерывает любое перемещение осей, кроме резбонарезания и нарезания резьбы метчиком в процессе их выполнения. Когда сигнал сбрасывается, перемещение осей возобновляется с ранее установленным заданием.
- ВНИМАНИЕ!** ЕСЛИ ВО ВРЕМЯ ЗАПРОСА **FOLD** ДАЁТСЯ «**СБРОС**», ТО НА ВРЕМЯ ВЫПОЛНЕНИЯ СБРОСА ЗАПРОС **FOLD** ДОЛЖЕН БЫТЬ УСТАНОВЛЕН В НУЛЬ.

- U10K6 (WAIC)** - сигнал учитывается только в том случае, когда активизирована какая-либо команда **MOV**, и соответствующий сигнал **ABI** находится в состоянии «1». В этом случае, если **WAIC=1**, интерфейс сразу же проверяет наличие **COMU=1** и обслуживает ось. Если **WAIC=0**, то система ждёт в течение двух циклов ПЛ и затем проверяет наличие **COMU=1**. Во время выполнения стандартного цикла **G84** сигнал **WAIC** должен быть равен нулю. Сигнал не применяется для осей, имеющих блокировочные устройства.
- U10K7 (RISPE)** - сигнал запроса, активизируется программой логики станка, когда необходимо выключить станок ввиду выявленных ею аномальных ситуаций. При этом после этого запроса автоматически отключает реле готовности УЧПУ («**SPEPN**») для выключения вспомогательных цепей станка. Для повторного разрешения включения станка необходимо сбросить данный запрос. Реле «**SPEPN**» включается сигналом **U10K20=1**.
- W10K1 (RAB1j, j=1-8)** - в этом слове ПЛ сообщает системе, для каких координатных осей процесса необходимо включить следящий режим, устанавливая соответствующие разряды (биты) в «1». Если этот запрос не выполняется, оси остаются работать не в следящем режиме. Необходимо, чтобы запрос на разрешение движения осей выполнялся бы после установления сигнала **CONP**.
- U10K16 (REGTOL)** - сигнал запрашивает у системы сброс визуализации и инструмента, находящегося в шпинделе, при управлении сменой инструмента типа **RANDOM** и просто сброс визуализации инструмента в шпинделе при обычном управлении.
- U10K17 (DITVI)** - сигнал «замораживает» счет срока службы инструмента в течение всего интервала, когда этот сигнал запроса остается в состоянии «1». Этот сигнал принимается только перед запросом на перемещение осей.
- U10K18 (SPGCOM)** - запрос активизируется в случае, если необходимо выполнить выбор программы из файла **FILCMD** для последующей отработки или установку команды, введенной с клавиатуры. Если интерфейс принимает этот запрос, то отвечает сигналом **ACKSPG** и выдает сигнал сброса. Если запрос не принимается, то интерфейс отвечает сигналом **NCKSPG** и удерживается до исчезновения запроса. Для более полной информации см. сигнал **U10K23**.
- U10K19 (CMDLOG)** - сигнал должен устанавливаться ПЛ в «1», когда в файле **FILCMD** или в разъеме пакета «**K**» присутствует какая-либо команда, которая вводится с клавиатуры нажатием клавиши «**ENTER**», но не выбор управляющей программы. Для более полной информации см. сигнал **U10K23**.

U10K20 (ASPEPN) - сигнал необходимо устанавливать в «1» по готовности станка для подачи управляющего напряжения. Он является запросом базовому Про для включения им реле «**SPEPN**» в блоке питания УЧПУ. Сигнал **U10K20** должен учитывать состояние сигнала **I06K21**, который показывает готовность УЧПУ к включению станка. При аварийной ситуации на станке сигнал должен быть сброшен в «0»;

$$U10K20 = I06K21 * \text{«СИГНАЛ ОТ КНОПКИ ВКЛЮЧЕНИЯ УПРАВЛЯЮЩЕГО НАПРЯЖЕНИЯ СТАНКА»} + U10K20 * I06K21 / \text{«СИГНАЛ ОТ КНОПКИ ВЫКЛЮЧЕНИЯ УПРАВЛЯЮЩЕГО НАПРЯЖЕНИЯ СТАНКА»}$$

Примечание - Замкнутое положение контактов реле «**SPEPN**» в блоке питания блока управления УЧПУ соответствует состоянию сигналов **U10K20=1** и **I06K21=1**.

U10K21 (AGTOL) - сигнал запрашивает от интерфейса обновление инструмента и соответствующего ему корректора. Этот запрос может выполняться какой-либо функцией, имеющей характеристики (в файле **IOSFIL**) «**смена коррекций**» и «**блокировка вычислений**». Для выполнения функции, запрошенной сигналом **AGTOL**, необходимо, чтобы сигнал **SEFAB** был установлен в «0» на всё время, при котором сигнал **AGTOL** установлен в «1». Пример диаграммы сигналов при установке сигнала **AGTOL** во время смены инструмента представлен на рисунке 24.2.

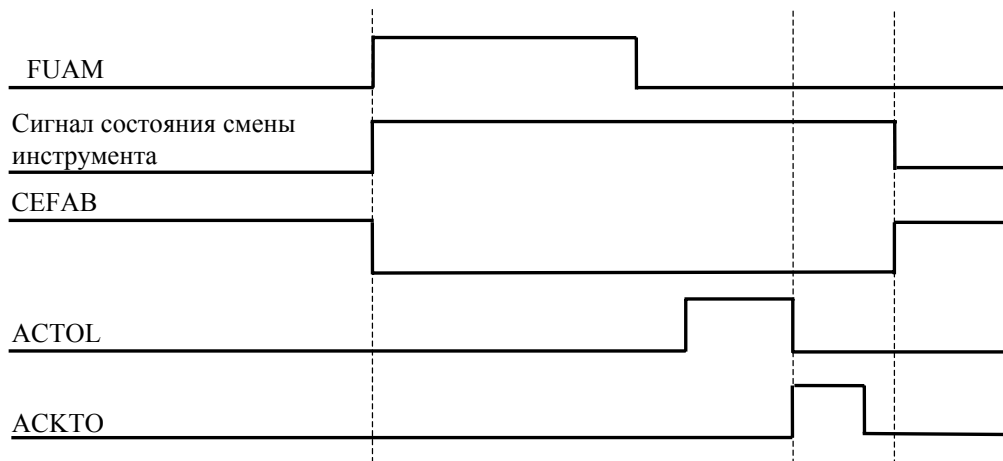


Рисунок 24.2 - Диаграмма сигналов при установке сигнала **ACTOL** во время смены инструмента

U10K23 (FILCMD) - если сигнал **FILCMD** установлен в «1», Про интерпретирует установленный код в слове **W17K0** как номер записи, предварительно установленной в файле **FILCMD**, которая должна быть выполнена.

Если сигнал **FILCMD** установлен в «0», система интерпретирует установленный код в слове **W17K0** как номер разъёма пакета «**К**», с которого записывается команда для выполнения кодами в **HEX** формате.

Пример команд, которые можно записать в пакет «**К**»:

- 1) команда, выполняемая с клавиатуры, например, **URL=1** в **ASCII** коде;
- 2) имя выбираемой УП и соответствующей ей памяти, например, **PROBA/MP2** в **ASCII** коде.

ВНИМАНИЕ! **ASCII** КОД КАЖДОЙ КОМАНДЫ ДОЛЖЕН БЫТЬ ЗАКОНЧЕН ШЕСТНАДЦАТИРИЧНЫМ КОДОМ **0AH**, СООТВЕТСТВУЮЩИМ СИМВОЛУ **LF**. ПОСЛЕ ВЫПОЛНЕНИЯ ЗАПРОСА **SPGCOM** СИСТЕМА УСТАНОВИТ СИГНАЛ **ACKSPG** В «1». ЕСЛИ ЗАПРОС НЕ БЫЛ ВЫПОЛНЕН СИСТЕМОЙ, ИЛИ БЫЛА ОБНАРУЖЕНА СИСТЕМНАЯ ОШИБКА, СИГНАЛ **ACKSPG** БУДЕТ УСТАНОВЛЕН В «0». В ЛЮБОМ СЛУЧАЕ ЛОГИКА ДОЛЖНА СБРОСИТЬ СИГНАЛ **SPGCOM** ПОСЛЕ УСТАНОВКИ **ACKSPG** ИЛИ **NCKSPG** В «1» ДО СЛЕДУЮЩЕГО ЗАПРОСА **SPGCOM**.

Если сигнал **SPGCOM** устанавливается в «0» до завершения выполнения команды, ни один из сигналов **ACKSPG** и **NCKSPG** установлен не будет.

Примеры

- 1) при активизации сигнала **SPGCOM** должна быть выполнена команда **SPG,C/MP2**, записанная **ASCII** строкой в пакет «К». Если сигнал **SPGCOM** устанавливается в «0» пока строка находится в процессе выполнения, выполнение будет доведено до конца, но сигнал **ACKSPG** не будет установлен в «1».
- 2) сигнал **SPGCOM** необходимо активизировать при выполнении серии клавиатурных команд посредством «косвенного файла».

Примеры

- файл **FILCMD** содержит запись: **\$CMDPP/MP2**;
- файл **CMDPP/MP2** содержит следующие записи:
UCV=1
USB=1
RAP=1
URL=1
SPG,PROBA/MP2

Символ «\$» указывает, что файл **CMDPP/MP2** содержит клавиатурные команды для выполнения.

Если **SPGCOM** установить в «0», пока файл **CMDPP/MP2** выполняется, то система выполнит только текущую запись. Оставшиеся записи не будут рассматриваться, пока сигнал **ACKSPG** не будет установлен в «1». При запросе **SPGCOM** система установит сигнал **NCKCOM** в следующих случаях:

- 1) в слове **w17k0** записан ноль;
- 2) нет кода символа **LF** в конце команды записанной **ASCII** строкой в пакете «К»;

- 3) **SPGCOM** был установлен в «1» и система находится в состоянии сброса;
- 4) система находится в состоянии **ERROR** или **W09K0=1** (**Сообщение 4__1**);
- 5) нет затребованной записи в файле **FILCMD**;
- 6) «**косвенный файл**» не существует или не содержит ни одной записи;
- 7) неправильный синтаксис или формат выполняемой команды.

U10K24 (COMU) - сигнал, если находится на логическом уровне «1», запрашивает разрешение на перемещение. Этот сигнал проверяется в следующих случаях:

- 1) перед каждым ручным перемещением;
- 2) перед каждым перемещением с функцией **GOO**;
- 3) перед каждым перемещением с функцией **G29**;
- 4) перед выходом из состояния **HOLD**, при сигнале **HOLDA=1**;
- 5) в первом кадре программы перед началом выполнения цикла с функцией **G28**.

Каждый раз, когда проверяется сигнал **COMU** и обнаруживается, что **COMU=0**, система не может принимать команды перемещения осей до тех пор, пока не будет **COMU=1**.

U10K25 (SEFA) - сигнал в своем активном состоянии «1», выдает системе разрешение на выдачу вспомогательных функций: «**S**», «**T**», «**индексной оси**», «**M**» (не немедленного действия).

U10K26 (SEFAB) - сигнал проверяется интерфейсом после выдачи какой-либо функции «**M**» конца движения, имеющей характеристику «**блокировка вычисления**». Он позволяет, синхронизировано выполнить запись из файла перемещений осей от системы (**FILMAS**). Когда сигнал **SEFAB=0**, возможны перемещения осей от системы и отработка вспомогательных функций в обрабатываемой записи. До тех пор пока сигнал **SEFAB=0**, система не может перейти к отработке другой информации кадра или закончить отработку текущего кадра, кроме вышеуказанной функции «**M**». Установкой **SEFAB=1** системе сообщается о выполнении функции «**M**» с блокировкой расчёта.

U10K27-U10K29
(MIZE1-MIZE3) } - сигнал (один из трёх) проверяется при программировании перемещения какой-либо «**индексной оси**» (1-ой, 2-ой, 3-ей). После получения запроса на перемещение «индексной оси» система проверяет этот сигнал и, если обнаруживает его активное состояние, выполняет полученный запрос. Если же обнаруживается, что **MIZE=0**, система проверяет, чтобы запрограммированное перемещение было равно «0», в противном случае - выдаёт ошибку на индикацию.

U10K30 (RMORE) - сигнал служит для выполнения функции запуска мультиблока, запрашиваемого от ПЛ. Сигнал функционально соответствует команде **MBR=1**, набранной с клавиатуры. Этот сигнал должен устанавливаться в «1» только после остановки перемещений по запросу **HLDР**, на что система должна ответить: «**HOLDA**». При **RMORE=1** после нажатия клавиши «**ПУСК**», или после запроса на **ПУСК**, установленного ПЛ, система выходит из состояния **HOLD** и начинает выполнение кадров технологической программы в режиме «**АВТОМАТИЧЕСКИЙ**», либо в режиме «**КАДР**». При этом система выполняет обработку максимального числа кадров, объявленных с инструкцией **MBR** в файле **PGCFIL**. Когда следует возобновить нормальное функционирование, ПЛ заново должна запросить **HOLD** с ранее установленными режимами. Когда система ответит: «**HOLDA**» - сигнал **RMORE** должен быть сброшен. С этого момента, нажимая клавишу «**ПУСК**», или запрашивая **ПУСК** от ПЛ, система выходит из состояния **HOLD** и возобновляет нормальное функционирование.

24.1.12. Разъём U11K (37K-63K-89K-115K)

U11K0 (ANGOM) - сигнал, если находится на логическом уровне «1», устанавливает запрос на ориентацию шпинделя. Аналоговый сигнал на выходе ЦАП должен соответствовать параметрам, заданным при конфигурации системы. Точка ориентации шпинделя будет выбрана системой, используя первый параметр инструкции **POM** (файл **AXCFIL**) или значение, установленное в разъёме 14 пакета «**К**». Когда датчик положения будет сигнализировать, что шпиндель находится в допуске позиционирования, система выдаст в ПЛ сигнал **POSIM=1**. После установки сигнала **POSIM** система будет ожидать сброса сигнала **ANGOM**. После этого шпиндель не будет управляться системой до получения других запросов. До тех пор пока имеется запрос на ориентацию шпинделя, сигналы **ROMAO**, **ROMAA**, **FOMAO**, **FOMAA** не будут учитываться

U11K1 (FOMAO) }
U11K2 (FOMAA) } - сигналы (если находятся в «1») устанавливают запрос на выдачу для шпинделя положительного или отрицательного аналогового сигнала в соответствии с данными, записанными в слове **W11K1**. Запрос на принудительную установку направления вращения шпинделя (**FOMAO** или **FOMAA**) имеет высший приоритет по отношению к обычным запросам на направление вращения (сигналы **ROMAO** и **ROMAA**). Эти сигналы могут быть использованы только для шпинделя с двигателем постоянного тока. Синхронизация сигнала идентична синхронизации сигнала **ROMAO**.

U11K3 (ROMAO) }
U11K4 (ROMAA) } - сигналы устанавливают запрос системе на выдачу для шпинделя положительного или отрицательного аналогового сигнала, соответствующего последнему запрограмми-

рованному значению «S» и включённому диапазону. Эти сигналы должны быть активизированы также для шпинделя с двигателем переменного тока.

U11K5 (FORID) - сигнал, если находится в активном состоянии, обеспечивает деление значения аналогового сигнала, заданного в канале шпинделя, на 10 при наличии сигнала **FOMAA** или сигнала **FOMAO**.

U11K6 (ORIMO) }
U11K7 (ORIMA) } - этими сигналами, установленными до сигнала ориентации **ANGOM**, устанавливается направление ориентации шпинделя. **ORIMO** устанавливает аналоговый сигнал при ориентации - положительным, а **ORIMA** - отрицательным. При состояниях этих сигналов, равных нулю, ориентация шпинделя по сигналу **ANGOM** выполняется по кратчайшему пути.

W11K1 - в слове задаётся значение аналогового сигнала (напряжение в вольтах и десятых долях вольта) в канале шпинделя при наличии сигнала **FOMAA** или сигнала **FOMAO**.

U11K16-U11K19 }
(GAMM1-GAMM4) } - сигналы имеют значение, только если байт **W16N1** равен нулю, и должны устанавливаться в «1» в ПЛ, когда механическое включение запрограммированного диапазона выполнено, чтобы задание на вращение шпинделя соответствовало тому, что определено при конфигурации системы. Эти сигналы должны быть активизированы также для шпинделя с двигателем переменного тока. ПЛ должна устанавливать эти сигналы в «0» только при **ABIM=0**.

U11K20 (RZDS) - сигнал активизирует чтение «смещение нуля» привода шпинделя. Процедура чтения-компенсации «смещение нуля»:

- 1) включить станок и ввести команду **SOM03**;
- 2) контролировать величину скорости вращения шпинделя по индикации;
- 3) задать **S<значение>**, равное скорости вращения шпинделя и обратное его направлению (направление выбрать функциями **M03** или **M04**), и контролировать останов вращения шпинделя или вращения в допустимых пределах;
- 4) установить сигнал **U11K20=1**;
- 5) установить сигнал **U11K21=1** через одну секунду после выполнения предыдущего действия.

Процедура чтения-компенсации «смещения нуля» привода для шпинделя с датчиком:

- 1) включить станок и ввести команду ориентации шпинделя и удерживать его в этом состоянии (**U11K0=1**) до окончания процедуры чтения-компенсации;

- 2) установить сигнал **U11K20=1**;
- 3) установить сигнал **U11K21=1** через одну секунду после выполнения предыдущего действия.

U11K21 (WZDS) - сигнал активизирует компенсацию «**смещения нуля**» привода шпинделя.

U11K22 (RZD) - сигнал активизирует чтение «**смещения нуля**» привода одновременно для всех интерполяционных осей. Состояние системы в момент активизации этого сигнала должно быть следующим:

- 1) все оси, включая оси «от точки к точке», должны быть в следящем режиме и выведены в «0»;
- 2) сигнал **STABY (I00K4)=1**;
- 3) сигнал **CYON (I00K7)=0**
- 4) сигнал **WZD (U11K23)=0**.

Рекомендуемое время чтения «**смещения нуля**» привода перед его компенсацией от 0,5 до 1,0 секунды.

U11K23 (WZD) - сигнал активизирует компенсацию «**смещения нуля**» привода одновременно для всех интерполяционных осей. Состояние системы в момент активизации этого сигнала должно быть следующим:

- 1) все оси, включая оси «от точки к точке», должны быть в следящем режиме и выведены в «0»;
- 2) сигнал **STABY (I00K4)=1**;
- 3) сигнал **CYON (I00K7)=0**
- 4) сигнал **RZD (U11K22)=1**.

После установки сигнала **WZD (U11K23)** в состояние «1» его необходимо удерживать в этом состоянии до следующей необходимости повторить операцию чтения сигналом **RZD (U11K22)**.

W11K3 - в этом слове записывается в коде **ASCII** наименование оси шпинделя, активизация которого запрашивается. Если значение этого слова равно «0», активным становится шпиндель, имя которого заявлено в инструкции **ASM** в файле **IOCFIL**. Если запрос разрешен и осуществляется коммутация, система отвечает сигналом **ACKMC=1** о том, что коммутация шпинделя выполнена, в противном случае система устанавливает сигнал **NCKCM**.

24.1.13. Разъём U12K (38K-64K-90K-116K)

U12K0 (FORZ1) }
U12K1 (FORZ2) } - посредством этих двух битов ПЛ выставляет порядковый номер активизируемой оси, объявленной в инструкции **UCDA** файла **IOCFIL**, на принудительную установку положительного или отрицательного аналогового сигнала на выходе преобразователя, номер которого определён име-

нем оси. Установка этих битов выполняется следующим образом:

U12K0 U12K1

0	0	- никакой установки ни на одном канале;
1	0	- установка для первой оси, определённой в UCDA ;
0	1	- установка для второй оси, определённой в UCDA ;
1	1	- установка для третьей оси, определённой в UCDA .

U12K2 (FORZN) - посредством установки этого бита в «0» выполняется принудительная установка положительного аналогового сигнала на канале, который был заранее выбран. Если бит установлен в «1» - сигнал будет отрицательным.

**U12K4, U12K5 }
FOPA1, FOPA2 }** - эти сигналы, если активизированы, задают положительное направление движения вращательной оси «**от точки к точке**», для которой запрошено перемещение. В противном случае система сама выбирает кратчайший путь для позиционирования оси. Данные сигналы должны быть активными перед выдачей запроса на позиционирование для того, чтобы интерфейс мог (при использовании гидравлических двигателей) корректно позиционировать оси сигналами **ROLE, ROLLE, ROPO, RONE**.

**U12K6, U12K7 }
FONA1, FONA2 }** - эти сигналы, если активизированы, задают отрицательное направление движения вращательной оси «**от точки к точке**», для которой запрошено перемещение. В противном случае система сама выбирает кратчайший путь для позиционирования оси. Данные сигналы должны быть активными перед выдачей запроса на позиционирование для того, чтобы интерфейс мог (при использовании гидравлических двигателей) корректно позиционировать оси посредством сигналов **ROLE, ROLLE, ROPO, RONE**.

W12K1 (WORD MAS) - в слове, если в нём ПЛ присвоены значения, реализуется запрос на перемещение осей на позицию, указанную в записи файла **FILMAS** (перемещение осей системой). Запись идентифицируется значением в двоичном коде, присвоенным слову **W12K1**. Данный запрос может активизироваться только функцией конца перемещения с характеристикой «**блокировка вычислений**» и сигналами **SEFA=1, SEFAB=0**. Такие ограничения позволяют программировать, кроме перемещения осей, также какую-либо функцию (не блокирующую вычисления) в записи, подлежащей обработке.

W12K2-W12K3 (WDISEC) - имеется возможность путём присвоения «1» различным битам двух слов запретить положительный или

отрицательный конец хода одной или нескольких осей, входящих в состав 8 осей процесса системы. Это присвоение должно быть выполнено только после того, как ось выведена в «0». Кроме того, следует помнить, что устанавливаемый бит запрещает контроль ограничения перемещения аппаратных выключателей и/или контроль ограничения перемещения, заданный программно. Действие запрета контроля ограничения перемещения снимается сигналом «Сброс». Поэтому, если требуется продлить это действие после сброса, то на время действия сброса (**I00K1**) требуемый сигнал должен быть установлен в «0» и сразу же должен быть установлен в «1».

Если в секции 2 файла **AXCFIL** вы устанавливаете один сигнал для микровыключателя нуля оси (**MCZ**) и для ограничения хода по данной оси, то на всё время, пока ось выходит в «0», контроль ограничения хода оси необходимо сбросить.

24.1.14. Разъём U13K (39K-65K-91K-117K)

- U13K1 (TEACH)** - запрос активизации режима обучения. Запрос активизации режима обучения **TEACH(U13K1)=1** будет выполнен только в режимах **MANU**, **MANJ** или **MDI**. При успешном выполнении данного запроса ПрО установит в ПЛ сигнал **TACKN(I9K19)** в состояние «1». Файл **TEACH** будет открыт для записи. Файл **TEACH** будет закрыт для записи при отмене режима обучения установкой сигнала **TEACH(U13K1)=0** в режимах **MANU**, **MANJ** или **MDI**. При успешном выполнении данного запроса ПрО установит в ПЛ сигнал **TACKN(I9K19)** в состояние «0».
- U13K2 (PSTOR)** - запрос, активизирующий сохранение фактических позиций перемещенных осей в режиме **MANU** или **MANJ** в файле **TEACH**. Запрос активизации сохранения **PSTOR(U13K2)=1** будет выполнен только в режимах **MANU** или **MANJ** при состоянии сигнала **TACKN(I9K19)=1**. При успешном выполнении данного запроса ПрО установит в ПЛ сигнал **TPOSS(I9K20)** в состояние «1», и в файл **TEACH** будет записан кадр с позицией всех перемещённых осей. Запрос активизации сохранения **PSTOR(U13K2)** должен быть сброшен в состояние «0» перед его следующим использованием.
- U13K3 (BSTOR)** - запрос, активизирующий сохранение выполненного кадра в режиме **MDI** в файле **TEACH**. Запрос активизации сохранения **BSTOR(U13K3)=1** будет выполнен только в режиме **MDI** и состоянии сигнала **TACKN(I9K19)=1**. При успешном выполнении данного запроса ПрО установит в ПЛ сигнал **TBLOS(I9K21)** в состояние «1», и в файл **TEACH** будет записан выполненный кадр. Запрос активизации сохранения **BSTOR(U13K3)** должен быть сброшен в состояние «0» перед его следующим использованием.

W13K2-W13K3 - в этих словах задаётся значение эталонного сигнала от 0 до 10 В, который будет активизироваться при наличии сигнала **FORZ1=1** или **FORZ2=1**. Запрос может быть асинхронным, и обновление производится в реальном масштабе времени (см. таблицу 24.1).

24.1.15. Разъём U14K (40K-66K-92K-118K)

W14K0-W14K3 - в этих словах можно от ПЛ присвоить (во время запроса ориентации шпинделя) значение смещения механического нуля относительно электрического нуля для ориентации шпинделя (в градусах с точностью до тысячных долей градуса, формат 5.3, см. таблицу 24.1).

24.1.16. Разъёмы U15K (41K-67K-93K-119K) и U16K (42K-68K-94K-120K)

Наличие этих двух разъёмов (8 слов) позволяет реализовать управление пультом со стороны программы ПЛ интерфейса посредством соответствующих запросов, т.е. имеется возможность запросить от программы ПЛ выполнения функций, которые обычно осуществляются оператором с пульта управления.

W15K0 - биты слова используются для сообщения системе о том, действия каких клавиш и корректоров должны быть запрещены на пульте. Например, установка в «1» третьего бита запрещает действие КОРРЕКТОРА ПОДАЧ «JOG». Последующие слова (**W15K1-W16K2**) используются для установления требуемых значений корректорам и режимным клавишам. Таким образом, если в **W15K0** разрешено действие клавиш и корректоров, системой опрашивается пульт, если запрещено - опрашиваются соответствующие слова в пакете «К».

W15K1 - слово заменяет клавиши режимов работы. В слове должен быть активизирован только один бит, который указывает на выбираемый режим. В случае, когда устанавливается в «1» более одного бита, системе ничего не передается, и установленное значение рассматривается как ошибочное.

U15K2 - слово используется для выбора оси при режимах «**РУЧНЫЕ БЕЗРАЗМЕРНЫЕ ПЕРЕМЕЩЕНИЯ**» и «**ФИКСИРОВАННЫЕ ПЕРЕМЕЩЕНИЯ**», «**ВОЗВРАТ НА ПРОФИЛЬ**», «**ВЫХОД В НУЛЬ**».

W15K3 - слово используется для установления значения КОРРЕКТОРУ ПОДАЧ «JOG». Слову будет присвоено значение, полученное от суммы веса битов, установленных в «1». Таким образом, можно ввести диапазон: **плюс 127%-0-минус 127%**. Старший бит слова используется для указания направления движения: «0» - положительное направление, «1» - отрицательное направление.

W16K0, W16K1 - слова используются для установления значений клавишам: КОРРЕКТОР ПОДАЧИ «F» и КОРРЕКТОР СКОРОСТИ ВРАЩЕНИЯ ШПИДЕЛЯ «S», аналогично слову **W15K3** (за исключением старшего бита).

W16K2 - каждому из первых семи битов этого слова соответствует определённое значение величины перемещения «JOG». Шестой бит не используется. Седьмой бит служит для запроса отработки полного деления «JOG» после команды «ПУСК» (**U10K4**) (даже в случае её выключения) только в том случае, если режим «РУЧНЫЕ ФИКСИРОВАННЫЕ ПЕРЕМЕЩЕНИЯ» установлен от ПЛ.

W16K3 - первые 4 бита слова (**U16K24-U16K27**) по клавише «F2» устанавливаются из ПЛ номера видеостраниц, требуемые для вывода на экран;
- вторые 4 бита слова (**U16K28-U16K31**) по клавише «F1» устанавливаются из ПЛ номера процессов, требуемые для вывода на экран.

Например, **W16K3=022D (0001 0110)** - устанавливает вывод на экран первого процесса и шестой видеостраницы для вывода графики, пока **W16K3=022D**.

24.1.17. Разъём U17K (43K-69K-95K-121K)

W17K0 - при **FILCMD=1** значение, записанное в этом слове, представляет собой двоичный код номера записи, содержащей клавиатурную команду, или имя управляющей программы, которая выбирается по запросу ПЛ;
- при **FILCMD=0** значение, записанное в этом слове, представляет собой двоичный код номера первого из десяти разъёмов пакета «K», который содержит имя, выбираемое от ПЛ.

W17K1 (WRCOMJ, J=1-8) - посредством этого слова переставляется возможным запрашивать у системы коммутацию двух осей процесса, принадлежащих одному процессу. В процессе можно установить максимально четыре пары коммутируемых осей. Коммутация двух осей выполняется установкой в «1» тех битов, которые соответствуют им в этом слове. Если интерфейс принимает запрос, он в этом случае сообщает об этом ПЛ посредством **W06K0**. После коммутации оси, система контролирует только допуск позиционирования.

W17K2 - число, записанное в данное слово, является коэффициентом деления для величины коррекции ручных перемещений «JOG», указанной в слове **W15K3**. Коэффициент, заданный в байте **W17K2**, учитывается только при состоянии сигнала **U15K3=1**. Коэффициент деления не учитывается, если **U15K3=0** или **W17K2=0** или **W17K2=1**.

Пример

W15K3=10 - текущая коррекция скорости ручных перемещений равна 10%.
 W17K2=200 - коэффициент деления равен 200.

Фактический процент: $10\%/200=0.05\%$.

Значение 0.05 индицируется в поле вывода процентов корректора ручных перемещений «**JOG**».

W17K3

- воспроизведение номера сообщения. Посредством присвоения этого слова можно воспроизводить одно из сообщений ПЛ, находящихся в файле **FILMS5**. Номер воспроизводимого сообщения определяется двоичным значением присвоенного слова. Поэтому можно воспроизводить одно из 255 сообщений ранее записанного файла.

Текст активизируемых сообщений должен быть записан в файле, имя которого записано в секции 2 файла **FCRSYS/MP0** справа от логического имени **FILMS5**. Если этого файла нет, то визуализируется только номер сообщения.

Файл сообщений создаётся командой **EDI**. Формат строк файла сообщений:

Код цвета#текст

где:

- | | |
|------------------|--|
| Код цвета | - десятичное число от 0 до 15 , определяющее цвет текста в поле его вывода на экран. Соответствие цвета и его кода указано в документе «Руководство программиста» в разделе 5 «Программирование на языке ASSET». |
| # | - символ разделителя. Не допускается символ <пробел> до и после знака разделителя. |
| Текст | - 43 символа. |

24.1.18. Разъёмы U18K (44K-70K-96K-122K) и U19K (45K-71K-97K-123K)

W18K0-W18K3
W19K0-W19K3 } - в этих четырёх словах ПЛ передаёт системе позицию, которую необходимо достичь при перемещении первой и второй осей «от точки к точке» (максимум 99999 позиций и 999 точек внутри каждой позиции). Количество позиций задаётся при характеристике в файле **IOSFIL**.

24.1.19. Разъём U20K (46K-72K-98K-124K)

W20K0 (WASMO1) } - два слова служат для задания двоичного номера оси
W20K1 (WASMO2) } «от точки к точке», которая должна перемещаться, когда параллельно присутствует запрос на позиционирование оси «от точки к точке» (первой или второй).

W20K2, W20K3 } - в двух словах ПЛ сообщает системе процент значения
(FEAS1, FEAS2) } скорости быстрого хода (по отношению к заданному при характеристике) для осей «от точки к точке» первой или второй соответственно.

24.1.20. Разъёмы U21K (47K-73K-99K-125K) и U22K (48K-74K-100K-126K)

W21K0-W21K3 } - с помощью сигналов этих разъёмов можно запросить
W22K1-W22K3 } визуализацию 64 сообщений. Текст активизируемых сообщений должен быть записан в файле, имя которого записано в секции 2 файла **FCRSYS/MP0** справа от логического имени **FILMS5**. Если этого файла нет, визуализируется только номер сообщения. Создание файла и формат его строк приведён в п.24.1.17 при описании байта **W17K3**.

24.2. Таблица базовых сигналов пакета «К»

В таблице 24.1 представлены все базовые сигналы пакета «К» интерфейса PLC.

Таблица 24.1 – Базовые сигналы пакета «К» интерфейса PLC

Разъём: I00K(26K-52K-78K-104K) Сигналы Про->PLC			
Слово	Бит	Сигнал	Назначение
0	0	EMERG	Процесс в аварийном состоянии
	1	RESE	Процесс в состоянии СБРОС; RESE=1 два медленных цикла ПЛ
	2	CONP	Процесс готов к обслуживанию осей
	3	CYCLE	Процесс в состоянии выполнения кадра
	4	STABY	В процессе «оси остановлены» и находятся в позиции
	5		
	6	RCM	Процесс в состоянии «поиска кадра с запоминанием»
	7	CYON	Включение ПУСКА
1	8	ABI 1	Ось 1 следящий режим включен
	9	ABI 2	Ось 2 следящий режим включен
	10	ABI 3	Ось 3 следящий режим включен
	11	ABI 4	Ось 4 следящий режим включен
	12	ABI 5	Ось 5 следящий режим включен
	13	ABI 6	Ось 6 следящий режим включен
	14	ABI 7	Ось 7 следящий режим включен
	15	ABI 8	Ось 8 следящий режим включен
			Ответ осей о готовности обслуживания привода
2	16	MOV 1	Ось 1 запрос движения
	17	MOV 2	Ось 2 запрос движения
	18	MOV 3	Ось 3 запрос движения
	19	MOV 4	Ось 4 запрос движения
	20	MOV 5	Ось 5 запрос движения
	21	MOV 6	Ось 6 запрос движения
	22	MOV 7	Ось 7 запрос движения
	23	MOV 8	Ось 8 запрос движения
			Признак «ось в движении»; устанавливается для каждой оси в начале движения и сбрасывается по окончании движения. При G27, G28 устанавливается в начале кадра
3	24	POSIA	Движение осей, запрошенное через W12K1, выполнено
	25	POSIM	Шпиндель сориентирован
	26	NCKTO	Обновление инструмента в шпинделе не выполнено
	27	ACKTO	Обновление инструмента в шпинделе выполнено
	28	HOLDA	Процесс в состоянии «СТОП»
	29	ACKCM	Коммутация оси шпинделя выполнена
	30	NCKCM	Коммутация оси шпинделя не выполнена
	31	DIRMN	Ручное перемещение осей в отрицательном направлении

Продолжение таблицы 24.1

Разъём: I01K(27K-53K-79K-105K) Сигналы ПрО->PLC			
Слово	Бит	Сигнал	Назначение
0	0		Тысячные 0,00X
	1		
	2		
	3		
	4		Сотые 0,0X
	5		
	6		
	7		
1	8		Десятые 0,X
	9		
	10		
	11		
	12		Целые X
	13		
	14		
	15		
2	16		Десятки X0
	17		
	18		
	19		
	20		Сотни X00
	21		
	22		
	23		
3	24		Тысячи X000
	25		
	26		
	27		
	28		Десятки тысяч X0000
	29		
	30		
	31		

Заданная величина перемещения для индексных осей в коде BCD. Формат записи 5.3.

Продолжение таблицы 24.1

Разъём: I02K(28K-54K-80K-108K) Сигналы ПрО->PLC				
Слово	Бит	Сигнал	Назначение	
0	0		Заданная функция «Т» (из таблицы RANDOM), заданная позиция магазина инструментов в коде BCD	
	1			
	2			
	3			
	4			Целые X
	5			
	6			
	7			
1	8		Десятки X0	
	9			
	10			
	11			
	12		Сотни X00	
	13			
	14			
	15			
2	16		Функция «Т» инструмента, предназначенного для установки в шпиндель (в коде BCD)	
	17			
	18			
	19			
	20			Целые X
	21			
	22			
	23			
3	24		Десятки X0	
	25			
	26			
	27			
	28		Сотни X00	
	29			
	30			
	31			
			Тысячи X000	

Продолжение таблицы 24.1

Разъём: I03K(29K-55K-81K-107K) Сигналы ПрО->PLC			
Слово	Бит	Сигнал	Назначение
0	0		Импульсные «М» функции в коде VCD («М» функции начала и конца движения). Код устанавливается при GEFA=1 и держится 2 цикла ПЛ.
	1		
	2		
	3		
	4		
	5		
	6		
	7		
1	8		Функции «М» немедленного действия в коде VCD. Код устанавливается в начале перемещения и держится при MOV=1 и COMU=1 .
	9		
	10		
	11		
	12		
	13		
	14		
	15		
2	16		
	17		
	18		
	19		
	20		
	21		
	22		
	23		
3	24		
	25		
	26		
	27		
	28		
	29		
	30		
	31		

Продолжение таблицы 24.1

Разъём: I04K(30K-56K-82K-108K) Сигналы ПрО->PLC			
Слово	Бит	Сигнал	Назначение
0	0		Программируемая «Т» функция в коде BCD. Код устанавливается при SEFA=1 и держится до смены или сброса
	1		
	2		
	3		
	4		
	5		
	6		
7			
1	8		
	9		
	10		
	11		
	12		
	13		
	14		
	15		
2	16	FUAS	Строб «S» функции
	17	FUAT	Строб «Т» функции
	18	FUAM	Строб «М» функции
	19		
	20	TASC1	Строб индексной оси 1; держится 2 цикла ПЛ
	21	TASC2	Строб индексной оси 2; держится 2 цикла ПЛ
	22	TASC3	Строб индексной оси 3; держится 2 цикла ПЛ
	23	FUTKO	Строб «Т» функции, когда инструмент выбирается из таблицы ресурса инструмента; держится 2 цикла ПЛ
3	24	S	Направление перемещения индексных осей: «0» - (+) направление, «1» - (-) направление
	25	MPROFI	Признак «ось движется по профилю»
	26	CUMAN	Признак ручной произвольной смены инструмента (при программировании «Т», не содержащегося в таблице RANDOM)
	27	PWMAN	Запрос на вращение шпинделя
	28	SOFIT	Признак быстрого хода в цикле измерения
	29	GOMAN	Превышение порога скорости шпинделя (заявляется РОМ в АХСFIL); устанавливается для шпинделя имеющего преобразователь
	30	ACKCY	Запрос на перемещение осей (задан с пульта) выполнен
	31	NCKCY	Запрос на перемещение осей (задан с пульта) не принят

Продолжение таблицы 24.1

Разъём: I05K(31K-57K-83K-109K) Сигналы ПрО->PLC			
Слово	Бит	Сигнал	Назначение
0	0	FG00	Признак отработки G00
	1	PROFIT	Признак отработки G1, G2 или G3
	2	MISU	Признак цикла измерения
	3	INTUT	Признак проверки целостности инструмента
	4	FG8.1	Признак циклов G81-G89 (код последней цифры в номере цикла 1-9)
	5	FG8.2	
	6	FG8.4	
	7	FG8.8	
1	8	ASRI 1	Признак выхода в «0» по осям.
	9	ASRI 2	
	10	ASRI 3	
	11	ASRI 4	
	12	ASRI 5	
	13	ASRI 6	
	14	ASRI 7	
	15	ASRI 8	
2	16	ROPO 1	Признак движения 1-ой оси от (.) к (.) в направлении (+)
	17	ROPO 2	Признак движения 2-ой оси от (.) к (.) в направлении (+)
	18	RONE 1	Признак движения 1-ой оси от (.) к (.) в направлении (-)
	19	RONE 2	Признак движения 2-ой оси от (.) к (.) в направлении (-)
	20	ROLE 1	Признак достижения 1-ой осью от (.) к (.) 1-го порога замедления
	21	ROLE 2	Признак достижения 2-ой осью от (.) к (.) 1-го порога замедления
	22	ROLLE1	Признак достижения 1-ой осью от (.) к (.) 2-го порога замедления
	23	ROLLE2	Признак достижения 2-ой осью от (.) к (.) 2-го порога замедления
3	24	POSI 1	1-я ось от (.) к (.) в зоне допуска позиционирования
	25	POSI 2	2-я ось от (.) к (.) в зоне допуска позиционирования
	26	BUSY 1	1-ый канал для движения оси от (.) к (.) занят
	27	BUSY 2	2-ой канал для движения оси от (.) к (.) занят
	28	KOSI 1	Ошибка запроса на движение оси в 1-ом канале
	29	KOSI 2	Ошибка запроса на движение оси во 2-ом канале
	30	ACKG	Запрос на выбор программы (SPG) или действие команды с ПО принят
	31	NCKG	Запрос на выбор программы (SPG) или действие команды с ПО не принят

Продолжение таблицы 24.1

Разъём: I06K(32K-58K-84K-110K) Сигналы ПрО->PLC			
Слово	Бит	Сигнал	Назначение
0	0	COM 1	Ответ о включении перекоммутируемой оси [номер оси (бита) определяется в AXCFIL в инструкции INx]
	1	COM 2	
	2	COM 3	
	3	COM 4	
	4	COM 5	
	5	COM 6	
	6	COM 7	
	7	COM 8	
1	8	NVID	Номер активной видеостраницы
	9		
	10		
	11		
	12	NPROC	Номер активного процесса
	13		
	14		
	15		
2	16	SGAM 1	1-й диапазон скорости шпинделя
	17	SGAM 2	2-й диапазон скорости шпинделя
	18	SGAM 3	3-й диапазон скорости шпинделя
	19	SGAM 4	4-й диапазон скорости шпинделя
	20	RETRA	Устанавливается в «1» при MBR=1 и держится до возврата к кадру активизации MBR
	21	RSPEPN	Сообщает ПЛ о готовности УЧПУ к подаче управляющего напряжения на станок
	22	MODEC	Сообщает ПЛ об активности режима «КОМАНДА». Сигнал устанавливается только в УЧПУ NC-200/201/201M/202/210/220/230/310
	23	ABIM	Признак - «шпиндель включён»
3	24		Номер оси, которая находится в состоянии аварии
	25		
	26		
	27		
	28		Двоичный код аварии: 1 - отключение от ПЛ; 2 - сбой привода; 3 - ошибка скорости; 5 - ошибка датчика; 6 - ось вне допуска
	29		
	30		
	31		

Продолжение таблицы 24.1

Разъём: I07K(33K-59K-81K-111K) Сигналы ПрО->PLC			
Слово	Бит	Сигнал	Назначение
0	0		Номер корректора для альтернативного инструмента в коде BCD
	1		
	2		
	3		
	4		
	5		
	6		
	7		
1	8		
	9		
	10		
	11		
	12		
	13		
	14		
	15		
2	16	Единицы	
	17		
	18		
	19		
	20	Десятки	
	21		
	22		
	23		
3	24	Сотни	
	25		
	26		
	27		
	28	Тысячи	
	29		
	30		
	31		

Продолжение таблицы 24.1

Разъем: I08K(34K-60K-86K-112K) Сигналы ПрО-→PLC			
Слово	Бит	Сигнал	Назначение
0	0		Программируемая функция «S» в коде BCD
	1		
	2		
	3		
	4		
	5		
	6		
7			
1	8		
	9		
	10		
	11		
	12		
	13		
	14		
15			
2	16		
	17		
	18		
	19		
	20		
	21		
	22		
23			
3	24	EMDI	Режим «РУЧНОЙ ВВОД»
	25	AUTO	Режим «АВТОМАТ»
	26	SEMI	Режим «КАДР»
	27	MANUC	Режим «НАЛАДКА-БЕЗРАЗМЕРНЫЕ ПЕРЕМЕЩЕНИЯ»
	28	MANUJ	Режим «НАЛАДКА-ФИКСИРОВАННЫЕ ПЕРЕМЕЩЕНИЯ»
	29	RIPRO	Режим «ВОЗВРАТ НА ТРАЕКТОРИЮ»
	30	RIMZE	Режим «ВЫХОД В НОЛЬ»
	31	RESET	Режим «СБРОС»

Продолжение таблицы 24.1

Разъём: I09K(35K-61K-81K-113K) Сигналы ПрО-→PLC			
Слово	Бит	Сигнал	Назначение
0	0		Код ошибки из файла FILMS4
	1		
	2		
	3		
	4		
	5		
	6		
	7		
1	8	VILA1	Сигнал балансировки для оси 1
	9	VILA2	Сигнал балансировки для оси 2
	10	VILA3	Сигнал балансировки для оси 3
	11	VILA4	Сигнал балансировки для оси 4
	12	VILA5	Сигнал балансировки для оси 5
	13	VILA6	Сигнал балансировки для оси 6
	14	VILA7	Сигнал балансировки для оси 7
	15	VILA8	Сигнал балансировки для оси 8
2	16		
	17		
	18		
	19	TACKN	Ответ о включении режима обучения TEACHING
	20	TPOSS	Ответ о сохранении позиции осей в файле TEACH
	21	TBLOS	Ответ о сохранении кадра в файле TEACH
	22	TPROB	Проблема в режиме обучения
	23		
3	24	INVER	Признак обратного вращения шпинделя для G84
	25	STOP	Признак останова шпинделя для G86
	26	FG96	Признак активности G96 или G97
	27		
	28		
	29		
	30		
	31		

Продолжение таблицы 24.1

Разъём: U10K(36К-62К-88К-114К) Сигналы PLC->Про			
Слово	Бит	Сигнал	Назначение
0	0	MUSPE	Признак выключения станка по аварии
	1	REAZ	Запрос на «СБРОС»
	2	HLDR	Запрос на остановку осей с повторным стартом
	3	RHOE	Запрос на остановку осей с автоматическим рестартом
	4	CYST	Запрос на «ПУСК» цикла
	5	FOLD	Запрос на прерывание подачи
	6	WAIC	«0»-запрос на ожидание 2-х циклов ПЛ перед началом движения осей, «1»-без ожидания 2-х циклов ПЛ перед началом движения осей
	7	RISPE	Запрос на отключение станка из-за неправильной функции
1	8	RABI 1	Запрос +включения следящего режима для осей, которые указаны в интерполяторе непрерывных координатных осе процесса (8 осей)
	9	RABI 2	
	10	RABI 3	
	11	RABI 4	
	12	RABI 5	
	13	RABI 6	
	14	RABI 7	
	15	RABI 8	
2	16	REGTOL	Сброс инструмента, находящегося в шпинделе
	17	DITVU	Запрет счета ресурса инструмента
	18	GCOM	Запрос на выбор программы (SPG) от ПЛ
	19	CMDLOG	Запрос на выполнение клавиатурной команды от ПЛ
	20	ASPEPN	Разрешение включения управляющего напряжения на станке (включение реле «SPEPN»)
	21	AGTOL	Запрос на обновление инструмента и соответствующего корректора
	22		
	23	FILCMD	Запрос доступа интерфейса к файлу, где описаны имена программ и устройства памяти (где эти программы находятся) и команды, набираемые на ПО
3	24	COMU	Разрешение движения осей
	25	CEFA	Разрешение выполнения функций начала и конца движения «М», «S», «Т» и «индексной оси»
	26	CEFAB	При CEFAB=0 разрешается выполнение функций «М» конца движения, имеющих блокировку вычислений
	27	MIZE 1	Признак выхода в «0» для индексной оси 1
	28	MIZE 2	Признак выхода в «0» для индексной оси 2
	29	MIZE 3	Признак выхода в «0» для индексной оси 3
	30	RMORE	Запрос на запуск мультиблока (при MBR=1)
	31		

Продолжение таблицы 24.1

Разъём: U11K(37K-63K-89K-115K) Сигналы PLC->Про				
Слово	Бит	Сигнал	Назначение	
0	0	ANGOM	Запрос ориентации шпинделя в соответствии с инструкцией POM в AXCFIL (разъём 14K)	
	1	FOMAO	Принудительное вращение шпинделя по часовой стрелке	Информация о величине напряжения в W11K1
	2	FOMAA	Принудительное вращение шпинделя против часовой стрелки	
	3	ROMAO	Вращение шпинделя по часовой стрелке	Диапазон вращения шпинделя в W11K2
	4	ROMAA	Вращение шпинделя против часовой стрелки	
	5	FORID	Коэффициент 0,1 мах напряжения на выходе в вольтах (для получения сотых долей максимального напряжения при использовании W11K1)	
	6	ORIMO	Установка направления ориентации шпинделя по часовой стрелке	
	7	ORIMA	Установка направления ориентации шпинделя против часовой стрелки	
1	8		Десятые доли	Напряжение в вольтах и десятых долях вольта, представляющее процент по отношению к максимальному уровню на выходе, если присутствует запрос на вращение шпинделя FOMAO или FOMAA
	9			
	10			
	11			
	12		Целые	
	13			
	14			
	15			
2	16	GAMM 1	На станке включен 1-й диапазон вращения шпинделя	
	17	GAMM 2	На станке включен 2-й диапазон вращения шпинделя	
	18	GAMM 3	На станке включен 3-й диапазон вращения шпинделя	
	19	GAMM 4	На станке включен 4-й диапазон вращения шпинделя	
	20	RZDS	Чтение смещения нуля привода шпинделя	
	21	WZDS	Компенсация смещения нуля привода шпинделя	
	22	RZD	Чтение смещения нуля привода	
	23	WZD	Компенсация смещения нуля привода	
3	24		Имя оси шпинделя, заявленное в инструкции NAS в файле AXCFIL (в шестнадцатеричном коде)	
	25			
	26			
	27			
	28			
	29			
	30			
	31			

Продолжение таблицы 24.1

Разъём: U12K(38К-64К-80К-116К) Сигналы PLC->Про			
Слово	Бит	Сигнал	Назначение
0	0	FORZ 1	Двоичный код номера оси, управляемой через ЦАП от ПЛ
	1	FORZ 2	
	2	FORZN	Запрос на принудительную установку отрицательного напряжения на выходе ЦАП оси
	3		
	4	FOPA 1	Задание на движение оси в положительном направлении
	5	FOPA 2	
	6	FONA 1	Задание на движение оси в отрицательном направлении
	7	FONA 2	
1	8	WORD MAS	Двоичный код номера записи в FILMAS (движение оси, задаваемое системой, а не технологической программой)
	9		
	10		
	11		
	12		
	13		
	14		
2	15		Отключение ограничения перемещений по осям
	16	DI 1+	
	17	DI 1-	
	18	DI 2+	
	19	DI 2-	
	20	DI 3+	
	21	DI 3-	
	22	DI 4+	
23	DI 4-		
3	24	DI 5+	
	25	DI 5-	
	26	DI 6+	
	27	DI 6-	
	28	DI 7+	
	29	DI 7-	
	30	DI 8+	
	31	DI 8-	

Продолжение таблицы 24.1

Разъём: U13K(39K-65K-91K-117K) Сигналы PLC->Про			
Слово	Бит	Сигнал	Назначение
0	0		
	1	TEACH	Запрос включения режима обучения TEACHING
	2	PSTOR	Запрос сохранения позиции осей в файле TEACH
	3	BSTOR	Запрос сохранения кадра в файле TEACH
	4		
	5		
	6		
	7		
1	8		
	9		
	10		
	11		
	12		
	13		
	14		
	15		
2	16		Величина напряжения в вольтах, подаваемого к преобразователю ЦАП для организации движения оси, объявленной в инструкции UCDA в файле IOCFIL (максимум 10 вольт)
	17		
	18	Милливольты	
	19		
	20		
	21	Сотые доли вольта	
	22		
	23		
3	24		
	25	Десятые доли вольта	
	26		
	27		
	28		
	29	Вольты	
	30		
	31		

Продолжение таблицы 24.1

Разъём: U14K(40K-66K-92K-118K) Сигналы PLC->ПрО			
Слово	Бит	Сигнал	Назначение
0	0		Тысячные
	1		
	2		
	3		
	4		Сотые
	5		
	6		
	7		
1	8		Десятые
	9		
	10		
	11		
	12		Единицы
	13		
	14		
	15		
2	16		Десятки
	17		
	18		
	19		
	20		Сотни
	21		
	22		
	23		
3	24		Тысячи
	25		
	26		
	27		
	28		Десятки тысяч
	29		
	30		
	31		

Смещение механического нуля шпинделя относительно электрического нуля в отличие от смещения, заявленного в инструкции **РОМ** в **АХСF1L**.
Запись в коде BCD (формат 5.3)

Продолжение таблицы 24.1

Разъём: U15K(41K-67K-93K-119K) Сигналы PLC->Про			
Слово	Бит	Сигнал	Назначение
0	0		Запрет клавиши «ПУСК»
	1		
	2		Запрет выбора режима работы
	3		Запрет корректора подач «JOG»
	4		Запрет корректора подачи «F»
	5		Запрет корректора скорости вращения шпинделя «S»
	6		Запрет выбора оси
	7		Запрет выбора «JOG»
1	8		Выбор режима «РУЧНОЙ ВВОД»
	9		Выбор режима «АВТОМАТ»
	10		Выбор режима «КАДР»
	11		Выбор режима «БЕЗРАЗМЕРНЫЕ РУЧНЫЕ ПЕРЕМЕЩЕНИЯ»
	12		Выбор режима «ФИКСИРОВАННЫЕ РУЧНЫЕ ПЕРЕМЕЩЕНИЯ»
	13		Выбор режима «ВОЗВРАТ НА ПРОФИЛЬ»
	14		Выбор режима «ВЫХОД В НОЛЬ»
	15		Выбор режима «СБРОС»
2	16		Назначение номера оси: 1
	17		Назначение номера оси: 2
	18		Назначение номера оси: 3
	19		Назначение номера оси: 4
	20		Назначение номера оси: 5
	21		Назначение номера оси: 6
	22		Назначение номера оси: 7
	23		Назначение номера оси: 8
3	24		Назначение величины корректора подач «JOG»: 1%
	25		Назначение величины корректора подач «JOG»: 2%
	26		Назначение величины корректора подач «JOG»: 4%
	27		Назначение величины корректора подач «JOG»: 8%
	28		Назначение величины корректора подач «JOG»: 16%
	29		Назначение величины корректора подач «JOG»: 32%
	30		Назначение величины корректора подач «JOG»: 64%
	31		Назначение направления перемещения оси от ПЛ: «0»-положительное направление; «1»-отрицательное

Продолжение таблицы 24.1

Разъём: U16K(42K-68K-94K-120K) Сигналы PLC->Про			
Слово	Бит	Сигнал	Назначение
0	0		Назначение величины корректора подачи «F»: 1%
	1		Назначение величины корректора подачи «F»: 2%
	2		Назначение величины корректора подачи «F»: 4%
	3		Назначение величины корректора подачи «F»: 8%
	4		Назначение величины корректора подачи «F»: 16%
	5		Назначение величины корректора подачи «F»: 32%
	6		Назначение величины корректора подачи «F»: 64%
	7		
1	8		Назначение величины кор-ра скор. вращ. шпинделя «S»: 1%
	9		Назначение величины кор-ра скор. вращ. шпинделя «S»: 2%
	10		Назначение величины кор-ра скор. вращ. шпинделя «S»: 4%
	11		Назначение величины кор-ра скор. вращ. шпинделя «S»: 8%
	12		Назначение величины кор-ра скор. вращ. шпинделя «S»: 16%
	13		Назначение величины кор-ра скор. вращ. шпинделя «S»: 32%
	14		Назначение величины кор-ра скор. вращ. шпинделя «S»: 64%
	15		
2	16		Назначение величины JOG: 0,001 мм
	17		Назначение величины JOG: 0,010 мм
	18		Назначение величины JOG: 0,100 мм
	19		Назначение величины JOG: 1,000 мм
	20		Назначение величины JOG: 10,000 мм
	21		Назначение величины JOG: 100,000 мм
	22		
	23		Запрос на выполнение всей величины «JOG» в режиме «ФИКСИРОВАННЫЕ РУЧНЫЕ ПЕРЕМЕЩЕНИЯ»
3	24		4 бита для задания требуемой видеостраницы
	25		
	26		
	27		
	28		4 бита для задания требуемого процесса
	29		
	30		
	31		

Продолжение таблицы 24.1

Разъём: U17K(43K-69K-95K-121K) Сигналы PLC->Про			
Слово	Бит	Сигнал	Назначение
0	0		При FILCMD=0 величина, записанная в этом слове, представляет собой двоичный номер первого из десяти разъемов пакета « K », где записано имя выбираемой управляющей программы и устройство, где она находится (в коде ASCII). При FILCMD=1 величина, записанная в этом слове, представляет двоичный номер записи в файле FILCMD , где записано имя и устройство выбираемой управляющей программы
	1		
	2		
	3		
	4		
	5		
	6		
	7		
1	8	RCOM1	Запрос о переключении осей. Возможно переключение четырех из 8 осей с другими четырьмя, определяемыми в AXCFIL . Ответом на переключение осей является I06K0 .
	9	RCOM2	
	10	RCOM3	
	11	RCOM4	
	12	RCOM5	
	13	RCOM6	
	14	RCOM7	
	15	RCOM8	
2	16		Число, записанное в данном слове, является коэффициентом деления для величины коррекции ручных перемещений « JOG », указанной в слове W15K3 . Коэффициент, заданный в этом слове, учитывается только при состоянии сигнала U15K3=1 . Коэффициент деления не учитывается, если U15K3=0 или W17K2=0 , или W17K2=1 .
	17		
	18		
	19		
	20		
	21		
	22		
	23		
3	24		Двоичный номер сообщения для визуализации на дисплее (из файла FILMS5). Максимальный номер - 255.
	25		
	26		
	27		
	28		
	29		
	30		
	31		

Продолжение таблицы 24.1

Разъём: U18K(44K-70K-96K-122K) Сигналы PLC->Про			
Слово	Бит	Сигнал	Назначение
0	0		0,00X
	1		
	2		
	3		
	4		0,0X
	5		
	6		
	7		
1	8		0,X
	9		
	10		
	11		
	12		X
	13		
	14		
	15		
2	16		X0
	17		
	18		
	19		
	20		X00
	21		
	22		
	23		
3	24		X000
	25		
	26		
	27		
	28		X0000
	29		
	30		
	31		

Позиция, которую должна достигнуть первая ось «от точки к точке». Число различных позиций определяется в **IOCFIL**.

Информация на этот разъём может поступать с разъёма **01K** или других разъёмов или слов.
Код BCD в формате 5.3.

Продолжение таблицы 24.1

Разъём: U19K(45K-71K-97K-123K) Сигналы PLC->Про			
Слово	Бит	Сигнал	Назначение
0	0		0,00X
	1		
	2		
	3		
	4		0,0X
	5		
	6		
	7		
1	8		0,X
	9		
	10		
	11		
	12		X
	13		
	14		
	15		
2	16		X0
	17		
	18		
	19		
	20		X00
	21		
	22		
	23		
3	24		X000
	25		
	26		
	27		
	28		X0000
	29		
	30		
	31		

Позиция, которую должна достигнуть вторая ось «от точки к точке». Число различных позиций определяется в **IOCFIL**. Информация на этот разъем может поступать с разъёма 01K или других разъемов или слов. Код BCD в формате 5.3.

Продолжение таблицы 24.1

Разъём: U20K (46K-72K-98K-124K) Сигналы PLC->Про				
Слово	Бит	Сигнал	Назначение	
0	0	WASMO1	Двоичный код номера оси «от точки к точке» при запросе на ее движение (по первому каналу связи)	
	1			
	2			
	3			
	4			
	5			
	6			
	7			
1	8	WASMO2	Двоичный код номера оси «от точки к точке» при запросе на ее движение (по второму каналу связи)	
	9			
	10			
	11			
	12			
	13			
	14			
	15			
2	16	FEAS1	Корректор скорости осей «от точки к точке» для первого канала:	1%
	17			2%
	18			4%
	19			8%
	20			16%
	21			32%
	22			64%
	23			-
3	24	FEAS2	Корректор скорости осей «от точки к точке» для второго канала:	1%
	25			2%
	26			4%
	27			8%
	28			16%
	29			32%
	30			64%
	31			-

Продолжение таблицы 24.1

Разъём: U21K (47K-73K-99K-125K) Сигналы PLC->Про			
Слово	Бит	Сигнал	Назначение
0	0	MSG 0	Номер сообщения для визуализации на дисплее. Текст сообщений должен быть предварительно за- писан в файле FILMS5. Если файл отсутствует, будет визуализироваться номер сообщения без текста. Номер сообщения соответствует номеру разряда в разъёме.
	1	MSG 1	
	2	MSG 2	
	3	MSG 3	
	4	MSG 4	
	5	MSG 5	
	6	MSG 6	
	7	MSG 7	
1	8	MSG 8	
	9	MSG 9	
	10	MSG 10	
	11	MSG 11	
	12	MSG 12	
	13	MSG 13	
	14	MSG 14	
	15	MSG 15	
2	16	MSG 16	
	17	MSG 17	
	18	MSG 18	
	19	MSG 19	
	20	MSG 20	
	21	MSG 21	
	22	MSG 22	
	23	MSG 23	
3	24	MSG 24	
	25	MSG 25	
	26	MSG 26	
	27	MSG 27	
	28	MSG 28	
	29	MSG 29	
	30	MSG 30	
	31	MSG 31	

Продолжение таблицы 24.1

Разъем: U22K (48K-74K-100K-126K) Сигналы PLC->Про			
Слово	Бит	Сигнал	Назначение
0	0	MSG 32	Номер сообщения для визуализации на дисплее. Текст сообщений должен быть предварительно записан в файле FILMS5 . Если файл отсутствует, будет визуализироваться номер сообщения без текста. Номер сообщения соответствует номеру разряда в разъёме
	1	MSG 33	
	2	MSG 34	
	3	MSG 35	
	4	MSG 36	
	5	MSG 37	
	6	MSG 38	
	7	MSG 39	
1	8	MSG 40	
	9	MSG 41	
	10	MSG 42	
	11	MSG 43	
	12	MSG 44	
	13	MSG 45	
	14	MSG 46	
	15	MSG 47	
2	16	MSG 48	
	17	MSG 49	
	18	MSG 50	
	19	MSG 51	
	20	MSG 52	
	21	MSG 53	
	22	MSG 54	
	23	MSG 55	
3	24	MSG 56	
	25	MSG 57	
	26	MSG 58	
	27	MSG 59	
	28	MSG 60	
	29	MSG 61	
	30	MSG 62	
	31	MSG 63	

24.3. Сигналы пакета «N»

24.3.1. Назначение сигналов пакета «N»

Развитие или модернизация функциональных возможностей УЧПУ влечёт за собой расширение протокола обмена между базовым ПрО и PLC, следствием чего являются новые сигналы, которые образуют новый пакет «N». Новый пакет «N» введён для расширения существующего пакета «K». Структуры пакетов «N» и «K» одинаковы.

24.3.2. Разъём 00N (26N-52N-78N-104N)

- I00N0 (ACUB)** - сигнал устанавливается в «1», если 3 оси находятся в объёмной защищённой зоне **CUB** при сигнале **U10N0=1 (ECDF)** .
- I00N8 (UAS)** - сигнал устанавливается в «1», если состояние команды **UAS=1**, и сбрасывается, если состоянии команды **UAS=0** .
- I00N9 (USB)** - сигнал устанавливается в «1», если состояние команды **USB=1**, и сбрасывается, если состоянии команды **USB=0** .
- I00N10 (USO)** - сигнал устанавливается в «1», если состояние команды **USO=1**, и сбрасывается, если состоянии команды **USO=0** .
- I00N11 (UVR)** - сигнал устанавливается в «1», если состояние команды **UVR=1**, и сбрасывается, если состоянии команды **UVR=0** .
- I00N12 (URL)** - сигнал устанавливается в «1», если состояние команды **URL=1**, и сбрасывается, если состоянии команды **URL=0** .
- I00N13 (MBR)** - сигнал устанавливается в «1», если состояние команды **MBR=1**, и сбрасывается, если состоянии команды **MBR=0** .
- I00N14 (RCM)** - сигнал устанавливается в «1», если состояние команды **RCM=1**, и сбрасывается, если состоянии команды **RCM=0** .
- I00N15 (VOL)** - сигнал устанавливается в «1», если состояние команды **VOL=1**, и сбрасывается, если состоянии команды **VOL=0** .
- I00N16 (RAP)** - сигнал устанавливается в «1», если состояние команды **RAP=1**, и сбрасывается, если состоянии команды **RAP=0** .
- I00N17 (UEP)** - сигнал устанавливается в «1», если состояние команды **UEP=1**, и сбрасывается, если состоянии команды **UEP=0** .

24.3.3. Разъём 01N (27N-53N-79N-105N)

- W01N0** - слово сообщает ПЛ значение в %, установленное на корректоре подачи «JOG», с округлением до целых. Стар-

ший бит этого слова **I01N7** указывает направление, установленное этим корректором:

- «1» - отрицательное;
- «0» - положительное.

W01N1 - слово сообщает ПЛ значение в %, установленное на корректоре подачи «**F**», с округлением до целых.

W01N2 - слово сообщает ПЛ значение в %, установленное на корректоре подачи «**S**», с округлением до целых.

W01N3 - слово сообщает ПЛ:

- позицию переключателя режимов работы «**MDI**», ..., «**RESET**» в УЧПУ NC-201/201M/202/210/220/230;
- «1» в одном из битов этого слова для УЧПУ NC-110/310 соответствует последней нажатой на СП клавише выбора режима работы.

24.3.4. Разъём 03N (29N-55N-81N-107N)

I03N0 (PPRI1) - признак выхода в «0» для оси «**ТТ**» №1 (**ТА1**) (**ECDF**).

I03N1 (PPRI2) - признак выхода в «0» для оси «**ТТ**» №2 (**ТА2**) (**ECDF**).

I03N2 (PPRI3) - признак выхода в «0» для оси «**ТТ**» №3 (**ТА3**) (**ECDF**).

I03N3 (PPRI4) - признак выхода в «0» для оси «**ТТ**» №4 (**ТА4**) (**ECDF**).

I03N4 (PPRI5) - признак выхода в «0» для оси «**ТТ**» №5 (**ТА5**) (**ECDF**).

I03N5 (PPRI6) - признак выхода в «0» для оси «**ТТ**» №6 (**ТА6**) (**ECDF**).

I03N6 (PPRI7) - признак выхода в «0» для оси «**ТТ**» №7 (**ТА7**) (**ECDF**).

I03N7 (PPRI8) - признак выхода в «0» для оси «**ТТ**» №8 (**ТА8**) (**ECDF**).

24.3.5. Разъём 04N (30N-56N-82N-108N)

W04N0-W04N3 - текущая позиция для оси «**ТТ**», номер которой установлен в слове **W20K0** (первый канал). Позиция представлена в коде **BCD** в формате 5.3 (**ECDF**).

24.3.6. Разъём 05N (31N-57N-83N-109N)

W05N0-W05N3 - текущая позиция для оси «**ТТ**», номер которой установлен в слове **W20K1** (второй канал). Позиция представлена в коде **BCD** в формате 5.3 (**ECDF**).

24.3.7. Разъём 06N (32N-58N-84N-110N)

W06N1

- байт служит для передачи в ПЛ информации о номере диапазона вращения шпинделя, устанавливается системой значением от 1 до 99 в зависимости от запрограммированной функции и инструкции **GMnn**, установленной при характеристике в файле **AXCFIL (GM1-GM99)**. ПЛ должна сообщить системе посредством байта **W16N1** о том, какой диапазон подключён на станке. Обновление аналогового сигнала шпинделя, функции «S», переданной в коде **BCD** в слова **W08K0, W08K1, W08K2**, и строба **FUAS** при смене диапазона выполняется в течение двух циклов ПЛ после обновления байта **W6N1**. Если имеется только один диапазон вращения шпинделя, ПЛ должна устанавливать бит соответственно диапазону 1 (**GAMMA1** или **W16N1=1**).

24.3.8. Разъём 10N (36N-62N-88N-114N)

U10N0 (CUB)

- запрос контроля объёмной защищённой зоны (**ECDF**).

U10N1 (Lox)

- запрос переключения зон **LO1** в **LO2 (ECDF)** :

- **U10N1=0** - активна зона определенная в **LO1**;
- **U10N1=1** - активна зона определенная в **LO2**.

U10N2 (ACC1)

U10N3 (ACC2)

} - запрос переключения законов ускорения (**ECDF**) :

U10N2	U10N3	тип ускорения
«0»	«0»	линейный;
«1»	«0»	S-образный;
«0»	«1»	экспоненциальный;
«1»	«1»	линейный.

U10N16-U10N23

(PPFOLD1-PPFOLD8)

} - эти сигналы, установленные в «1», прерывают любое перемещение соответствующей оси «**TT**» №1-№8 (**TA1-TA8**) (**ECDF**).

24.3.9. Разъём 15N (41N-67N-93N-119N)

W15N0

- двоичный код номера оси, управляемой штурвалом по каналу 1 (порядок осей определён инструкцией **MAS** файла **PGCFIL**). Сигнал **U15N7** - «ось не выбрана» должен быть установлен в состояние «1», если биты **U15N0-U15N6** равны «0».

Примечание - Номер канала штурвала определён инструкцией **ADV** (секция 3 файла **IOCFIL**).

W15N1

- двоичный код номера оси, управляемой штурвалом по каналу 2 (порядок осей определён инструкцией **MAS** файла **PGCFIL**). Сигнал **U15N15** - «ось не выбрана»

должен быть установлен в состояние «1», если биты **U15N8 – U15N14** равны «0».

Примечание - Номер канала штурвала определён инструкцией **ADV** (секция 3 файла **IOCFIL**).

W15N2 - младшие четыре бита этого слова задают шкалу работы штурвала на канале №1, а старшие четыре бита задают шкалу работы штурвала на канале №2. Задание значения шкалы обязательно.

U15N24 - запрос работы со штурвалом на канале №1. Данный запрос можно устанавливать, если:

- выбрана шкала штурвала (в младших четырёх битах байта **W15N2**);
- выбрана ось для движения (**/[W15N0=128] * [W15N0>0]**).

U15N25 - запрос работы со штурвалом на канале №2. Данный запрос можно устанавливать, если:

- выбрана шкала штурвала (в старших четырёх битах байта **W15N2**);
- выбрана ось для движения (**/[W15N0=128] * [W15N0>0]**).

U15N31 (HNDWP) - запрос работы со штурвалами некоторого процесса из видеостраниц других процессов.

Сигнал соответствующего процесса (**U15N31, U41N31, U67N31, U93N31, U119N31**), установленный в состояние «1», разрешает управлять движением осей данного процесса от штурвалов, заявленных в инструкции **ADV** данного процесса, из видеостраниц любого процесса, в том числе, и из видеостраниц **СУПЕРВИЗОРА**.

Сигнал соответствующего процесса (**U15N31, U41N31, U67N31, U93N31, U119N31**), установленный в состояние «0», разрешает управлять движением осей данного процесса от штурвалов, заявленных в инструкции **ADV** данного процесса, из видеостраниц только данного процесса.

24.3.10. Разъём 16N (42N-68N-94N-120N)

U16N0 (SMRIF) - сигнал устанавливает режим работы шпинделя: 1 оборот шпинделя - **N** оборотов датчика. Сигнал используется для обеспечения нарезания резьбы и его ориентации по концевому микроулы шпинделя и используется обычно, если датчик шпинделя закреплен на валу его двигателя.

U16N1 (SMCZ) - сигнал устанавливается при наезде на микроконцевик и сбрасывается сразу при съезде с микроконцевика шпинделя, при активном сигнале **SMRIF**.

Для ориентации шпинделя важно обеспечить после включения УЧПУ принудительное вращение двух оборотов шпинделя всегда в одном направлении.

U16N2 (RIFOR)

- состояние сигнала **RIFOR** определяет два алгоритма ориентации шпинделя по запросу сигналом **ANGOM**:

- 1) **RIFOR=0** алгоритм выполняется с использованием накопителя импульсов датчика шпинделя (стандартный алгоритм). «Ноль-метка» датчика определяется при первом обороте вращения или первой ориентации шпинделя и далее ориентация выполняется как позиционирование в координату, учитывающую значение смещения (первый параметр инструкции **POM** и разъем «14K»).

Если шпиндель после включения станка не вращался, то первая ориентация выполняется на скорости, определённой вторым параметром **POM**. Направление ориентации зависит от состояния сигналов **ORIMO (U11K6)** и **ORIMA (U11K7)**. Если состояние сигналов **U11K6** и **U11K7** равно «0», то направление ориентации шпинделя будет соответствовать направлению вращения по сигналу **ROMOO**, последующие ориентации выполняются с разгоном и торможением по значениям параметров инструкции **TSM** по кратчайшему пути.

Если состояние одного из сигналов **U11K6** или **U11K7** не равно «0», то направление ориентации шпинделя будет соответствовать направлению вращения, которое задано одним из них.

- 2) **RIFOR=1** алгоритм выполняется всегда с поиском «ноль-метки» датчика шпинделя. «Ноль-метка» датчика определяется при каждой ориентации. Ориентация всегда выполняется на скорости, определённой вторым параметром **POM**, и всегда в одном направлении. Накопитель импульсов датчика шпинделя при нахождении «ноль-метки» датчика обнуляется, после чего обрабатывается смещение, определённое первым параметром **POM** и значением разъёма «14K», по кратчайшему пути с разгоном и торможением по значениям параметров инструкции **TSM**.

Значение скорости ориентации должно быть в пределах V_{POMmin} и V_{POMmax} , которые рассчитываются по формулам:

$$V_{POMmin} = 60000 / (\text{электрический шаг} / \text{механический шаг}) / \text{ТИК}$$

$$V_{POMmax} = \pm 32768 / (\text{электрический шаг} / \text{механический шаг}) ,$$

где:

электрический шаг	- первый параметр инструкции PAS для шпинделя в секции 2 файла AXCFIL ;
механический шаг	- второй параметр инструкции PAS для шпинделя в секции 2 файла AXCFIL ;
ТИК	- значение ТИКа в инструкции CAS секции 1 файла AXCFIL .

Примечание - Инструкция **PAS** секции 2 файла **AXCFIL** и инструкции **CAS** секции 1 файла **AXCFIL** описаны в документе «Руководство по характеристикам».

Кроме того, что значение **VROM** должно быть в промежутке между значениями **VROMmin** и **VROMmax**, оно должно быть больше дрейфа на шпинделе.

ВНИМАНИЕ! СИГНАЛ **RIFOR (U16N2)** ДОЛЖЕН БЫТЬ УСТАНОВЛЕН В СОСТОЯНИЕ «0» ИЛИ «1» ДО ЗАПРОСА ОРИЕНТАЦИИ СИГНАЛОМ **U11K0 (ANGOM)** И ДОЛЖЕН УДЕРЖИВАТЬСЯ В ЭТОМ СОСТОЯНИИ НА ВСЁ ВРЕМЯ ОРИЕНТАЦИИ. ЕСЛИ ВЫБРАННЫЙ АЛГОРИТМ ОРИЕНТАЦИИ ДОЛЖЕН БЫТЬ ЕДИНСТВЕННЫМ, ТО СИГНАЛ **U16N2** МОЖНО УСТАНОВИТЬ В СООТВЕТСТВУЮЩЕЕ СОСТОЯНИЕ «1» ИЛИ «0» НА ВСЁ ВРЕМЯ РАБОТЫ УЧПУ, НАПРИМЕР: **U16N2=I0K2**.

U16N3 (TRMTR)

- состояние сигнала **TRMTR** определяет два алгоритма определения скорости вращения шпинделя.

- 1) При состоянии сигнала **U16N3=0** система определяет, что один оборот шпинделя равен одному обороту его датчика обратной связи.
- 2) При состоянии сигнала **U16N3=1** система определяет, что один оборот шпинделя не равен одному обороту его датчика обратной связи. В этом случае необходимо сформировать значения в инструкции **PAS** так, чтобы индикация скорости вращения шпинделя в поле **S** на первом его диапазоне (**GM1**) соответствовала заданной в кадре. Для диапазонов шпинделя, начиная со второго, индикация скорости его вращения будет пересчитана с учётом отношения скорости и напряжения, установленных в инструкциях **GMnn**, к скорости и напряжению первого диапазона **GM1**.

W16N1

- номер диапазона шпинделя, действительно установленный в станке; **nn** может принимать значения от 1 до 99. Число **nn** указывает системе на номер инструкции **GMnn** оси шпинделя в файле **AXCFIL**, которая опре-

деляет рабочую шкалу «Скорость - напряжение». Например, если в станке установлен диапазон 24, то байту **W16N1** необходимо присвоить число 24 (**W16N1=24D**), при этом в системе будет активна шкала **GM24=XXX,YY,.**

Данный способ определения номера диапазона шпинделя существует параллельно с имеющимся способом, который использует сигналы **U11K16-U11K19**.

Номер диапазона (число > 0), записанный в байте **W16N1**, имеет в системе более высокий приоритет по отношению к сигналам **U11K16-U11K19**.

ПЛ должна устанавливать байт **W16N1** в «0» только при **ABIM=0**.

24.4. Таблица базовых сигналов пакета «N»

В таблице 24.2 представлены базовые сигналы пакета «N» интерфейса PLC.

Таблица 24.2 – Базовые сигналы пакета «N» интерфейса PLC

Разъём: I00N (26N-52N-78N-104N)			Сигналы Про -> PLC	
Байт	Бит	Сигнал	Назначение	
0	0	АСUB	3 оси находятся в объёмной защищённой зоне СUB (версия Про ...РИБ)	
	1			
	2			
	3			
	4			
	5			
	6			
	7			
1	8		UAS	В этих разрядах фиксируются состояния трёхбуквенных кодов, используемых в процессе выполнения УП
	9		USB	
	10		USO	
	11		UVR	
	12		URL	
	13		MBR	
	14		RCM	
	15		VOL	
2	16		RAP	
	17		UEP	
	18			
	19			
	20			
	21			
	22			
	23			
3	24			
	25			
	26			
	27			
	28			
	29			
	30			
	31			

Продолжение таблицы 24.2

Разъём: I01N (27N-53N-79N-105N)			Сигналы ПрО -> PLC	
Байт	Бит	Сигнал	Назначение	
0	0		1%	В этом байте записываются значения %, установленные на корректоре подачи «JOG» с округлением до целых
	1		2%	
	2		4%	
	3		8%	
	4		16%	
	5		32%	
	6		64%	
	7		Направление движения I01N7=1 - отрицательное	
1	8		1%	В этом байте записываются значения %, установленные на корректоре подачи «F» с округлением до целых
	9		2%	
	10		4%	
	11		8%	
	12		16%	
	13		32%	
	14		64%	
	15			
2	16		1%	В этом байте записываются значения %, установленные на корректоре оборотов шпинделя «S» с округлением до целых
	17		2%	
	18		4%	
	19		8%	
	20		16%	
	21		32%	
	22		64%	
	23			
3	24	MDI	NC-201/201M/202/210/220/230: «1» в одном из битов этого байта соответствует текущему положению переключателя режимов работы «MDI», ..., «RESET».	NC-110/310: «1» в одном из битов этого байта соответствует последней нажатой на станочном пульте кнопке выбора режима.
	25	AUTO		
	26	STEP		
	27	MANU		
	28	MANJ		
	29	PROF		
	30	HOME		
	31	RESET		

Продолжение таблицы 24.2

Разъём: I03N (29N-55N-81N-107N)		Сигналы Про -> PLC	
Байт	Бит	Сигнал	Назначение
0	0	PPRI1	Признак выхода в «0» для оси «ТТ» №1 (ТА1) (ECDF)
	1	PPRI2	Признак выхода в «0» для оси «ТТ» №2 (ТА2) (ECDF)
	2	PPRI3	Признак выхода в «0» для оси «ТТ» №3 (ТА3) (ECDF)
	3	PPRI4	Признак выхода в «0» для оси «ТТ» №4 (ТА4) (ECDF)
	4	PPRI5	Признак выхода в «0» для оси «ТТ» №5 (ТА5) (ECDF)
	5	PPRI6	Признак выхода в «0» для оси «ТТ» №6 (ТА6) (ECDF)
	6	PPRI7	Признак выхода в «0» для оси «ТТ» №7 (ТА7) (ECDF)
	7	PPRI8	Признак выхода в «0» для оси «ТТ» №8 (ТА8) (ECDF)
1	8		
	9		
	10		
	11		
	12		
	13		
	14		
	15		
2	16		
	17		
	18		
	19		
	20		
	21		
	22		
	23		
3	24		
	25		
	26		
	27		
	28		
	29		
	30		
	31		

Продолжение таблицы 24.2

Разъём: I04N			Сигналы ПрО -> PLC	
Байт	Бит	Сигнал	Назначение	
0	0	PPCP	0,00X	Текущая позиция оси «ТТ» в первом канале. Код VCD в формате 5.3 (ECDF).
	1	PPCP		
	2	PPCP		
	3	PPCP		
	4	PPCP	0,0X	
	5	PPCP		
	6	PPCP		
1	7	PPCP	0,X	
	8	PPCP		
	9	PPCP		
	10	PPCP		
	11	PPCP	X	
	12	PPCP		
	13	PPCP		
	14	PPCP		
2	15	PPCP	X0	
	16	PPCP		
	17	PPCP		
	18	PPCP	X00	
	19	PPCP		
	20	PPCP		
	21	PPCP		
	22	PPCP		
3	23	PPCP	X000	
	24	PPCP		
	25	PPCP		
	26	PPCP	X0000	
	27	PPCP		
	28	PPCP		
	29	PPCP		
	30	PPCP		
	31	PPCP		

Продолжение таблицы 24.2

Разъём: I05N (31N-57N-83N-109N)			Сигналы ПрО -> PLC	
Байт	Бит	Сигнал	Назначение	
0	0	PPCP	0,00X	Текущая позиция оси «ТТ» во втором канале. Код BCD в формате 5.3 (BCDF).
	1	PPCP		
	2	PPCP		
	3	PPCP		
	4	PPCP	0,0X	
	5	PPCP		
	6	PPCP		
	7	PPCP		
1	8	PPCP	0,X	
	9	PPCP		
	10	PPCP		
	11	PPCP		
	12	PPCP	X	
	13	PPCP		
	14	PPCP		
	15	PPCP		
2	16	PPCP	X0	
	17	PPCP		
	18	PPCP		
	19	PPCP		
	20	PPCP	X00	
	21	PPCP		
	22	PPCP		
	23	PPCP		
3	24	PPCP	X000	
	25	PPCP		
	26	PPCP		
	27	PPCP		
	28	PPCP	X0000	
	29	PPCP		
	30	PPCP		
	31	PPCP		

Продолжение таблицы 24.2

Разъём: I06N (32N-58N-84N-110N)			Сигналы ПрО -> PLC
Байт	Бит	Сигнал	Назначение
0	0		
	1		
	2		
	3		
	4		
	5		
	6		
	7		
1	8		Номер диапазона шпинделя, определяемый системой по заданной скорости вращения шпинделя с адресом S. Номер представлен десятичным числом от 1 до 99.
	9		
	10		
	11		
	12		
	13		
	14		
	15		
2	16		
	17		
	18		
	19		
	20		
	21		
	22		
	23		
3	24		
	25		
	26		
	27		
	28		
	29		
	30		
	31		

Продолжение таблицы 24.2

Разъём: U10N (36N-62N-88N-114N)			Сигналы PLC -> Про
Байт	Бит	Сигнал	Назначение
0	0	CUB	Запрос контроля объёмной защищённой зоны (ECDF)
	1	LOX	Запрос переключения зон LO1 в LO2 (ECDF)
	2	ACC1	Запрос на установку одного из законов разгона/торможения (ECDF)
	3	ACC2	
	4		
	5		
	6		
	7		
1	8		
	9		
	10		
	11		
	12		
	13		
	14		
	15		
2	16	PPFOLD1	Запрет подачи для оси « ТТ » №1 (ТА1) (ECDF)
	17	PPFOLD2	Запрет подачи для оси « ТТ » №2 (ТА2) (ECDF)
	18	PPFOLD3	Запрет подачи для оси « ТТ » №3 (ТА3) (ECDF)
	19	PPFOLD4	Запрет подачи для оси « ТТ » №4 (ТА4) (ECDF)
	20	PPFOLD5	Запрет подачи для оси « ТТ » №5 (ТА5) (ECDF)
	21	PPFOLD6	Запрет подачи для оси « ТТ » №6 (ТА6) (ECDF)
	22	PPFOLD7	Запрет подачи для оси « ТТ » №7 (ТА7) (ECDF)
	23	PPFOLD8	Запрет подачи для оси « ТТ » №8 (ТА8) (ECDF)
3	24		
	25		
	26		
	27		
	28		
	29		
	30		
	31		

Продолжение таблицы 24.2

Разъём: U15N (41N-67N-93N-119N)			Сигналы Про -> PLC		
Байт	Бит	Сигнал	Назначение		
0	0		Двоичный код номера оси, управляемой штурвалом на канале 1 (порядок осей определён инструкцией MAS файла PGCFIL)		
	1				
	2				
	3				
	4				
	5				
	6				
		7		Ось не выбрана (U15N7=1)	
1	8		Двоичный код номера оси, управляемой штурвалом на канале 2 (порядок осей определён инструкцией MAS файла PGCFIL)		
	9				
	10				
	11				
	12				
	13				
	14				
		15		Ось не выбрана (U15N15=1)	
2	16		0.01 мм/0.001"	Шкала работы штурвала на канале 1	
	17		0.10 мм/0.010"		
	18		1.00 мм/0.100"		
	19		10.00 мм/1.000"		
		20		0.01 мм/0.001"	Шкала работы штурвала на канале 2
	21		0.10 мм/0.010"		
	22		1.00 мм/0.100"		
	23		10.00 мм/1.000"		
3	24		Запрос работы со штурвалом на канале 1		
	25		Запрос работы со штурвалом на канале 2		
	26				
	27				
	28				
	29				
	30				
		31	HNDWP	Разрешение работы штурвалов процесса n из видеостраниц других процессов	

Продолжение таблицы 24.2

Разъём: U16N (42N-68N-94N-120N) Сигналы PLC -> Про			
Слово	Бит	Сигнал	Назначение
0	0	SMRIF	устанавливает режим шпинделя: 1 оборот шпинделя - N оборотов датчика
	1	SMCZ	1. Запрос поиска ближайшей «ноль-метки» шпинделя при активном сигнале SMRIF при резьбонарезании; 2. Запрос поиска точки ориентации, которая была определена УЧПУ при первом вращении шпинделя. Сигнал имеет значение только при активном сигнале SMRIF .
	2	RIFOR	Алгоритм ориентации шпинделя: RIFOR=0 - с использованием накопителя импульсов с датчика шпинделя. RIFOR=1 - ориентирование шпинделя по «ноль-метке» датчика шпинделя.
	3	TRMTR	Датчик шпинделя на двигателе
	4		
	5		
	6		
	7		
1	8		Номер диапазона шпинделя (от 1 до 99), действительно установленный в станке. Записывается в десятичном виде.
	9		
	10		
	11		
	12		
	13		
	14		
15			
2	16		
	17		
	18		
	19		
	20		
	21		
	22		
23			
3	24		
	25		
	26		
	27		
	28		
	29		
	30		
31			

25. СИГНАЛЫ СТАНОЧНОГО ПУЛЬТА

25.1. Сигналы программируемых кнопок СП NC110-7

25.1.1. Разъёмы 157К-160К

В зоне пакета «К» для УЧПУ NC-110 выделена зона из 4 разъёмов **157К - 160К** по 32 сигнала каждый для управления станком со станочного пульта.

Сигналы разъёмов **157К - 160К** разбиты на две группы:

- 1) кнопки (разъём **157К** + 14 бит разъёма **158К**);
- 2) лампочки (разъём **159К** + 14 бит разъёма **160К**).

Примечание - Все кнопки выполнены по алгоритму: нажата - «1», отжата - «0».

Сигналы разъёмов **157К-160К** представлены в таблице 25.1. Нумерация кнопок и лампочек СП NC-110 в таблице 25.1 соответствует рисунку 25.1.

Кнопки №1 («ПУСК») и №2 («СТОП») запрограммированы из ПрО и не имеют сигналов в таблице 25.1.

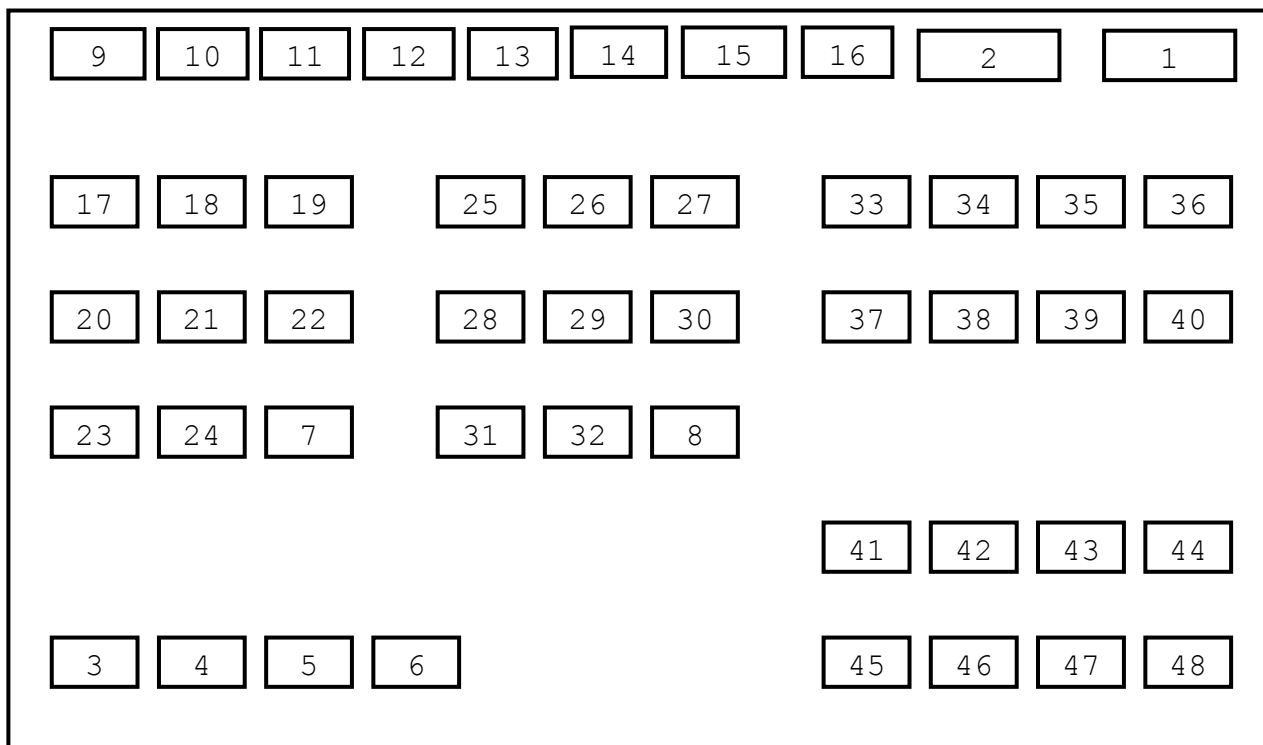


Рисунок 25.1 - Нумерация кнопок и лампочек СП NC110-7

Таблица 25.1 - Сигналы пакета «К» интерфейса PLC для СП NC110-7

Разъём: 157К			
Байт	Бит	Сигнал	Назначение
0	0	43	Кнопка № 3
	1	44	Кнопка № 4
	2	45	Кнопка № 5
	3	46	Кнопка № 6
	4	47	Кнопка № 7
	5	48	Кнопка № 8
	6	MDI (49)	Кнопка № 9 (запрограммирована из ПрО)
	7	AUTO (4A)	Кнопка № 10 (запрограммирована из ПрО)
1	8	STEP (4B)	Кнопка № 11 (запрограммирована из ПрО)
	9	MANU (4C)	Кнопка № 12 (запрограммирована из ПрО)
	10	MANJ (4D)	Кнопка № 13 (запрограммирована из ПрО)
	11	PROF (4E)	Кнопка № 14 (запрограммирована из ПрО)
	12	HOME (4F)	Кнопка № 15 (запрограммирована из ПрО)
	13	RESET (50)	Кнопка № 16 (запрограммирована из ПрО)
	14	51	Кнопка № 17
	15	52	Кнопка № 18
2	16	53	Кнопка № 19
	17	54	Кнопка № 20
	18	55	Кнопка № 21
	19	56	Кнопка № 22
	20	57	Кнопка № 23
	21	58	Кнопка № 24
	22	59	Кнопка № 25
	23	5A	Кнопка № 26
3	24	5B	Кнопка № 27
	25	5C	Кнопка № 28
	26	5D	Кнопка № 29
	27	5E	Кнопка № 30
	28	5F	Кнопка № 31
	29	60	Кнопка № 32
	30	61	Кнопка № 33
	31	62	Кнопка № 34

Продолжение таблицы 25.1

Разъём: 158К			
Байт	Бит	Сигнал	Назначение
0	0	63	Кнопка № 35
	1	64	Кнопка № 36
	2	65	Кнопка № 37
	3	66	Кнопка № 38
	4	67	Кнопка № 39
	5	68	Кнопка № 40
	6	69	Кнопка № 41
	7	6A	Кнопка № 42
1	8	6B	Кнопка № 43
	9	6C	Кнопка № 44
	10	6D	Кнопка № 45
	11	6E	Кнопка № 46
	12	6F	Кнопка № 47
	13	70	Кнопка № 48
	14		
	15		
2	16		
	17		
	18		
	19		
	20		
	21		
	22		
	23		
3	24		
	25		
	26		
	27		
	28		
	29		
	30		
	31		

Продолжение таблицы 25.1

Разъём: 159К			
Байт	Бит	Сигнал	Назначение
0	0		Лампочка № 3
	1		Лампочка № 4
	2		Лампочка № 5
	3		Лампочка № 6
	4		Лампочка № 7
	5		Лампочка № 8
	6		Лампочка № 9 (запрограммирована из ПрО)
	7		Лампочка № 10 (запрограммирована из ПрО)
1	8		Лампочка № 11 (запрограммирована из ПрО)
	9		Лампочка № 12 (запрограммирована из ПрО)
	10		Лампочка № 13 (запрограммирована из ПрО)
	11		Лампочка № 14 (запрограммирована из ПрО)
	12		Лампочка № 15 (запрограммирована из ПрО)
	13		Лампочка № 16 (запрограммирована из ПрО)
	14		Лампочка № 17
	15		Лампочка № 18
2	16		Лампочка № 19
	17		Лампочка № 20
	18		Лампочка № 21
	19		Лампочка № 22
	20		Лампочка № 23
	21		Лампочка № 24
	22		Лампочка № 25
	23		Лампочка № 26
3	24		Лампочка № 27
	25		Лампочка № 28
	26		Лампочка № 29
	27		Лампочка № 30
	28		Лампочка № 31
	29		Лампочка № 32
	30		Лампочка № 33
	31		Лампочка № 34

Продолжение таблицы 25.1

Разъём: 160К			
Байт	Бит	Сигнал	Назначение
0	0		Лампочка № 35
	1		Лампочка № 36
	2		Лампочка № 37
	3		Лампочка № 38
	4		Лампочка № 39
	5		Лампочка № 40
	6		Лампочка № 41
	7		Лампочка № 42
1	8		Лампочка № 43
	9		Лампочка № 44
	10		Лампочка № 45
	11		Лампочка № 46
	12		Лампочка № 47
	13		Лампочка № 48
	14		
	15		
2	16		
	17		
	18		
	19		
	20		
	21		
	22		
	23		
3	24		
	25		
	26		
	27		
	28		
	29		
	30		
	31		

25.2. Сигналы СП УЧПУ NC-200, NC-201, NC-201M, NC-202, NC-210, NC-220, NC-230

25.2.1. Разъём 02N

25.2.1.1. Разъём 02N для NC-200, NC-210, NC-220, NC-230


В УЧПУ NC-200, NC-210, NC-220, NC-230 функции СП реализуют функциональные клавиши «F1»-«F8», «F11»-«F18». Все клавиши выполнены по алгоритму: нажата - «1», отжата - «0». В пакете «N» для них выделен разъём «02N».

W02N0 - в этом слове из Про устанавливается сигнал, соответствующий одной из нажатых клавиш «F1»-«F8».


W02N1 - в этом слове из Про устанавливается сигнал, соответствующий одной из нажатых клавиш «F11»-«F18».


Сигналы разъёма **02N** для обслуживания функциональных клавиш «F1»-«F8» и «F11»-«F18» УЧПУ NC-200/210/220/230 представлены в таблице 25.2. Сигналы таблицы 25.2 соответствуют также сигналам обслуживания функциональных клавиш «F1»-«F8» и «F11»-«F18» УЧПУ NC-310.

25.2.1.2. Разъём 02N для NC-201, NC-201M, NC-202

В УЧПУ NC-201/201M/202 функции СП реализуют функциональные клавиши «F1»-«F8» и клавиши «+X», «-X», «+Z», «-Z», «», «+Y», «-Y». Все клавиши выполнены по алгоритму: нажата - «1», отжата - «0». В пакете «N» для них выделен разъём **02N**.

W02N0 - в этом слове из Про устанавливается сигнал, соответствующий одной из нажатых клавиш «F1»-«F8».

W02N1 - в этом слове из Про устанавливается сигнал, соответствующий одной из нажатых клавиш: «+X», «-X», «+Z», «-Z», «», «+Y», «-Y».

Сигналы разъёма **02N** для обслуживания функциональных клавиш «F1»-«F8» и клавиш «+X», «-X», «+Z», «-Z», «», «+Y», «-Y». УЧПУ NC-201/201M/202 представлены в таблице 25.3.



Выполнение действий на станке при использовании клавиш «+X», «-X», «+Z», «-Z», «», «+Y», «-Y» программируется в ПЛ с помощью сигналов разъёма **15K** и сигнала **U10K4**. Пример использования приведён в программе **LNC201/MP0**.

Таблица 25.2 - Сигналы пакета «N» для СП УЧПУ NC-200/210/220/230/310

Разъём: I02N			
Байт	Бит	Сигнал	Назначение
0	0		Нажата кнопка «F1»
	1		Нажата кнопка «F2»
	2		Нажата кнопка «F3»
	3		Нажата кнопка «F4»
	4		Нажата кнопка «F5»
	5		Нажата кнопка «F6»
	6		Нажата кнопка «F7»
	7		Нажата кнопка «F8»
1	8		Нажата кнопка «F11»
	9		Нажата кнопка «F12»
	10		Нажата кнопка «F13»
	11		Нажата кнопка «F14»
	12		Нажата кнопка «F15»
	13		Нажата кнопка «F16»
	14		Нажата кнопка «F17»
	15		Нажата кнопка «F18»
2	16		
	17		
	18		
	19		
	20		
	21		
	22		
	23		
3	24		
	25		
	26		
	27		
	28		
	29		
	30		
	31		

Таблица 25.3 - Сигналы пакета «N» для СП УЧПУ NC-201/201M/202

Разъем: I02N			
Байт	Бит	Сигнал	Назначение
0	0		Нажата кнопка «F1»
	1		Нажата кнопка «F2»
	2		Нажата кнопка «F3»
	3		Нажата кнопка «F4»
	4		Нажата кнопка «F5»
	5		Нажата кнопка «F6»
	6		Нажата кнопка «F7»
	7		Нажата кнопка «F8»
1	8		Нажата кнопка (F11) : NC-201/202: «+X»; NC-201M : «+Z».
	9		Нажата кнопка (F12) : NC-201/202: «-X»; NC-201M : «-Z».
	10		Нажата кнопка (F13) : NC-201/202: «+Z»; NC-201M : «+X».
	11		Нажата кнопка (F14) : NC-201/202: «-Z»; NC-201M : «-X».
	12		Нажата кнопка (F15) быстрого хода  для NC-201/201M/202
	13		Нажата кнопка (F16) : NC-201M: «+Y».
	14		Нажата кнопка (F17) : NC-201M: «-Y».
	15		
2	16		
	17		
	18		
	19		
	20		
	21		
	22		
	23		
3	24		
	25		
	26		
	27		
	28		
	29		
	30		
	31		

25.2.2. Разъёмы 11N, 12N, 13N, 14N

1) Сигналы разъёмов **11N, 12N** используются для создания станочного меню в видеокадре **#7** УЧПУ NC-200/201/201M/202/210/220/230/310:

W11N0-W11N3 }
W12N0-W12N3 } - Данные байты предназначены для выполнения запросов вывода текста из файла **LGCMNU/MP0** в поля над клавишами «**F1**»-«**F8**» в видеостранице **#7**. В файле **LGCMNU/MP0** допускается максимум 255 текстовых записей до 16 символов в каждой строке с указанием кода цвета текста. Для того чтобы нужный текст появился в поле над одной из клавиш «**F1**»-«**F8**», необходимо в соответствующий этому полю байт записать порядковый номер строки, на которой данный текст расположен в файле **LGCMNU/MP0**.

Файл **LGCMNU** создаётся командой **EDI**. Формат строк файла **LGCMNU/MP0**:

Код цвета#текст ,

где:

Код цвета	- десятичное число от 0 до 15 , определяющее цвет текста в полях над клавишами « F1 »-« F8 » Соответствие цвета и его кода приведено в документе «Руководстве программиста» в разделе 5 «Программирование на языке ASSET»;
#	- символ разделителя. Не допускается символ <пробел> до и после знака разделителя;
Текст	- 16 символов. Поле индикации меню для клавиш « F1 »-« F8 » состоит из двух строк по 8 символов в каждой.

2) Сигналы разъёмов **13N, 14N** используются для создания вертикальных иконок в видеокадре **#7** станочного меню УЧПУ NC-200/201/201M/202/210/220/230/310:

W13N0-W13N3 }
W14N0-W14N3 } - Данные байты предназначены для выполнения запросов вывода иконок СП из файла **C:\CNC32\MP0\CNC.USR** в поля, расположенные в видеостранице **#7** слева от клавиш «**F11**»-«**F18**». В файле **CNC.USR** допускается максимум 255 иконок. Для появления нужной иконки в поле слева от одной из клавиш «**F11**»-«**F18**», необходимо в соответствующий этому полю байт записать порядковый номер этой иконки.

Примечание - Создание файла **CNC.USR** рассмотрено в разделе 26.2.

3) Сигналы пакета «**N**» для создания станочного меню и вертикальных иконок в видеокадре **#7** УЧПУ NC-200/201/201M/202/210/220/230/310 сведены в таблицу 25.4. Для УЧПУ NC-201/201M/202 соответствие кнопок «токарного креста» обозначениям **F11-F17** указано в таблице 25.3.

Таблица 25.4 – Сигналы пакета «N» для станочного меню и вертикальных иконок видеокadra #7 NC-200/201/201M/202/210/220/230/310

Разъём: 11N					
Байт	Бит	Сигнал	Назначение		
0	0		1	Номер записи из файла LGCMNU для её индикации в меню видеокadra #7 над клавишей «F1»	
	1		2		
	2		4		
	3		8		
	4		16		
	5		32		
	6		64		
1	7		128	Номер записи из файла LGCMNU для её индикации в меню видеокadra #7 над клавишей «F2»	
	8		1		
	9		2		
	10		4		
	11		8		
	12		16		
	13		32		
2	14		64	Номер записи из файла LGCMNU для её индикации в меню видеокadra #7 над клавишей «F3»	
	15		128		
	16		1		
	17		2		
	18		4		
	19		8		
	20		16		
3	21		32	Номер записи из файла LGCMNU для её индикации в меню видеокadra #7 над клавишей «F4»	
	22		64		
	23		128		
	24		1		
	25		2		
	26		4		
	27		8		
4	28		16	Номер записи из файла LGCMNU для её индикации в меню видеокadra #7 над клавишей «F5»	
	29		32		
	30		64		
	31		128		
	Разъём: 12N				
	Байт	Бит	Сигнал		Назначение
	0	0			1
1			2		
2			4		
3			8		
4			16		
5			32		
6			64		
1	7		128	Номер записи из файла LGCMNU для её индикации в меню видеокadra #7 над клавишей «F6»	
	8		1		
	9		2		
	10		4		
	11		8		
	12		16		
	13		32		
2	14		64	Номер записи из файла LGCMNU для её индикации в меню видеокadra #7 над клавишей «F7»	
	15		128		
	16		1		
	17		2		
	18		4		
	19		8		
	20		16		
3	21		32	Номер записи из файла LGCMNU для её индикации в меню видеокadra #7 над клавишей «F8»	
	22		64		
	23		128		
	24		1		
	25		2		
	26		4		
	27		8		
4	28		16	Номер записи из файла LGCMNU для её индикации в меню видеокadra #7 над клавишей «F8»	
	29		32		
	30		64		
	31		128		

Продолжение таблицы 25.4

Разъём: 13N			
Байт	Бит	Сигнал	Назначение
0	0		1
	1		2
	2		4
	3		8
	4		16
	5		32
	6		64
	7		128
1	8		1
	9		2
	10		4
	11		8
	12		16
	13		32
	14		64
	15		128
2	16		1
	17		2
	18		4
	19		8
	20		16
	21		32
	22		64
	23		128
3	24		1
	25		2
	26		4
	27		8
	28		16
	29		32
	30		64
	31		128
Разъём: 14N			
Байт	Бит	Сигнал	Назначение
0	0		1
	1		2
	2		4
	3		8
	4		16
	5		32
	6		64
	7		128
1	8		1
	9		2
	10		4
	11		8
	12		16
	13		32
	14		64
	15		128
2	16		1
	17		2
	18		4
	19		8
	20		16
	21		32
	22		64
	23		128
3	24		1
	25		2
	26		4
	27		8
	28		16
	29		32
	30		64
	31		128

25.3. Сигналы СП УЧПУ NC-310

25.3.1. Способы организации СП УЧПУ NC-310

Способы организации станочного пульта УЧПУ NC-310:

- 1) программирование в ПЛ назначения 24 свободных кнопок СП NC310-7;
- 2) программирование в ПЛ меню назначения функциональных клавиш «F1»-«F8» и «F11»-«F18» в видеокадре #7 режима «УПРАВЛЕНИЕ СТАНКОМ».

В первом случае сигналами СП являются сигналы 24 программируемых кнопок СП NC310-7. Во втором случае сигналами СП являются сигналы двух групп функциональных клавиш «F1»-«F8» и «F11»-«F18» на панели ПО.

Для организации СП УЧПУ NC-310 можно использовать либо один из указанных способов, либо два способа одновременно.

25.3.2. Сигналы программируемых кнопок СП NC310-7

Для сигналов СП NC310-7 в зоне пакета «К» выделены 2 разъёма **157К** и **159К** по 32 сигнала каждый. Сигналы этих двух разъёмов разбиты на две группы:

- 1) кнопки (разъём **157К**);
- 2) лампочки (разъём **159К**).

Примечание - Все кнопки выполнены по алгоритму: нажата - «1», отжата - «0».

Сигналы разъёмов **157К**, **159К** представлены в таблице 25.5. Нумерация кнопок и лампочек СП NC310-7 в таблице 25.5 соответствует рисунку 25.2. Кнопки №1 («ПУСК») и №2 («СТОП») запрограммированы из ПрО и не имеют сигналов в таблице 25.5.

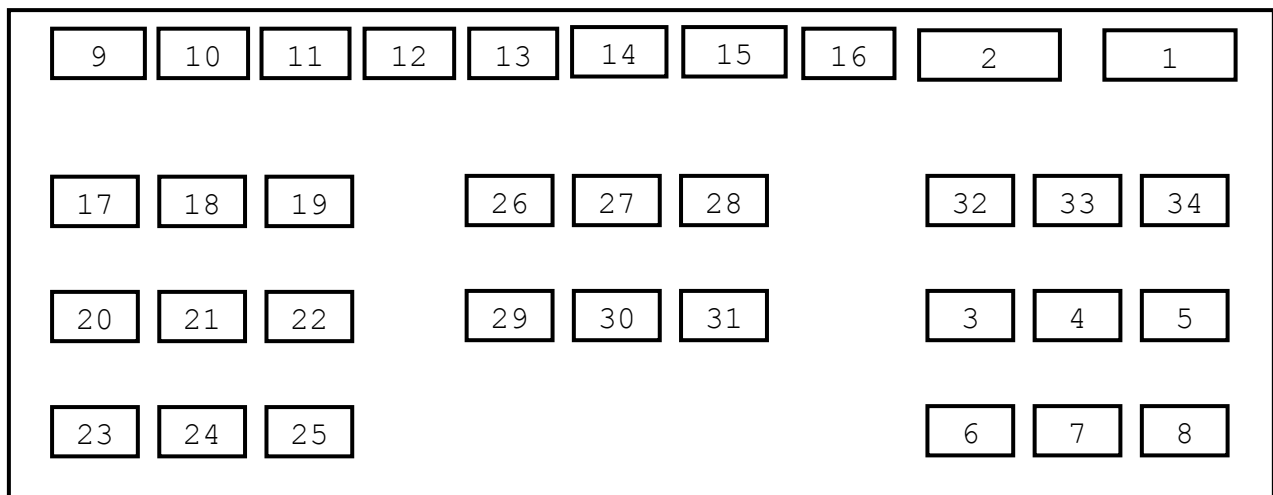


Рисунок 25.2 - Нумерация кнопок и лампочек СП NC310-7

Таблица 25.5 - Сигналы пакета «К» интерфейса PLC для СП NC310-7

Разъём: 157К			
Байт	Бит	Сигнал	Назначение
0	0	43	Кнопка № 3
	1	44	Кнопка № 4
	2	45	Кнопка № 5
	3	46	Кнопка № 6
	4	47	Кнопка № 7
	5	48	Кнопка № 8
	6	MDI (49)	Кнопка № 9 (запрограммирована из ПрО)
	7	AUTO (4A)	Кнопка № 10 (запрограммирована из ПрО)
1	8	STEP (4B)	Кнопка № 11 (запрограммирована из ПрО)
	9	MANU (4C)	Кнопка № 12 (запрограммирована из ПрО)
	10	MANJ (4D)	Кнопка № 13 (запрограммирована из ПрО)
	11	PROF (4E)	Кнопка № 14 (запрограммирована из ПрО)
	12	HOME (4F)	Кнопка № 15 (запрограммирована из ПрО)
	13	RESET (50)	Кнопка № 16 (запрограммирована из ПрО)
	14	51	Кнопка № 17
	15	52	Кнопка № 18
2	16	53	Кнопка № 19
	17	54	Кнопка № 20
	18	55	Кнопка № 21
	19	56	Кнопка № 22
	20	57	Кнопка № 23
	21	58	Кнопка № 24
	22	59	Кнопка № 25
	23	5A	Кнопка № 26
3	24	5B	Кнопка № 27
	25	5C	Кнопка № 28
	26	5D	Кнопка № 29
	27	5E	Кнопка № 30
	28	5F	Кнопка № 31
	29	60	Кнопка № 32
	30	61	Кнопка № 33
	31	62	Кнопка № 34

Продолжение таблицы 25.5

Разъём: 159К			
Байт	Бит	Сигнал	Назначение
0	0		Лампочка № 3
	1		Лампочка № 4
	2		Лампочка № 5
	3		Лампочка № 6
	4		Лампочка № 7
	5		Лампочка № 8
	6		Лампочка № 9 (запрограммирована из ПрО)
1	7		Лампочка № 10 (запрограммирована из ПрО)
	8		Лампочка № 11 (запрограммирована из ПрО)
	9		Лампочка № 12 (запрограммирована из ПрО)
	10		Лампочка № 13 (запрограммирована из ПрО)
	11		Лампочка № 14 (запрограммирована из ПрО)
	12		Лампочка № 15 (запрограммирована из ПрО)
	13		Лампочка № 16 (запрограммирована из ПрО)
2	14		Лампочка № 17
	15		Лампочка № 18
	16		Лампочка № 19
	17		Лампочка № 20
	18		Лампочка № 21
	19		Лампочка № 22
	20		Лампочка № 23
3	21		Лампочка № 24
	22		Лампочка № 25
	23		Лампочка № 26
	24		Лампочка № 27
	25		Лампочка № 28
	26		Лампочка № 29
	27		Лампочка № 30
	28		Лампочка № 31
29		Лампочка № 32	
30		Лампочка № 33	
31		Лампочка № 34	

25.3.3. Сигналы СП УЧПУ NC-310 в видеокадре #7

Сигналы СП УЧПУ NC-310, программируемого в видеокадре #7 в режиме «УПРАВЛЕНИЕ СТАНКОМ», аналогичны сигналам СП УЧПУ NC-210, описанным в п.25.2.

Сигналы разъёма «02N», указанные в таблице 25.2, используются для программирования функциональных клавиш «F1»-«F8» и «F11»-«F18».

Сигналы разъёмов «11N» и «12N» указанные, в таблице 25.4, предназначены для создания в видеокадре #7 станочного меню с использованием функциональных клавиш «F1»-«F8». Сигналы разъёмов «13N» и «14N» указанные, в таблице 25.4, предназначены для создания в видеокадре #7 иконок станочного меню с использованием функциональных клавиш «F11»-«F18».

26. БИБЛИОТЕКА ИКОНОК CNC.USR УЧПУ NC-200, NC-201, NC-201M, NC-202, NC-210, NC-220, NC-230, NC-310 (ВЕРСИИ ПРО 2.60 И ВЫШЕ)

26.1. Файл-архив BMP32.RAR

Файл **CNC.USR** представляет собой библиотеку иконок пользователя, доступных для индикации в области видеокадра **#7** в режиме «**УПРАВЛЕНИЕ СТАНКОМ**», отведённой под СП (слева от клавиш «**F11**»-«**F18**»).

Библиотека иконок создаётся на базе файл-архива **BMP32.RAR**, записанного на дискете **COPYFLASH №0**, входящей в комплект поставки УЧПУ. Файл-архив **BMP32.RAR** содержит:

- иконки с именами **000.BMP-256.BMP**;
- файлы **РИСУНКИ.DOC** и **РИСУНКИ.TXT**;
- файл **USERBMP32.EXE**;
- файл **CNCSYS.PAL**;
- файл **XCHPAL.EXE**.

Назначение файлов файл-архива **BMP32.RAR**:

- файлы **000.BMP-256.BMP**:
 - иконки СП, сохранённые в графическом редакторе как 256-цветный рисунок, размером 56x43 точки каждый;
 - имена рисунков должны быть 3-х символьными номерами, идущими подряд (без пропусков номеров) от **000.BMP** до **255.BMP** максимум. По этим номерам иконки будут выбираться из библиотеки для индикации, как указано в разделе 0;
 - рисунок **000.BMP** является шаблоном, который может использоваться при создании других рисунков, и не подлежит модификации или удалению.
- файлы **РИСУНКИ.DOC** и **РИСУНКИ.TXT**:
 - содержат краткое назначение основных иконок, содержащихся в файл-архиве **BMP32.RAR** на дискете **COPYFLASH №0**.
- файл **USERBMP32.EXE**:
 - преобразует иконки с именами **000.BMP-256.BMP** в текущем каталоге в единый файл **CNC.USR**;
 - автоматически добавляет новые цвета, используемые при создании иконок в графическом редакторе, в файл-палитру иконок **CNCSYS.PAL** и выводит на экран ПК уведомление, если были внесены изменения в файле **CNCSYS.PAL**.

Примечание - Программа **USERBMP32.EXE** применяется также и для преобразования рисунков встроенных постоянных циклов визуального программирования.

ВНИМАНИЕ! НЕ ПРИМЕНЯТЬ ПРОГРАММУ **USERBMP32.EXE** ДЛЯ СОЗДАНИЯ ФАЙЛА **CNC.USR** ДЛЯ ВЕРСИЙ ПРО ДО НОМЕРА **Z.60!**

- файл **CNCSYS.PAL**:

- хранит цветовую палитру иконок СП, а также рисунки встроенных постоянных циклов визуального программирования.

Примечание - Файл **CNCSYS.PAL** в ПК может быть не в текущем каталоге, где расположена программа **USERBMP32.EXE**, а в корневом каталоге диска **C:**.

Это может быть удобным, если файл **CNCSYS.PAL** используется не только для цветовой палитры иконок станочного пульта, но и для рисунков встроенных циклов визуального программирования.

- файл **XCHPAL.EXE**:

- преобразует цветовую палитру в иконках, которые были созданы для версий ПрО с номерами до **Z.60** в цветовую палитру, используемую в версиях ПрО, начиная с номера **Z.60**.

26.2. Создание библиотеки иконок CNC.USR

Процедура создания файла **CNC.USR**:

- 1) создать на ПК отдельный каталог;
- 2) создать библиотеку иконок СП в этом каталоге, для чего:
 - скопировать в созданный каталог файл-архив **BMP32.RAR** с дискеты **COPYFLASH №0** из каталога **BMP32**;
 - выполнить распаковку файл-архива **BMP32.RAR** с помощью программы-архиватора **RAR.EXE**; в результате распаковки файл-архива **BMP32.EXE** будут созданы:
 - иконки с именами **000.BMP-256.BMP**;
 - файлы **РИСУНКИ.DOC** и **РИСУНКИ.TXT**, **USERBMP32.EXE**, **CNCSYS.PAL**, **XCHPAL.EXE**.
- 3) выполнить следующие действия для создания индивидуальных иконок конкретного станка:
 - отредактировать файлы **000.BMP-256.BMP**, используя любой графический редактор, позволяющий сохранять рисунки в формате **BMP**, например, **PAINT (PBRUSH)**;

- выполнить файл **USERBMP32.EXE** из текущего каталога, где расположены файлы **000.BMP-256.BMP** и **CNC32.PAL** для конкретного проекта СП.

ВНИМАНИЕ !

1. ПРИ ПЕРЕХОДЕ К ПРОЕКТУ КОНКРЕТНОГО СТАНКА НЕ ЗАБЫВАЙТЕ ЗАМЕНЯТЬ ФАЙЛ **CNC32.PAL** В ТЕКУЩЕМ ИЛИ В КОРНЕВОМ КАТАЛОГЕ ДИСКА **C:** НА ФАЙЛ **CNC32.PAL**, СООТВЕТСТВУЮЩИЙ КОНКРЕТНОМУ ПРОЕКТУ.
 2. ПРИ СОЗДАНИИ ФАЙЛА **CNC32.PAL** ФАЙЛ **CNC32.PAL** ПЕРЕД ПЕРВЫМ ЗАПУСКОМ ПРОГРАММЫ **USERBMP32.EXE** ВСЕГДА ДОЛЖЕН БЫТЬ ИДЕНТИЧЕН ФАЙЛУ, КОТОРЫЙ РАСПОЛОЖЕН В УЧПУ В КАТАЛОГЕ **C:\CNC32\MP0**. ЧТОБЫ ИЗБЕЖАТЬ ПОТЕРИ ЦВЕТА В ПАЛИТРЕ КОНКРЕТНОГО ПРОЕКТА, НЕ РЕКОМЕНДУЕТСЯ ИСПОЛЬЗОВАТЬ ФАЙЛ **CNC32.PAL** ОТ ДРУГОГО СТАНКА.
 3. ЕСЛИ ПРИ СОЗДАНИИ ФАЙЛА **CNC32.PAL** ФАЙЛ **CNC32.PAL** НЕ БУДЕТ НАЙДЕН ПРОГРАММОЙ **USERBMP32.EXE** В ТЕКУЩЕМ КАТАЛОГЕ, БУДЕТ СФОРМИРОВАНО СООБЩЕНИЕ: «**CAN'T OPEN CNC32.PAL**» - И БУДЕТ ВЫПОЛНЕНА ПОПЫТКА ОТКРЫТЬ ФАЙЛ **CNC32.PAL** В КОРНЕВОМ КАТАЛОГЕ ДИСКА **C:**.
 4. ЕСЛИ ФАЙЛ **CNC32.PAL** БУДЕТ НАЙДЕН В КОРНЕВОМ КАТАЛОГЕ ДИСКА **C:**, ЭТОТ ФАЙЛ БУДЕТ СЧИТАТЬСЯ АКТИВНОЙ ПАЛИТРОЙ, И ФАЙЛ **CNC32.PAL** БУДЕТ СОЗДАН.
 5. ЕСЛИ ФАЙЛ **CNC32.PAL** НЕ БУДЕТ НАЙДЕН И В КОРНЕВОМ КАТАЛОГЕ ДИСКА **C:**, БУДЕТ СФОРМИРОВАНО СООБЩЕНИЕ: «**CAN'T OPEN CNC32.PALCAN'T OPEN CNC32.PAL**» - И ФАЙЛ **CNC32.PAL** НЕ БУДЕТ СОЗДАН.
- 4) результат работы программы **USERBMP32.EXE**:
- создание файла **CNC32.PAL**, который представляет собой библиотеку иконок конкретного станка;
 - возможное изменение файла **CNC32.PAL**.
- 5) подключить файлы **CNC32.PAL** и **CNC32.PAL** к Про, для этого скопировать их в тот каталог УЧПУ, где расположен файл **CNC32.PAL** (обычно **C:\CNC32\MP0**).

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

Вкл./выкл.	- включено/выключено
Вх./вых.	- вход/выход
ОЗУ	- оперативное запоминающее устройство (RAM)
ПК	- персональный компьютер
ПЛ	- программа логики станка
ПО	- пульт оператора
Про	- программное обеспечение
СП	- станочный пульт
УП	- управляющая программа
УЧПУ	- устройство числового программного управления
ЦАП	- цифро-аналоговый преобразователь
VCD	- двоично-десятичный код
PLC	- программируемый логический контроллер

ПЕРЕЧЕНЬ РИСУНКОВ

Рисунок 2.1	- Архитектура Про управления процессом
Рисунок 2.2	- Линии связи между модулями управления логикой
Рисунок 5.1	- Схема таймера
Рисунок 5.2	- Диаграмма работы таймера
Рисунок 5.3	- Схема счётчика
Рисунок 5.4	- Диаграмма работы одновибратора
Рисунок 5.5	- Диаграмма работы одновибратора (пример)
Рисунок 5.6	- Применение скобок в логических схемах. Пример 1
Рисунок 5.7	- Применение скобок в логических схемах. Пример 2
Рисунок 9.1	- Топология изображения страниц «Среды» PLC
Рисунок 9.2	- Алгоритм выполнения ПЛ
Рисунок 24.1	- Диаграмма изменения сигналов CYCLE и STABY
Рисунок 24.2	- Диаграмма сигналов при установке сигнала АСТОЛ во время смены инструмента
Рисунок 25.1	- Нумерация кнопок и лампочек СП NC110-7
Рисунок 25.2	- Нумерация кнопок и лампочек СП NC310-7

ПЕРЕЧЕНЬ ТАБЛИЦ

Таблица 9.1	- Перечень директив, используемых в PLC
Таблица 10.1	- Ошибки использования языка PLC
Таблица 10.2	- Ошибки программы, выявляемые при компиляции
Таблица 10.3	- Ошибки компиляции
Таблица 10.4	- Ошибки отладчика
Таблица 24.1	- Базовые сигналы пакета «К» интерфейса PLC
Таблица 24.2	- Базовые сигналы пакета «N» интерфейса PLC
Таблица 25.1	- Сигналы пакета «К» интерфейса PLC для СП NC110-7
Таблица 25.2	- Сигналы пакета «N» для СП NC-200/210/220/230/310
Таблица 25.3	- Сигналы пакета «N» для СП NC-201/201M/202
Таблица 25.4	- Сигналы пакета «N» для станочного меню и вертикальных иконок видеокадра #7 <u>NC-200/201/201M/202/210/220/230/310</u>
Таблица 25.5	- Сигналы пакета «К» интерфейса PLC для СП NC310-7

ПЕРЕЧЕНЬ ПРИЛОЖЕНИЙ

Приложение А	- Блок-схемы алгоритмов программного интерфейса PLC
--------------	---

ПРИЛОЖЕНИЕ А
(обязательное)

БЛОК-СХЕМЫ АЛГОРИТМОВ ИНТЕРФЕЙСА PLC

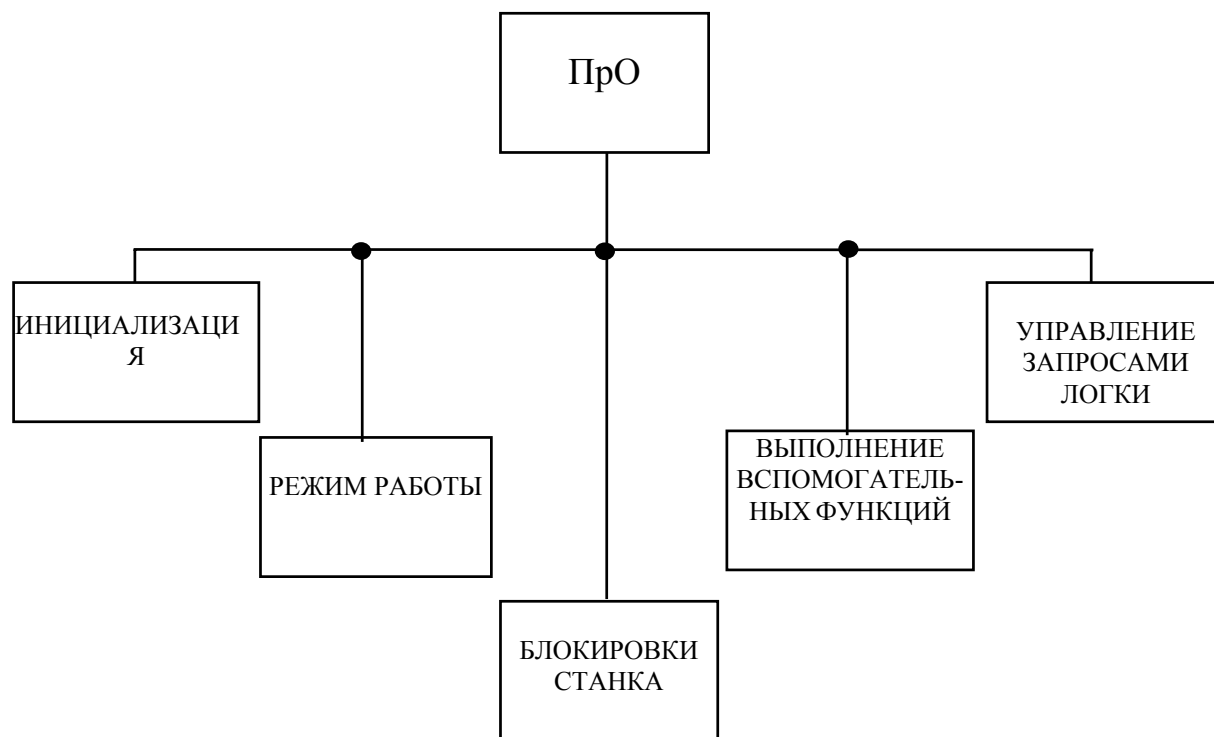


Рисунок А.1 - Управление ПрО

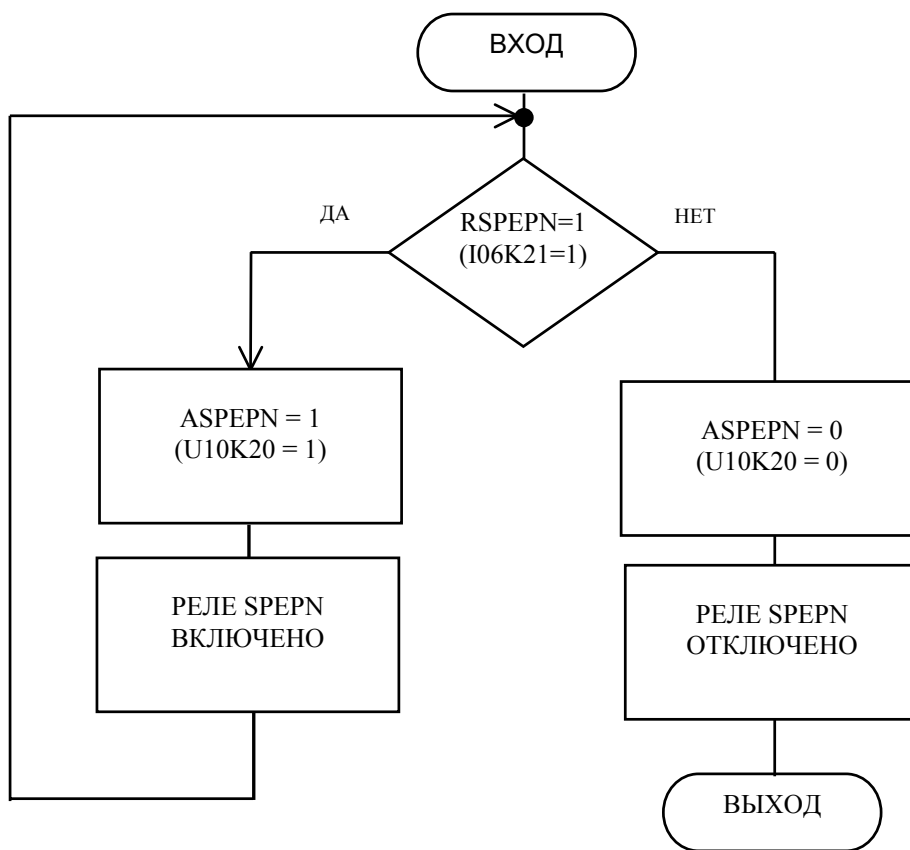


Рисунок А.2 - Процедура «ВКЛЮЧЕНИЕ РЕЛЕ SPEPN»

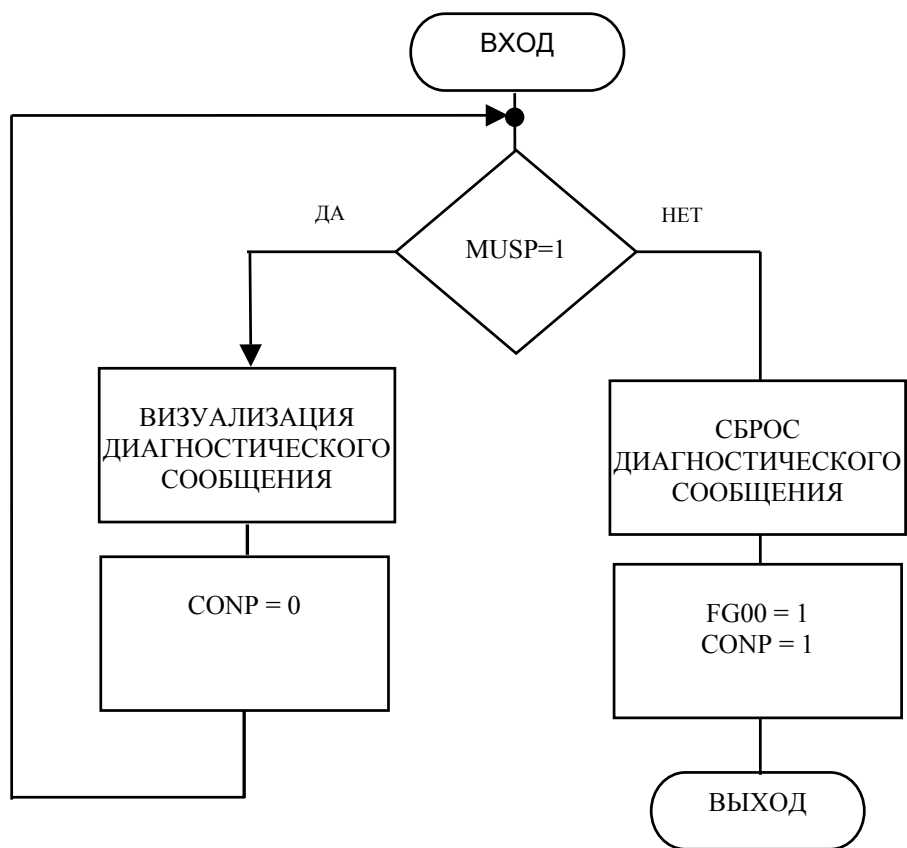


Рисунок А.3 - Процедура «ОТКРЫТИЕ»

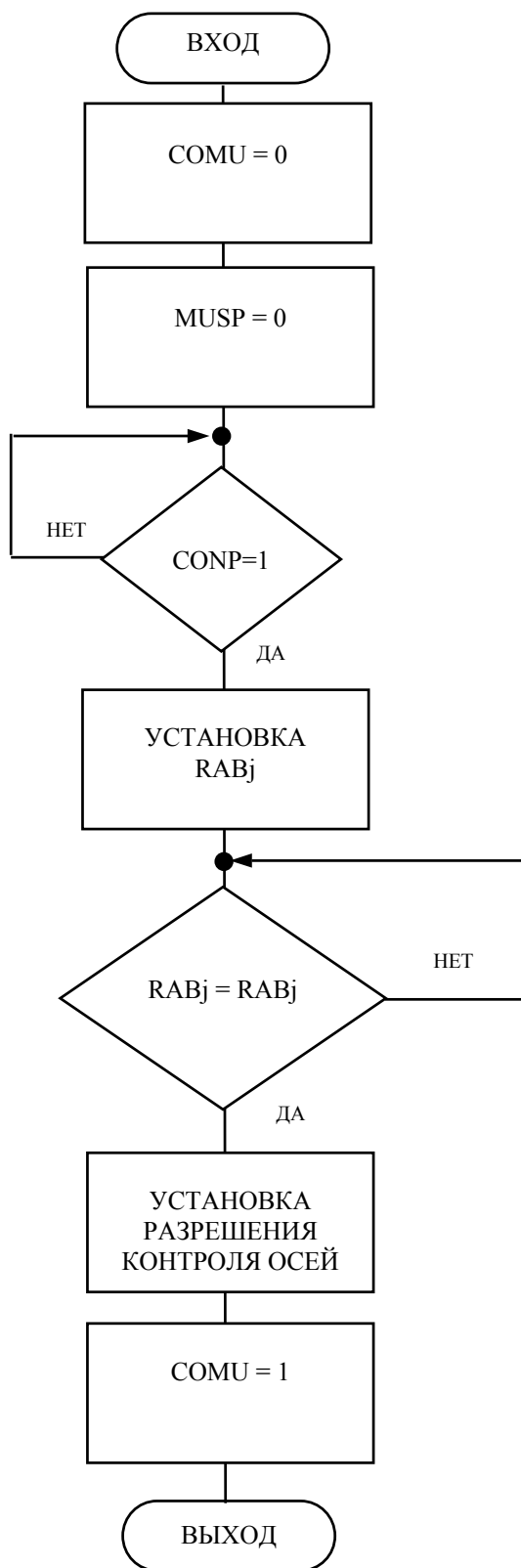


Рисунок А.4 - Разрешение осей

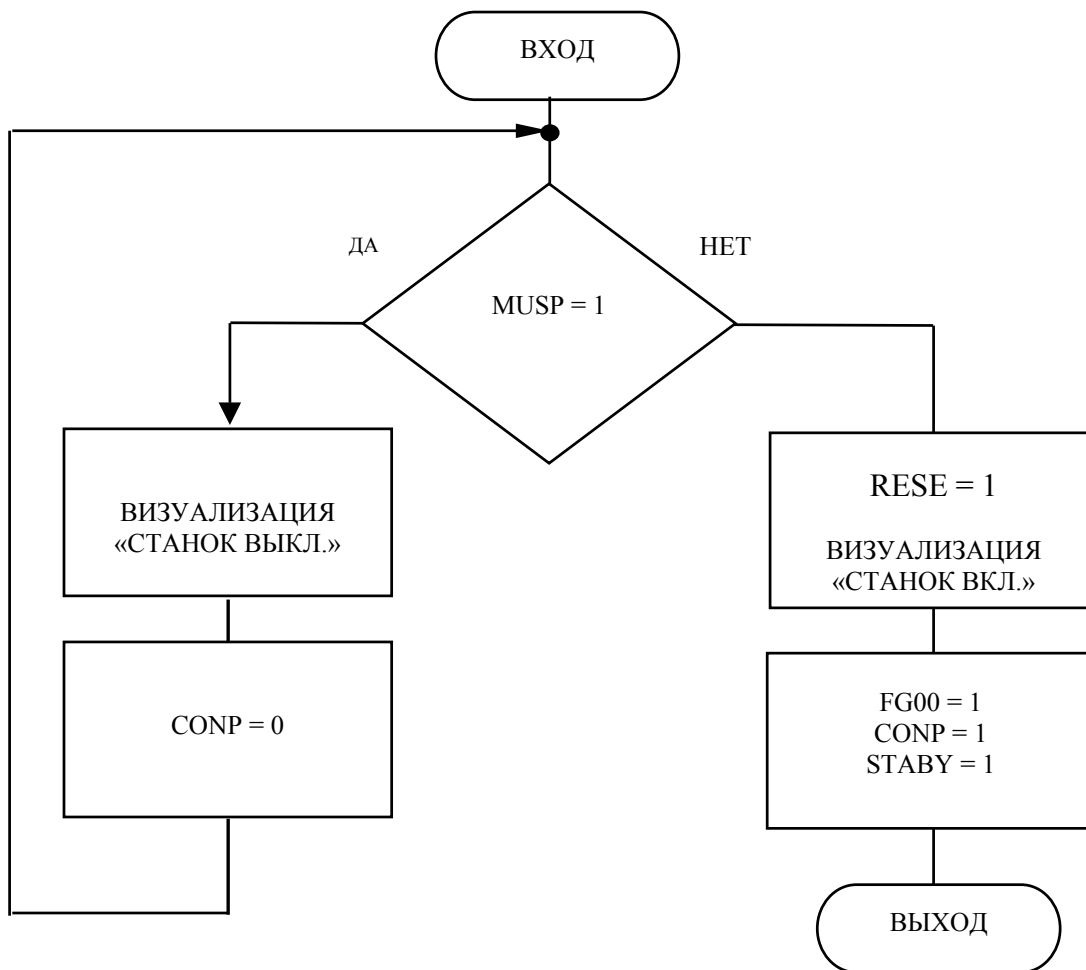


Рисунок А.5 - Включение после аварии



Рисунок А.6 - Процедура «СБРОС»

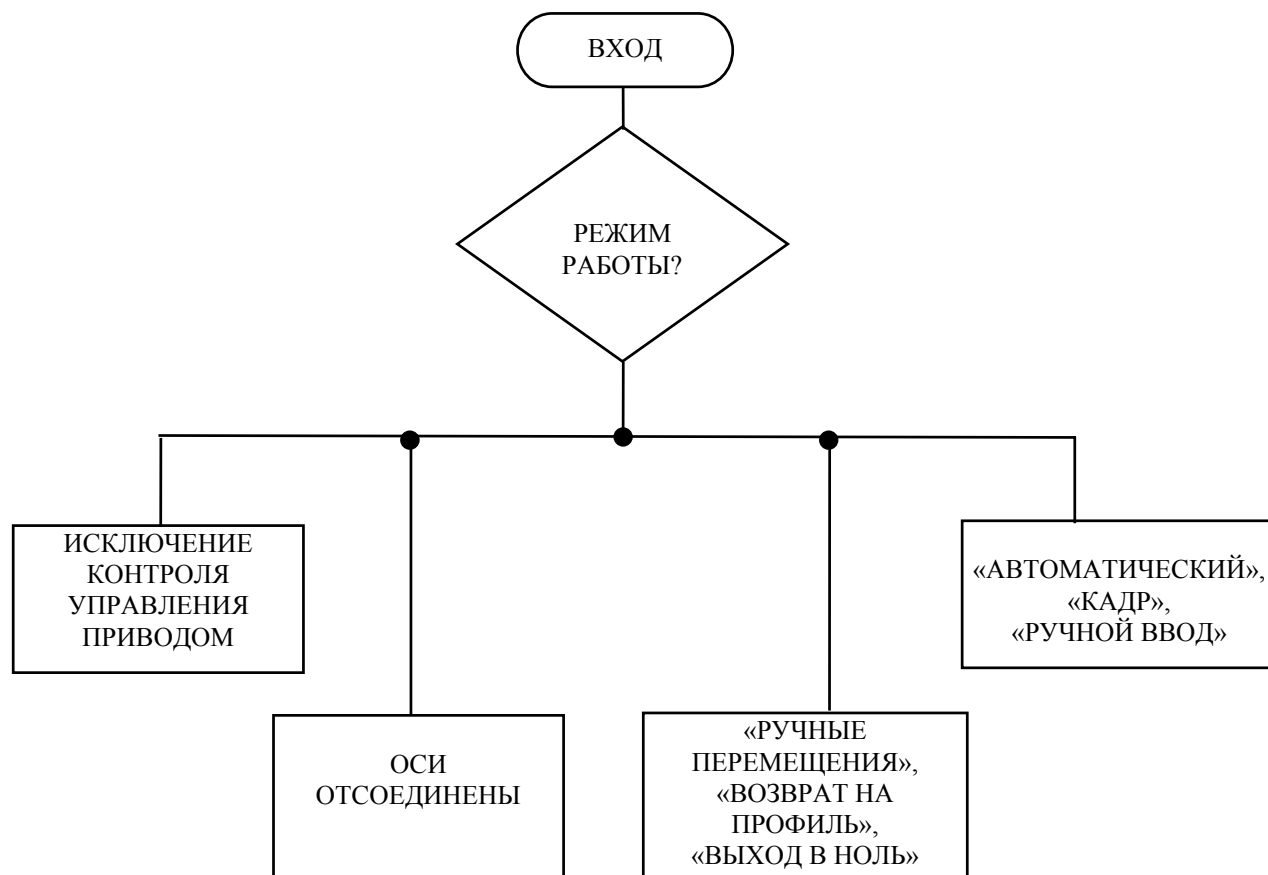


Рисунок А.7 - Циклы режимов работы

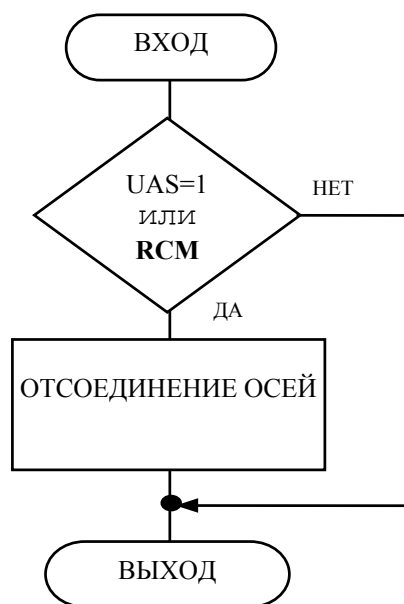


Рисунок А.8 - Цикл отсоединения осей

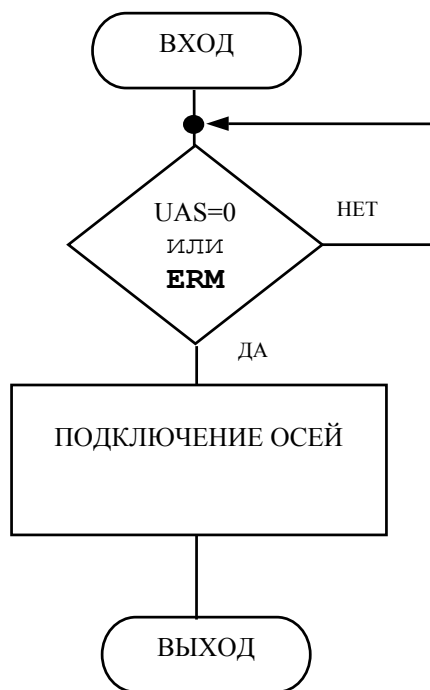


Рисунок А.9 Цикл выхода отсоединения осей

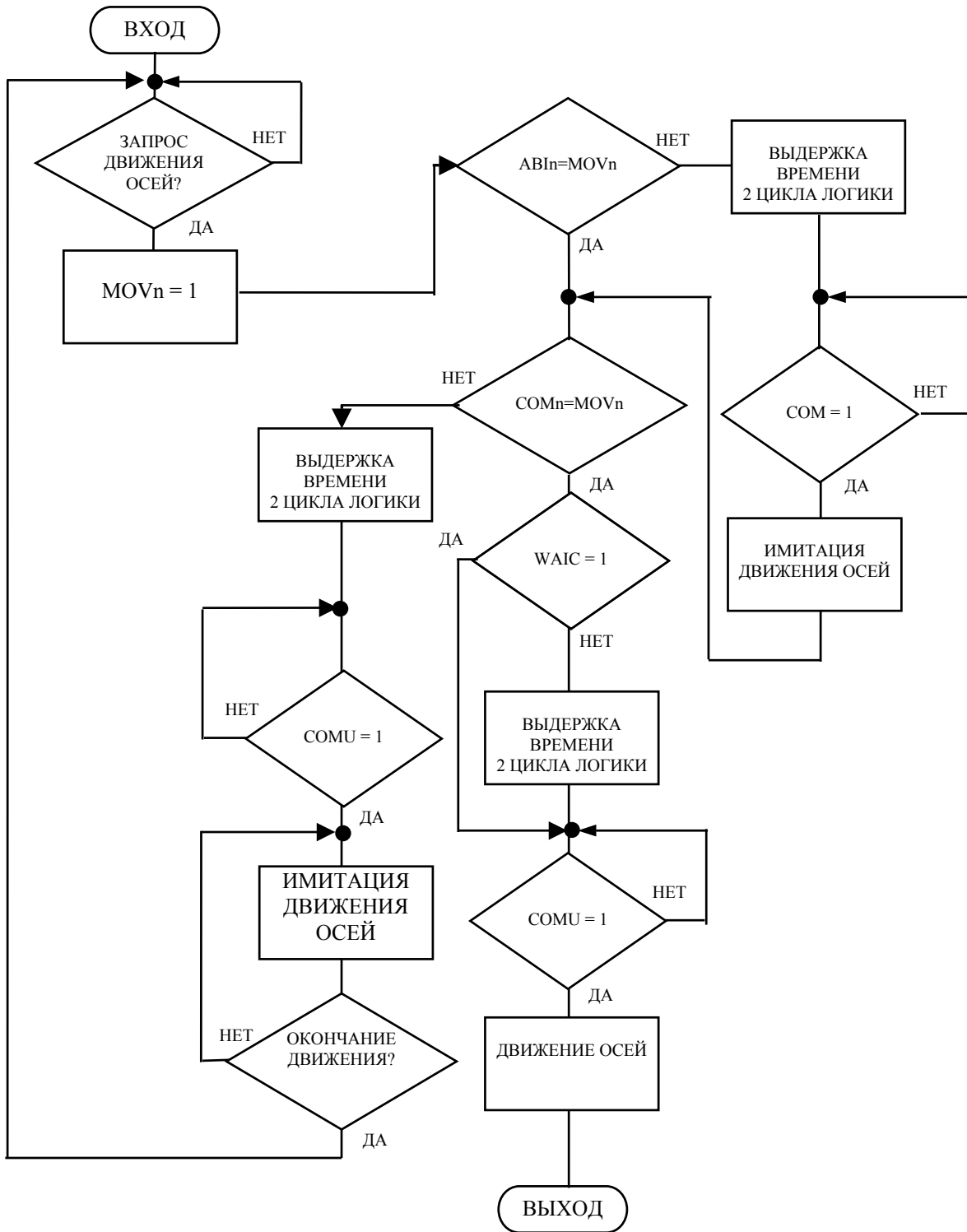


Рисунок А.10 - Переключение и разрешение движения осей

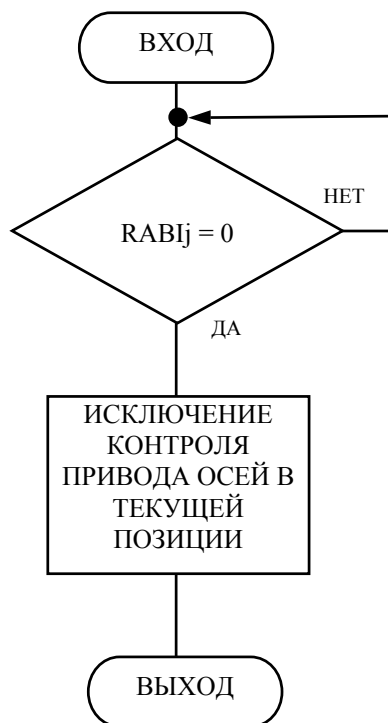


Рисунок А.11 - Исключение контроля за осями

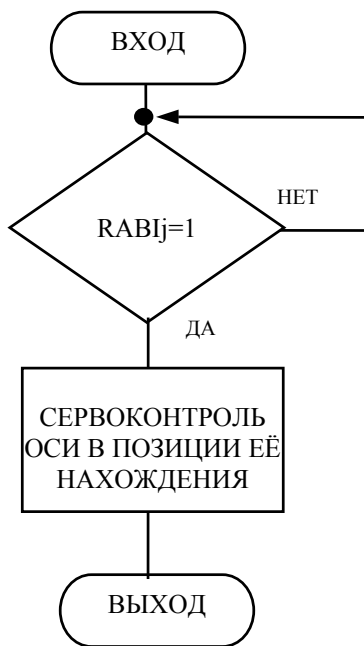
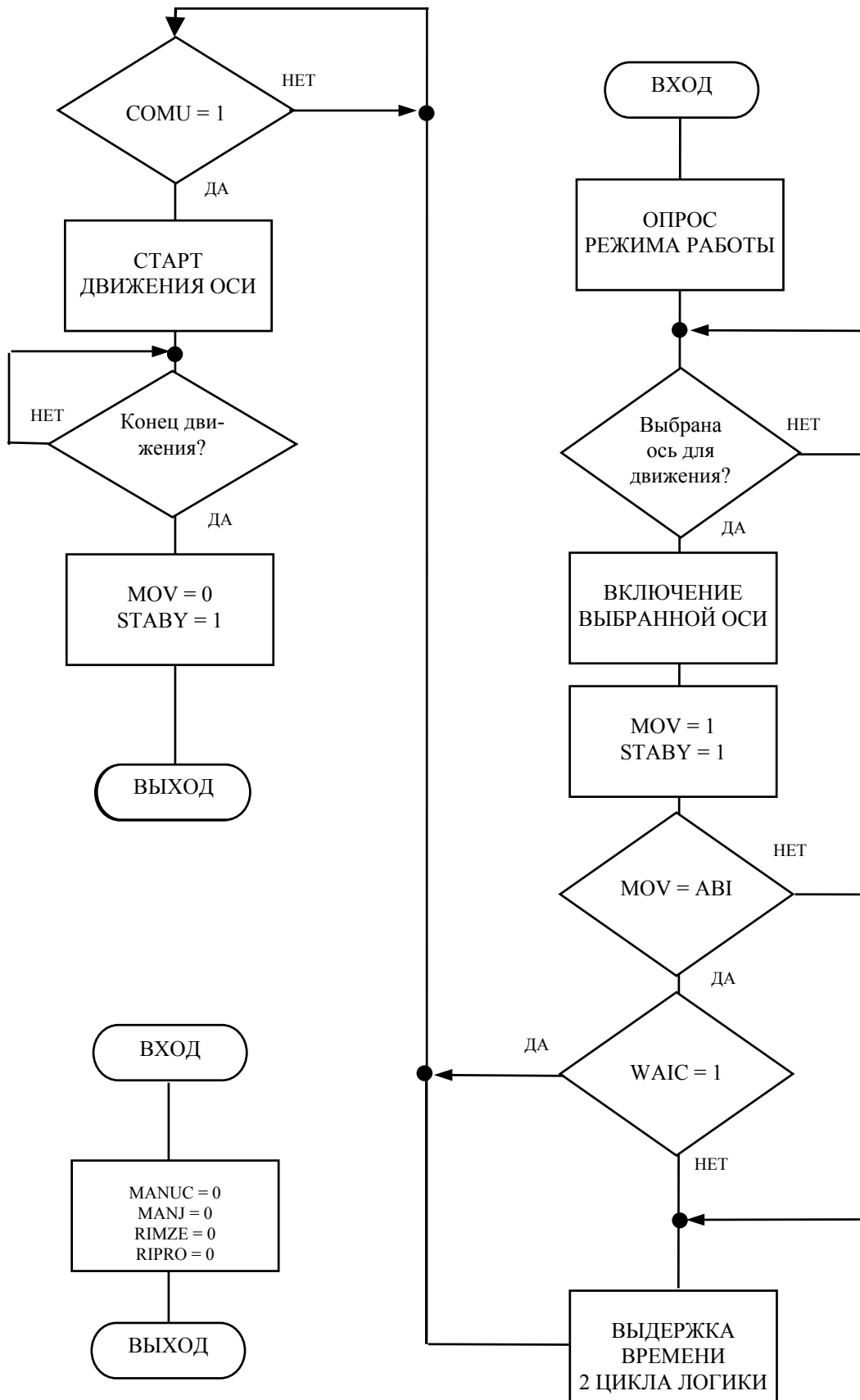


Рисунок А.12 - Включение контроля за осями



Рисунки А.13 - Ручное перемещение осей

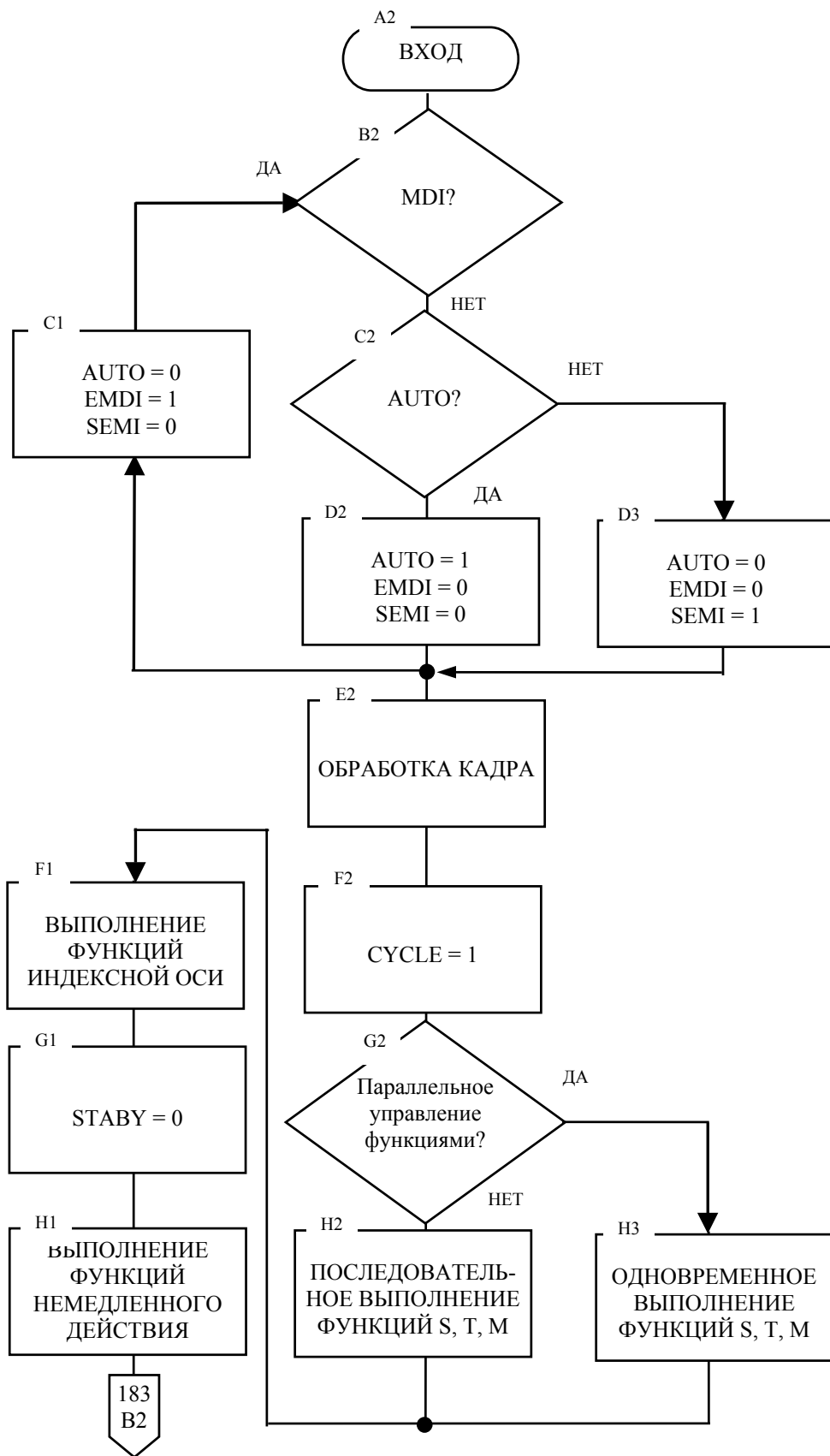


Рисунок А.14 (лист 1/3) - Управление в режимах «АВТОМАТИЧЕСКИЙ», «КАДР», «РУЧНОЙ ВВОД»

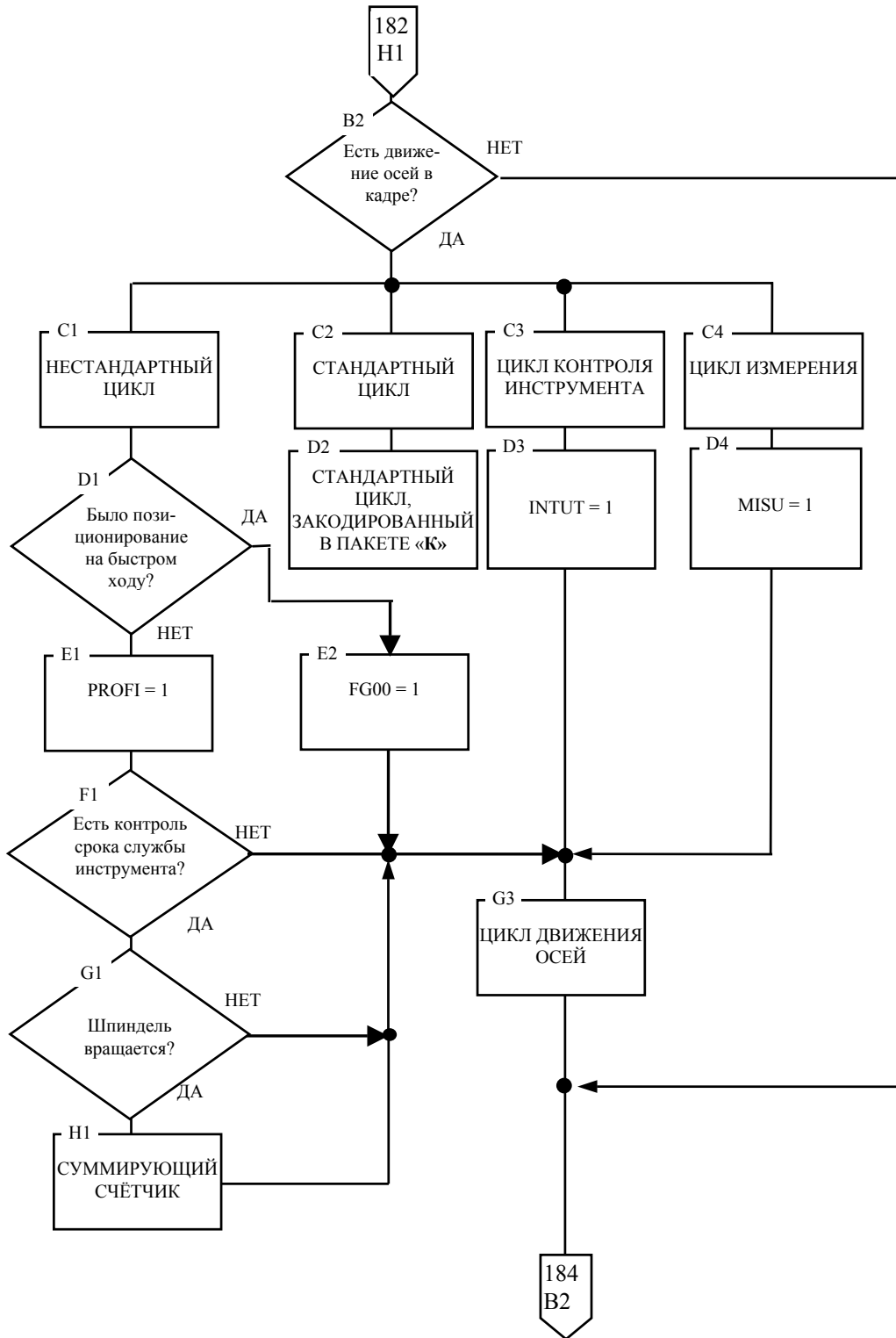


Рисунок А.14 (лист 2/3) - Управление в режимах «АВТОМАТИЧЕСКИЙ», «КАДР», «РУЧНОЙ ВВОД»

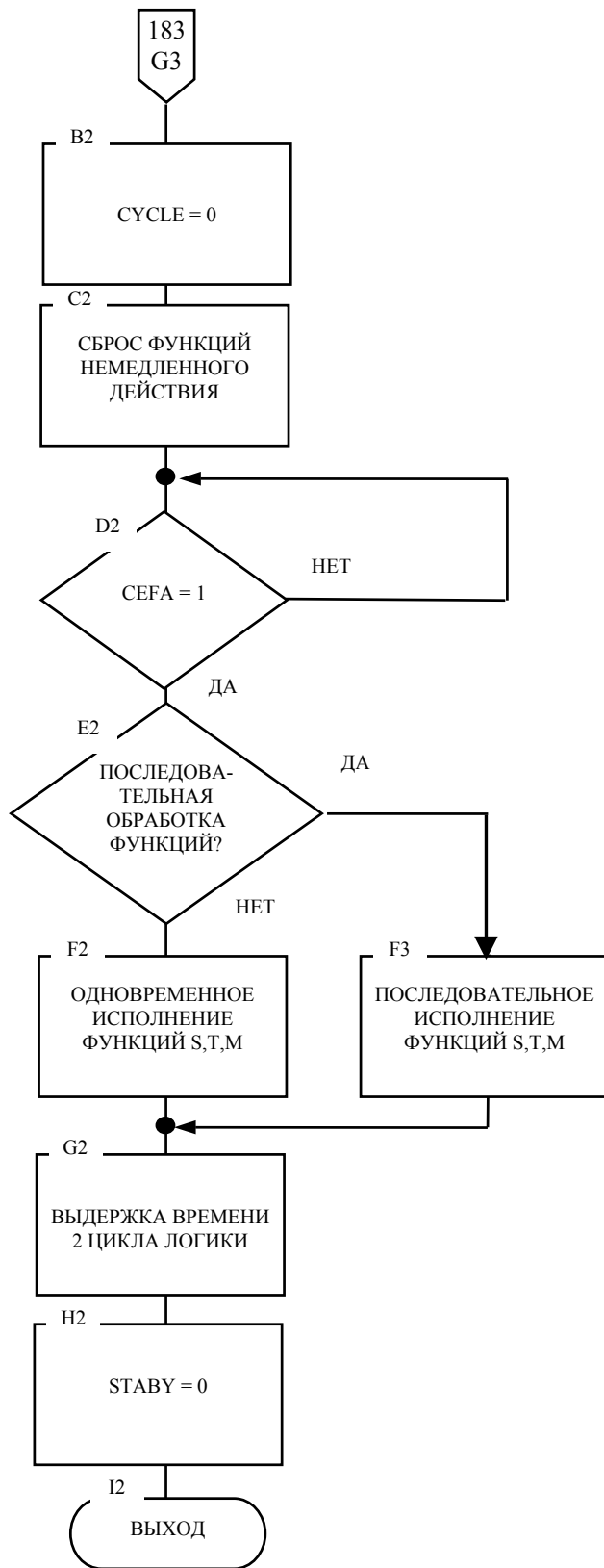


Рисунок А.14 (лист 3/3) - Управление в режимах «АВТОМАТИЧЕСКИЙ», «КАДР», «РУЧНОЙ ВВОД»

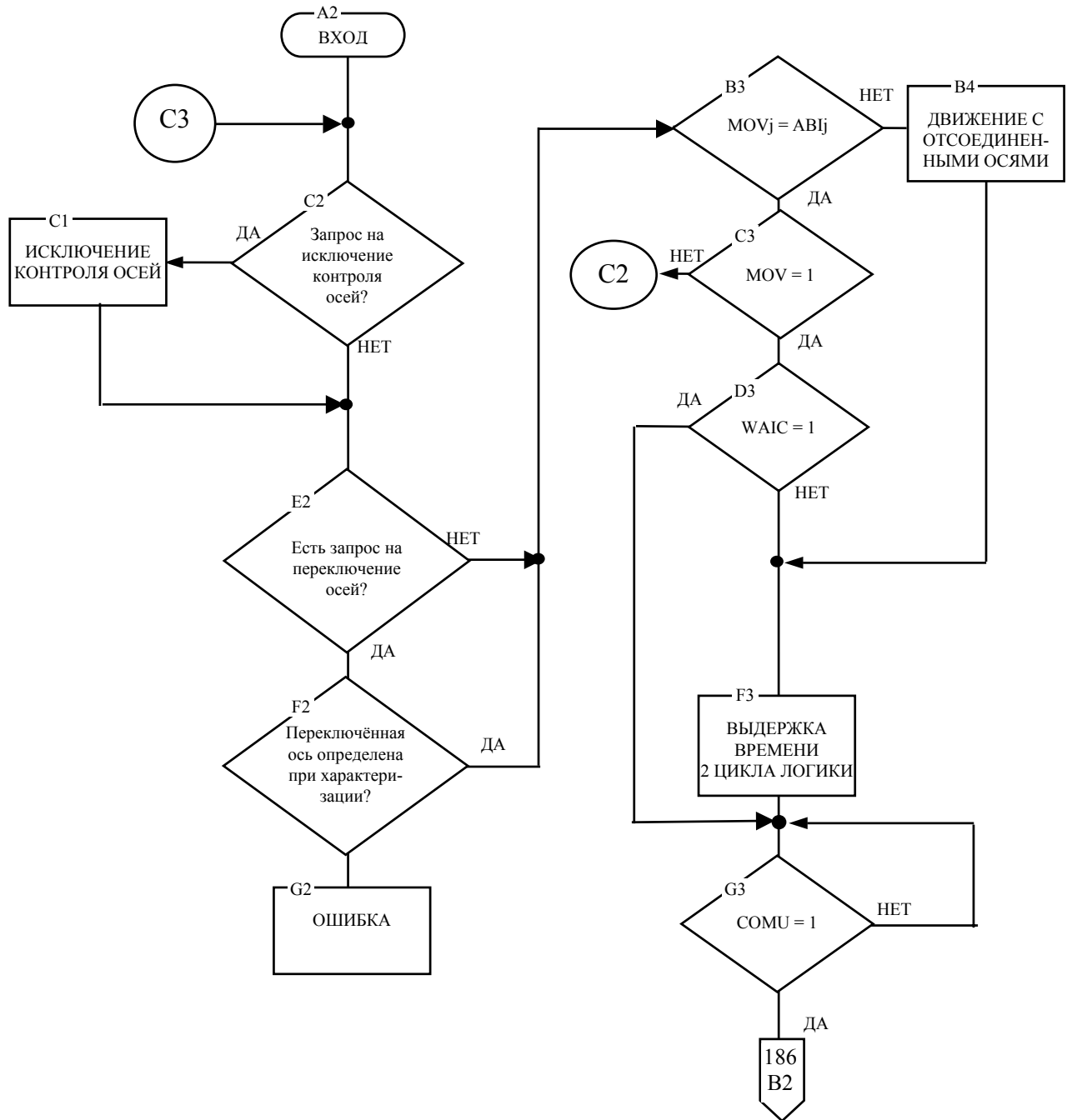


Рисунок А.15 (лист 1/2) - Движение осей

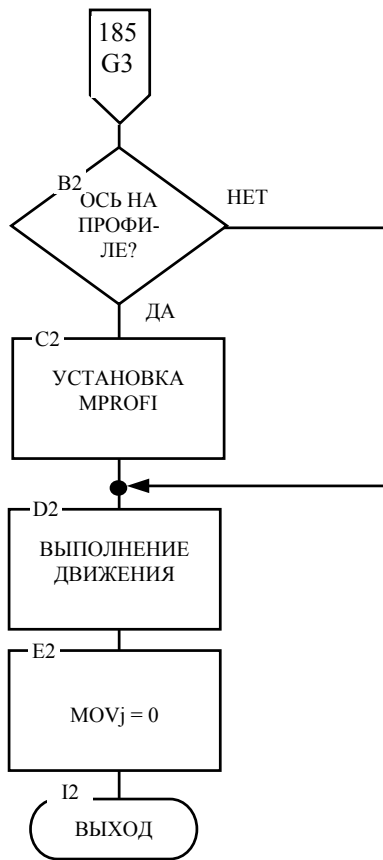


Рисунок А.15 (лист 2/2) - Движение осей

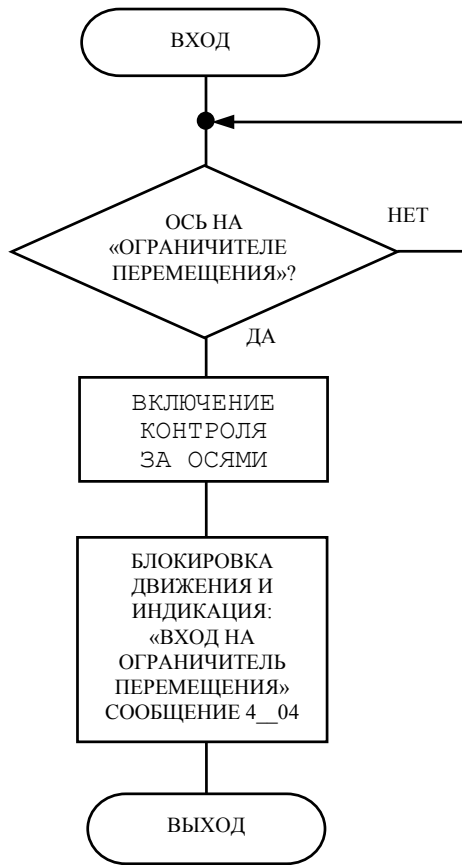


Рисунок А.16 – Цикл «ВХОД НА ОГРАНИЧЕНИЕ ПЕРЕМЕЩЕНИЯ»

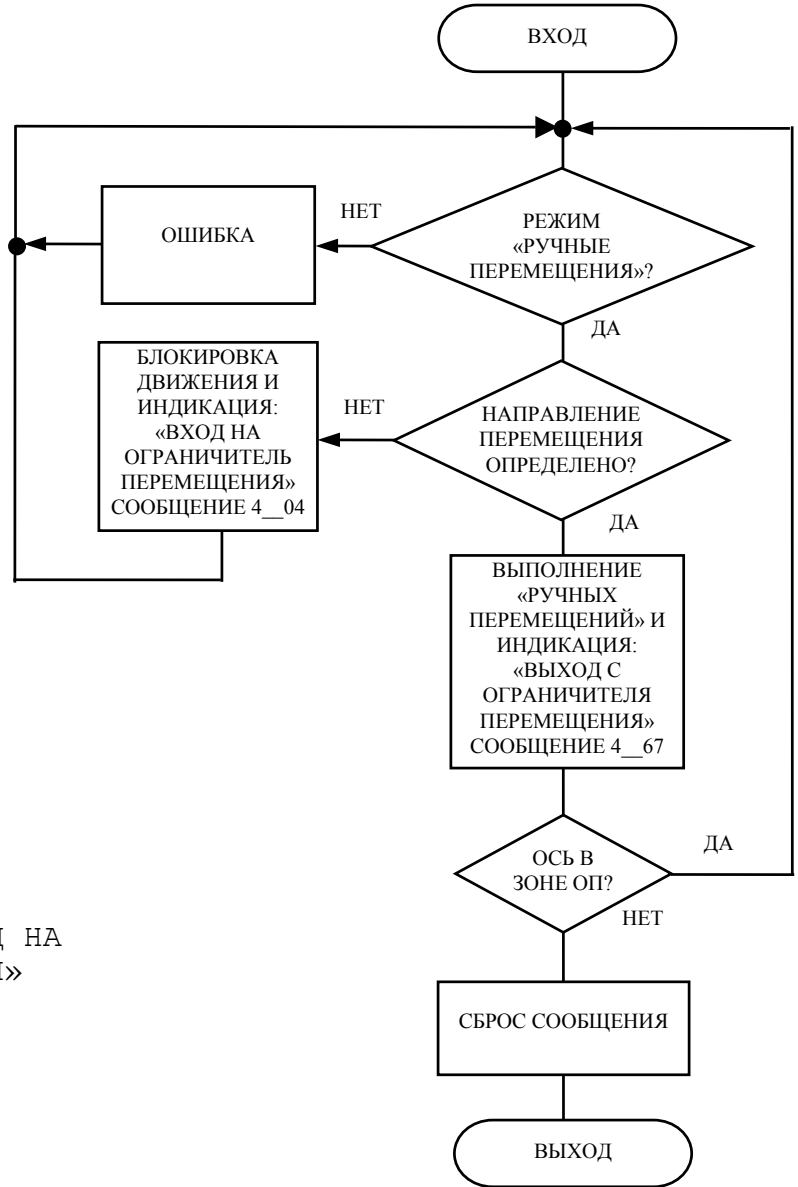


Рисунок.А.17 Цикл «ВЫХОД ИЗ ОГРАНИЧЕНИЯ ПЕРЕМЕЩЕНИЯ»

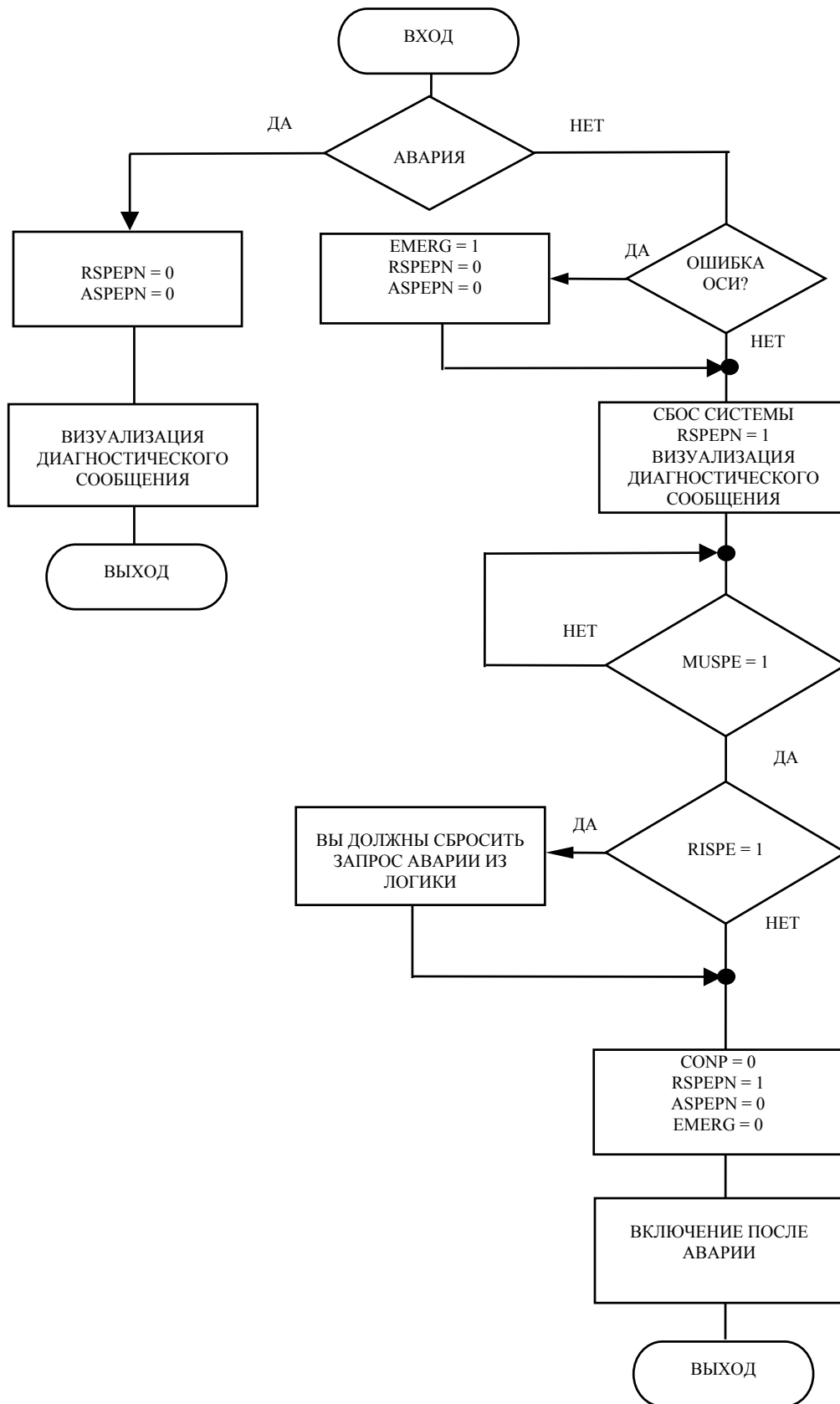


Рисунок А.18 - Цикл «АВАРИЯ»

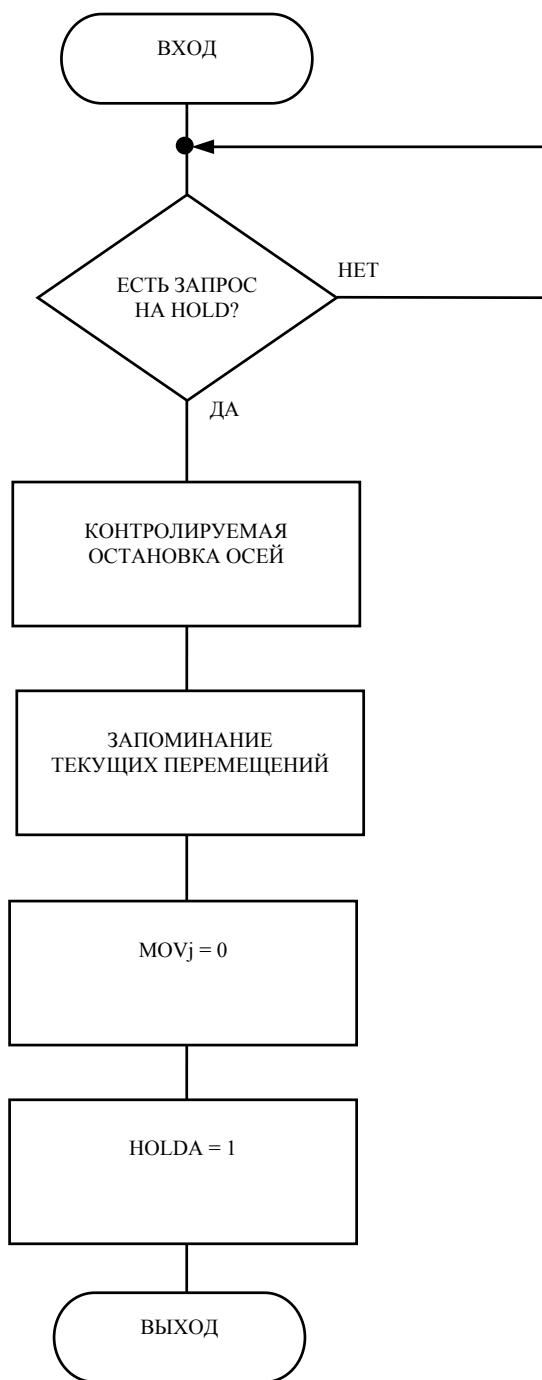


Рисунок А.19 - Цикл «ВХОД В HOLD»

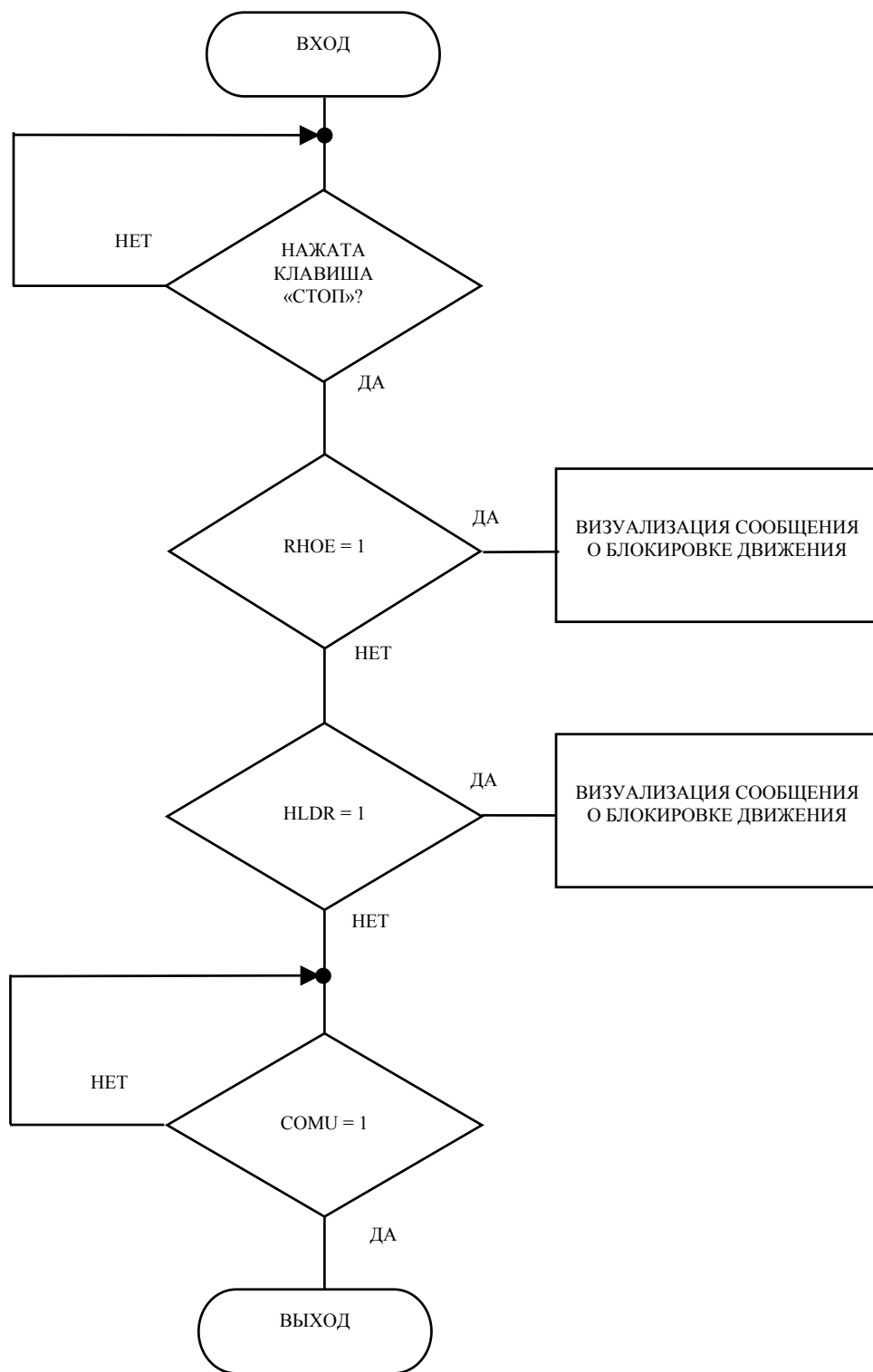


Рисунок А.20 - Цикл «РАЗРЕШЕНИЕ НА ВЫХОД ИЗ HOLD»

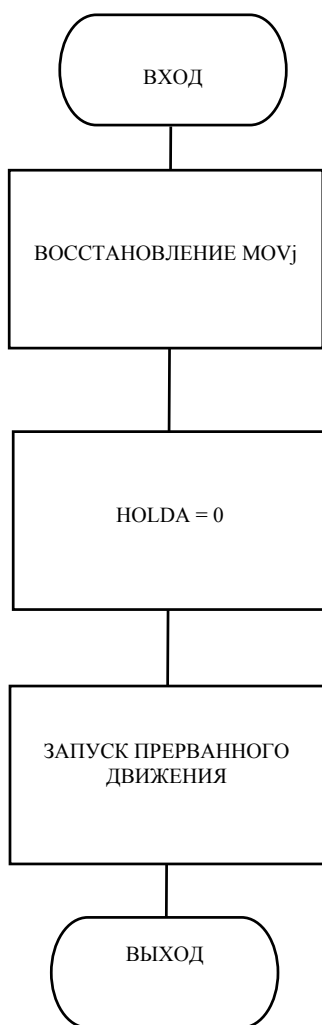


Рисунок А.21 - Цикл «ВЫХОД ИЗ HOLD»

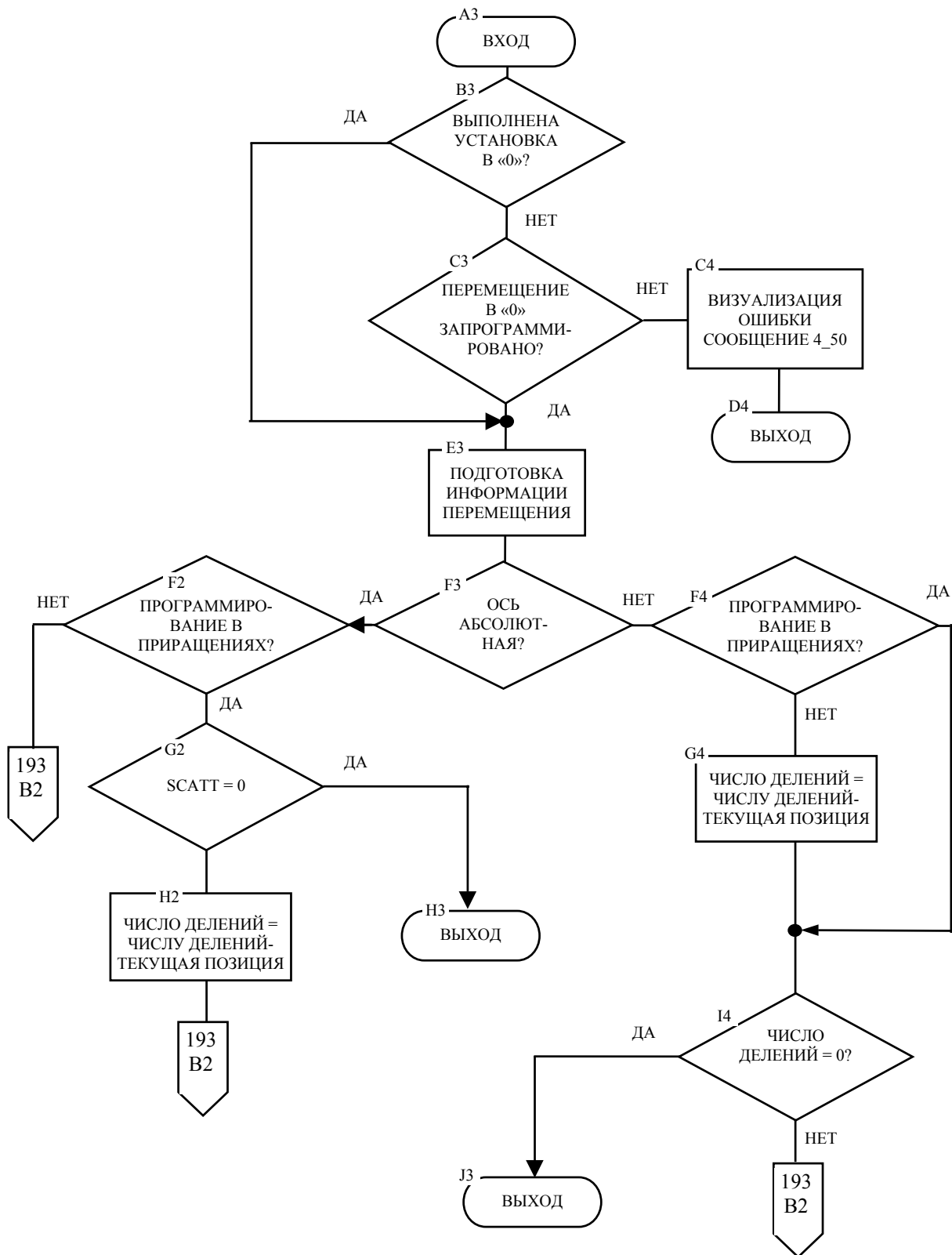


Рисунок А.22 (лист 1/2) - Обработка функций «ИНДЕКСНАЯ ОСЬ»

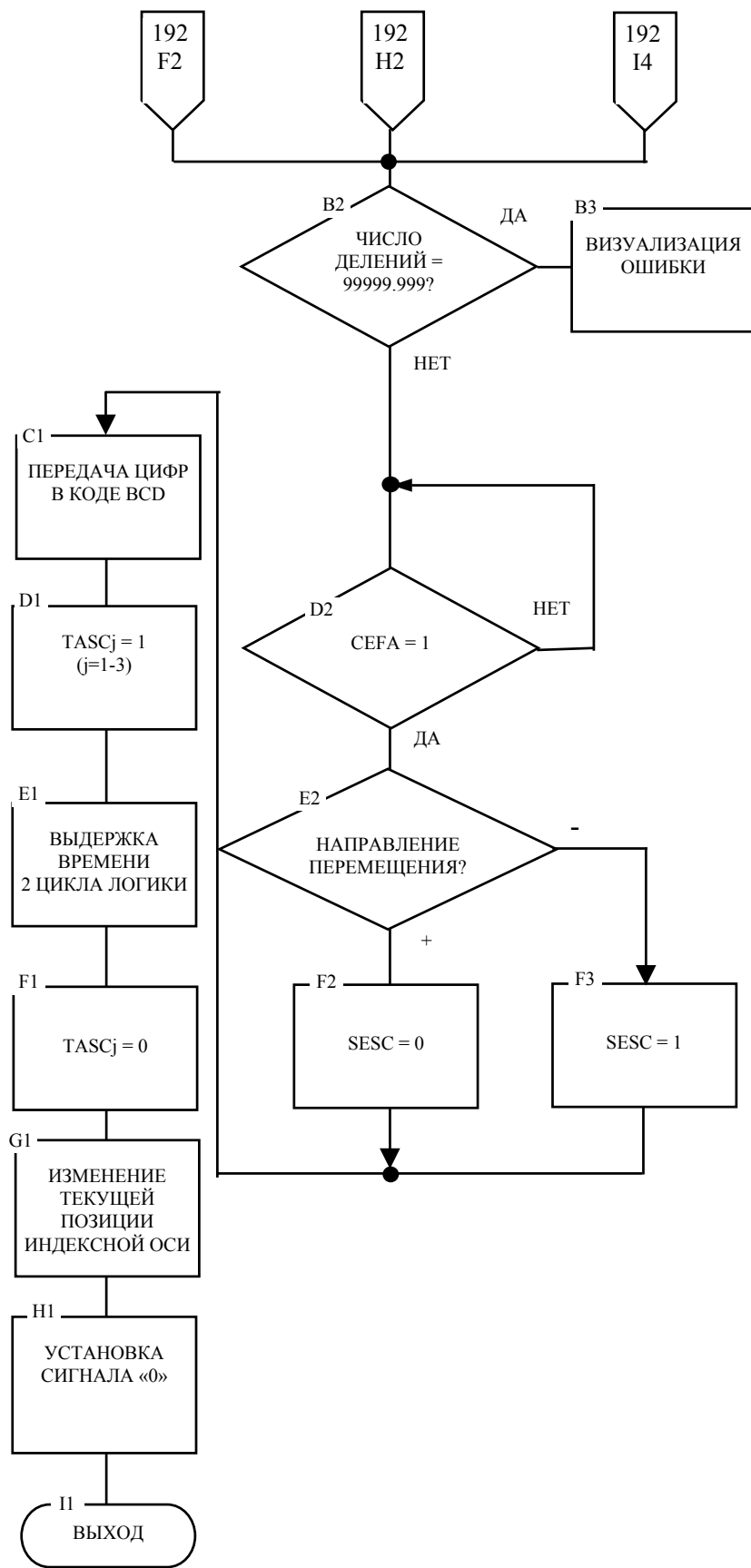


Рисунок А.22 (лист 2/2) - Обработка функций «ИНДЕКСНАЯ ОСЬ»

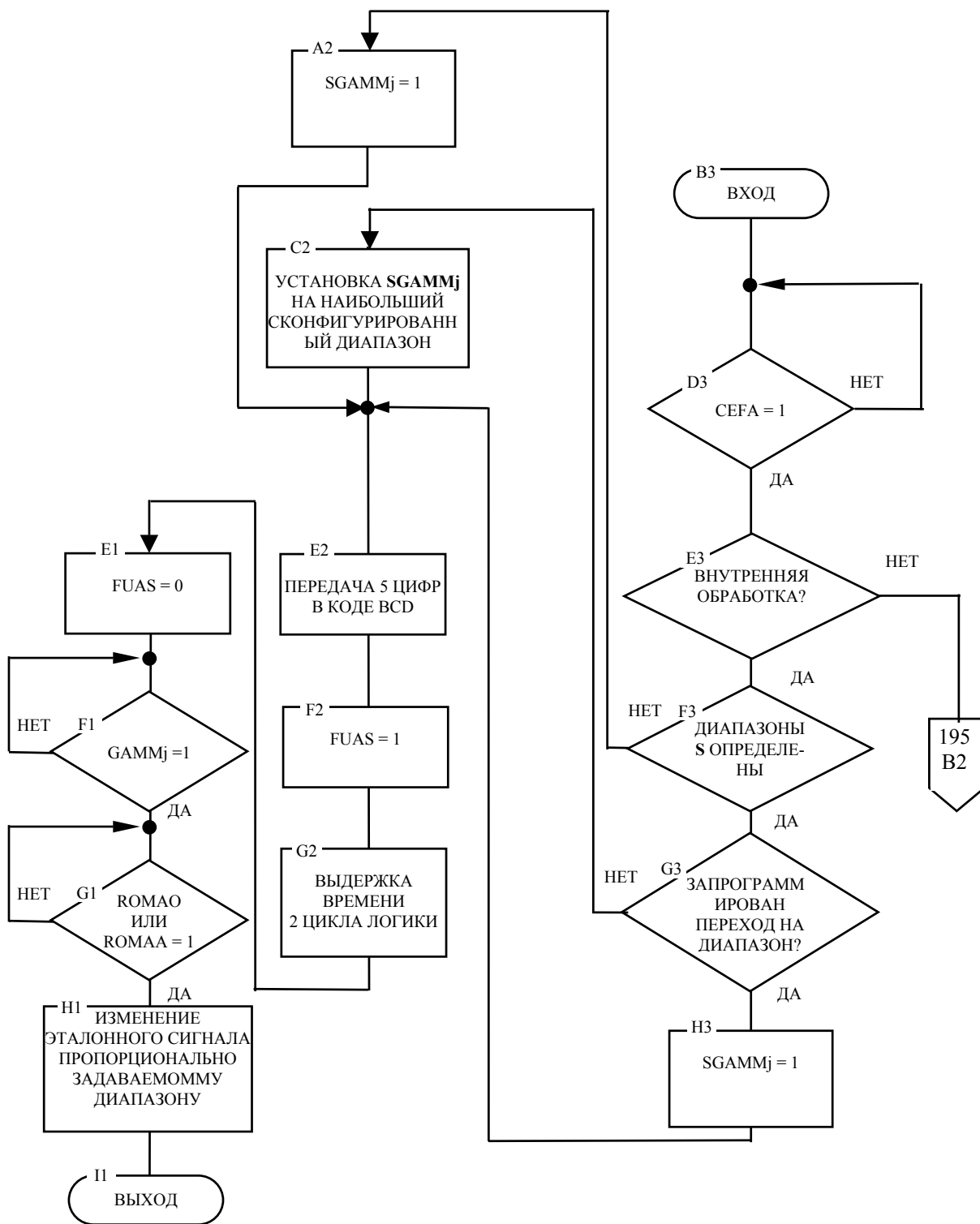


Рисунок А.23 (лист 1/2) – Цикл управления шпинделем



Рисунок А.23 (лист 2/2) - Цикл управления шпинделем

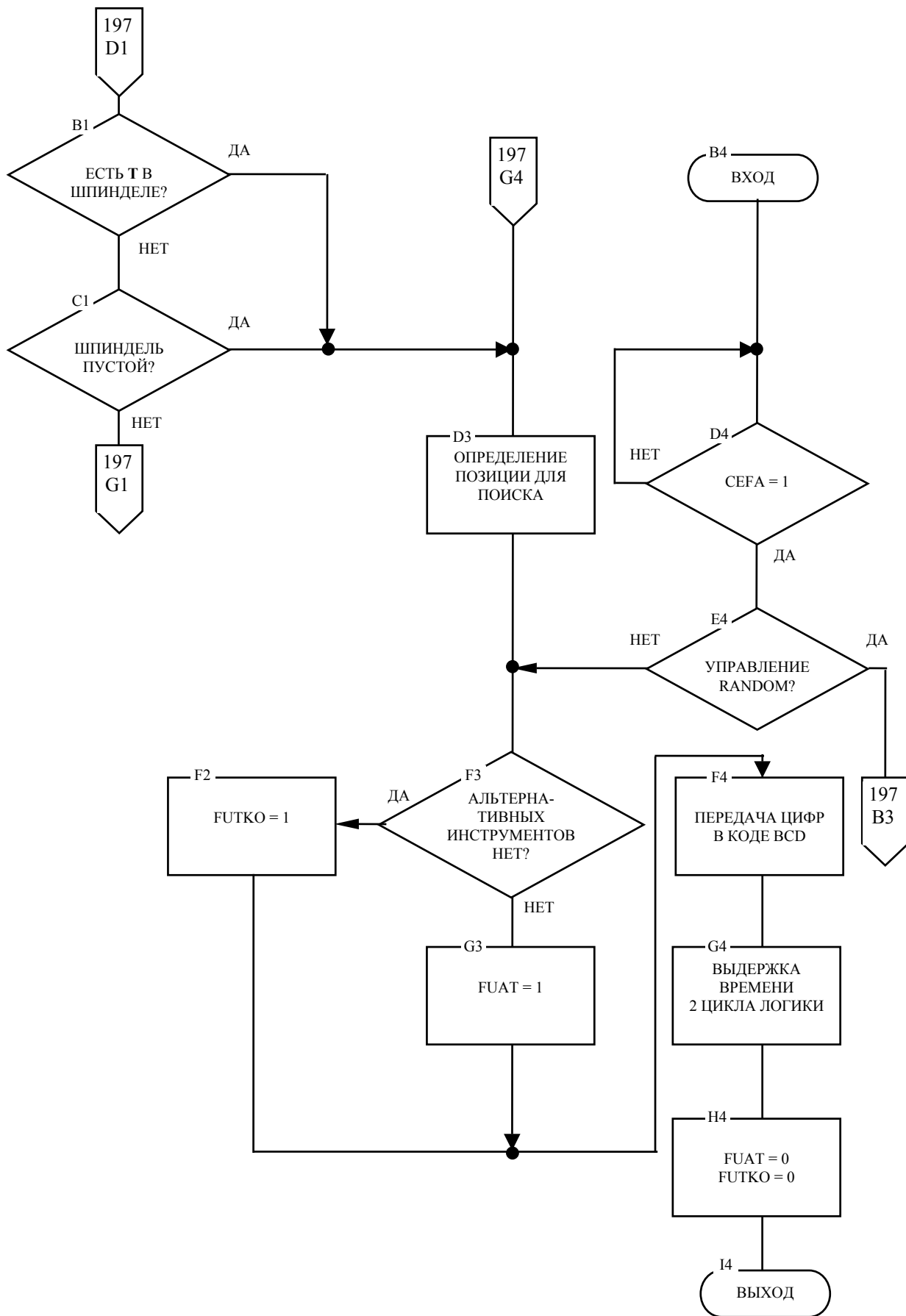


Рисунок А.24 (лист 1/2) - Обработка функций «Т»

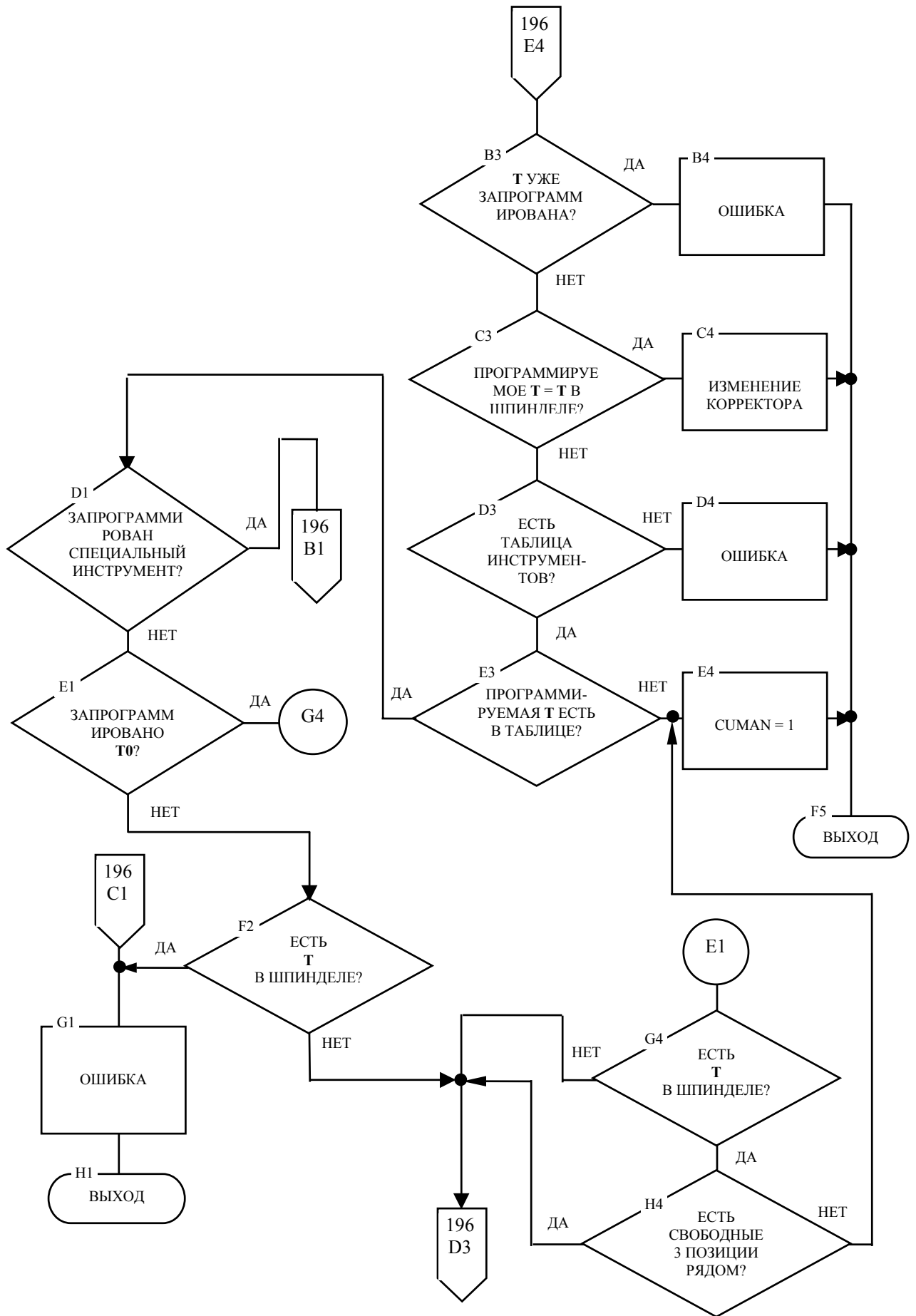


Рисунок А.24 (лист 2/2) - Обработка функций «Т»

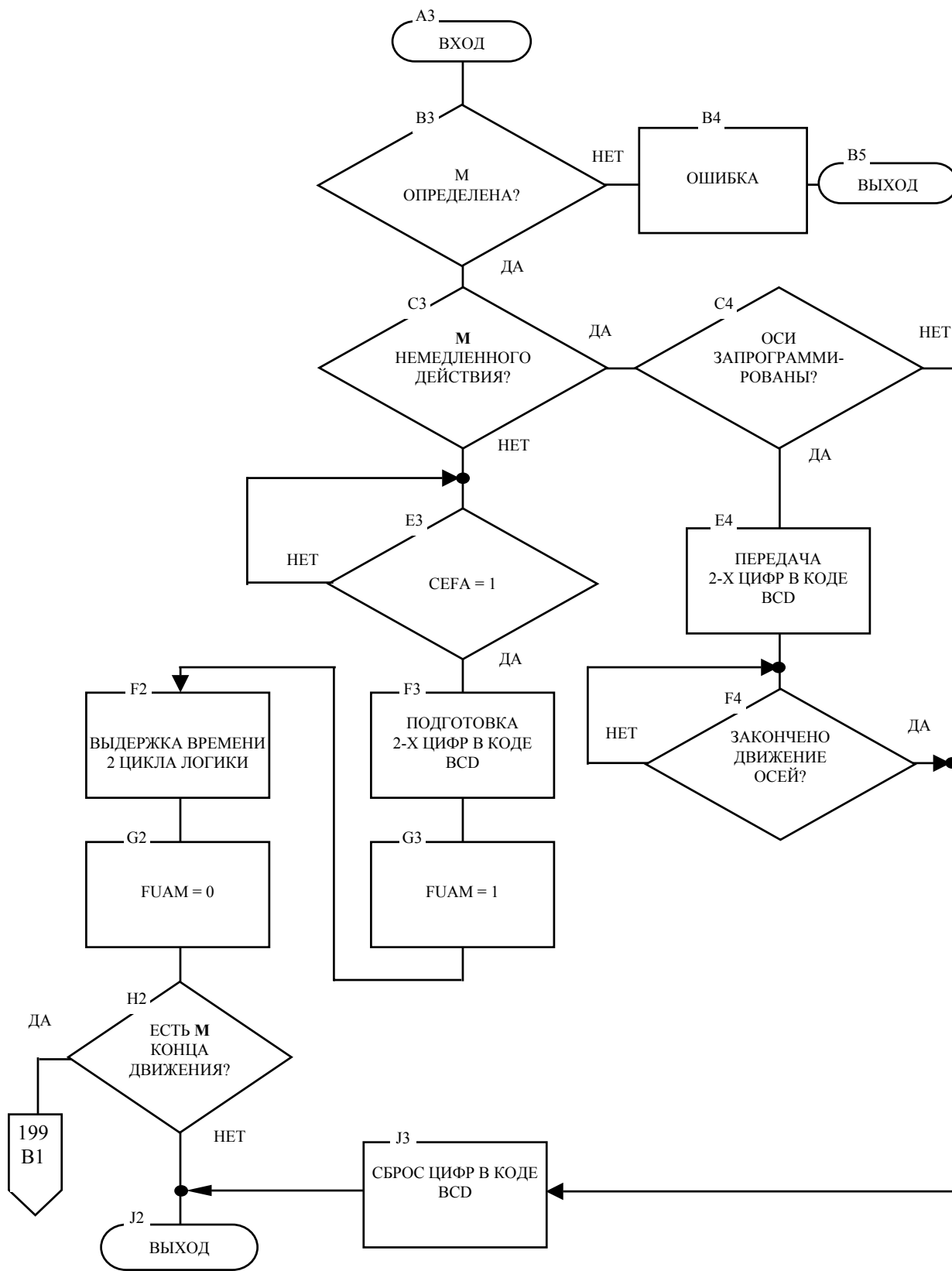


Рисунок А.25 (лист 1/2) – Обработка функций «М»

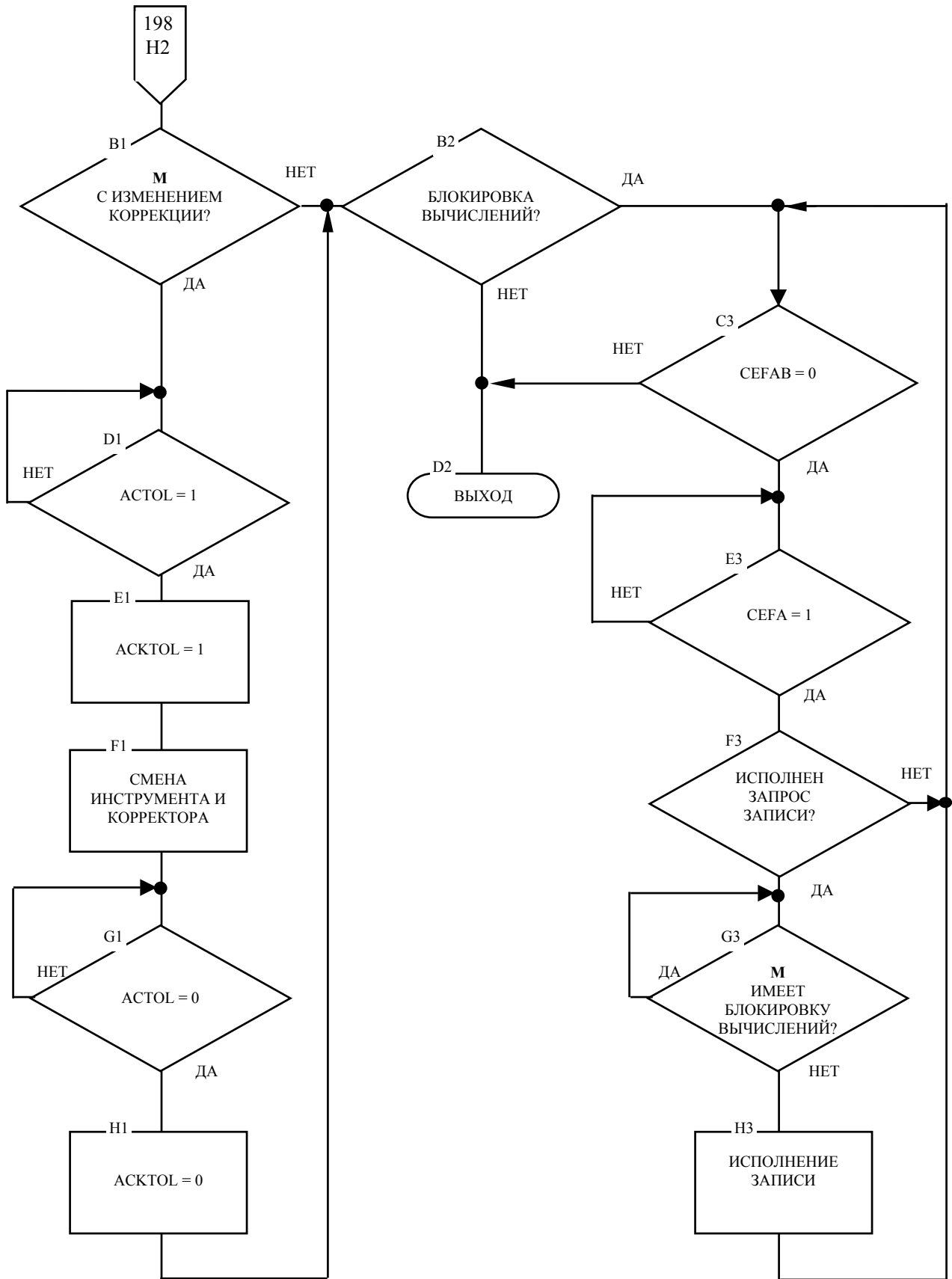


Рисунок А.25 (лист 2/2) - Обработка функций «М»

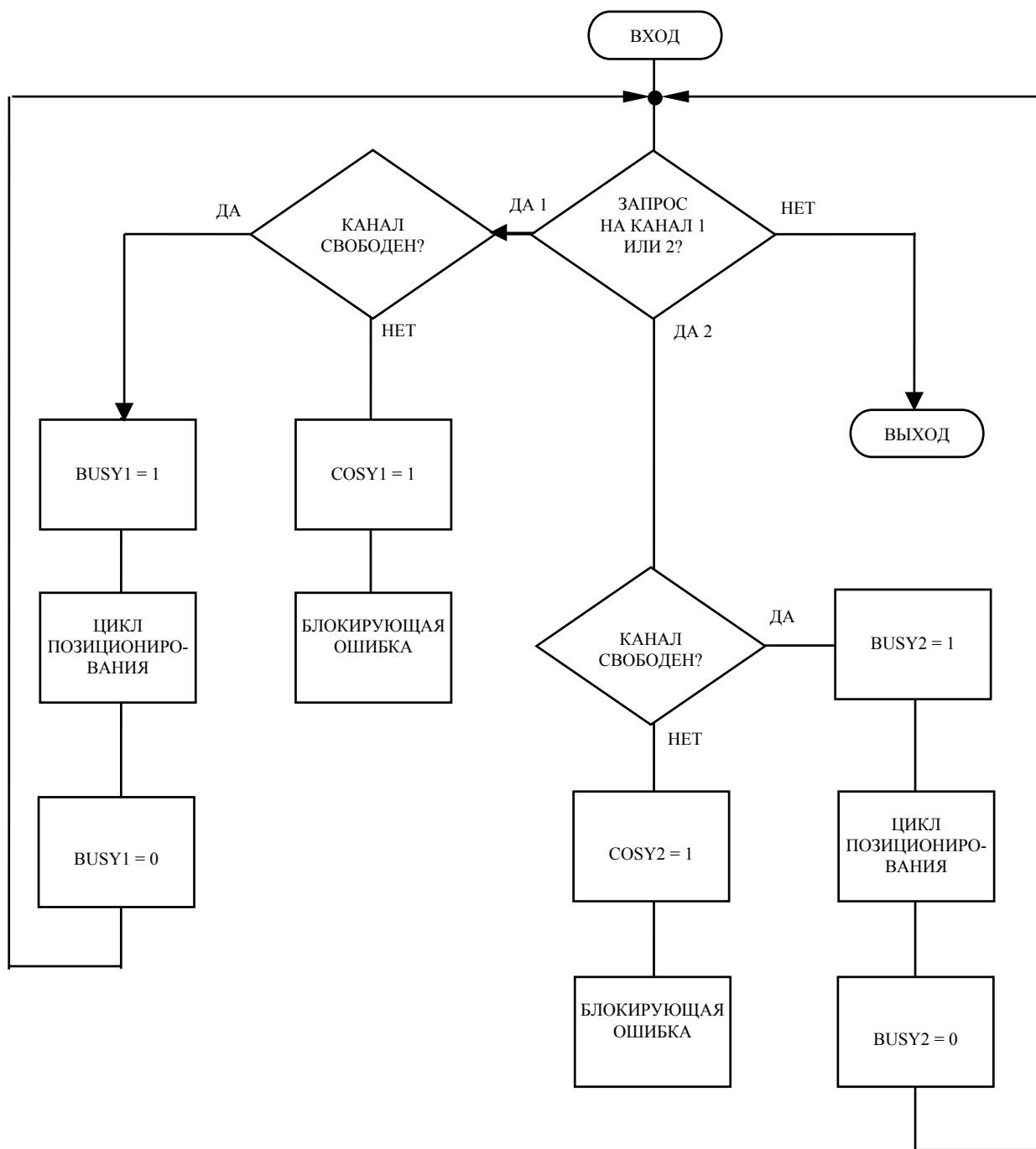


Рисунок А.26 - Цикл «Перемещение осей от точки к точке»

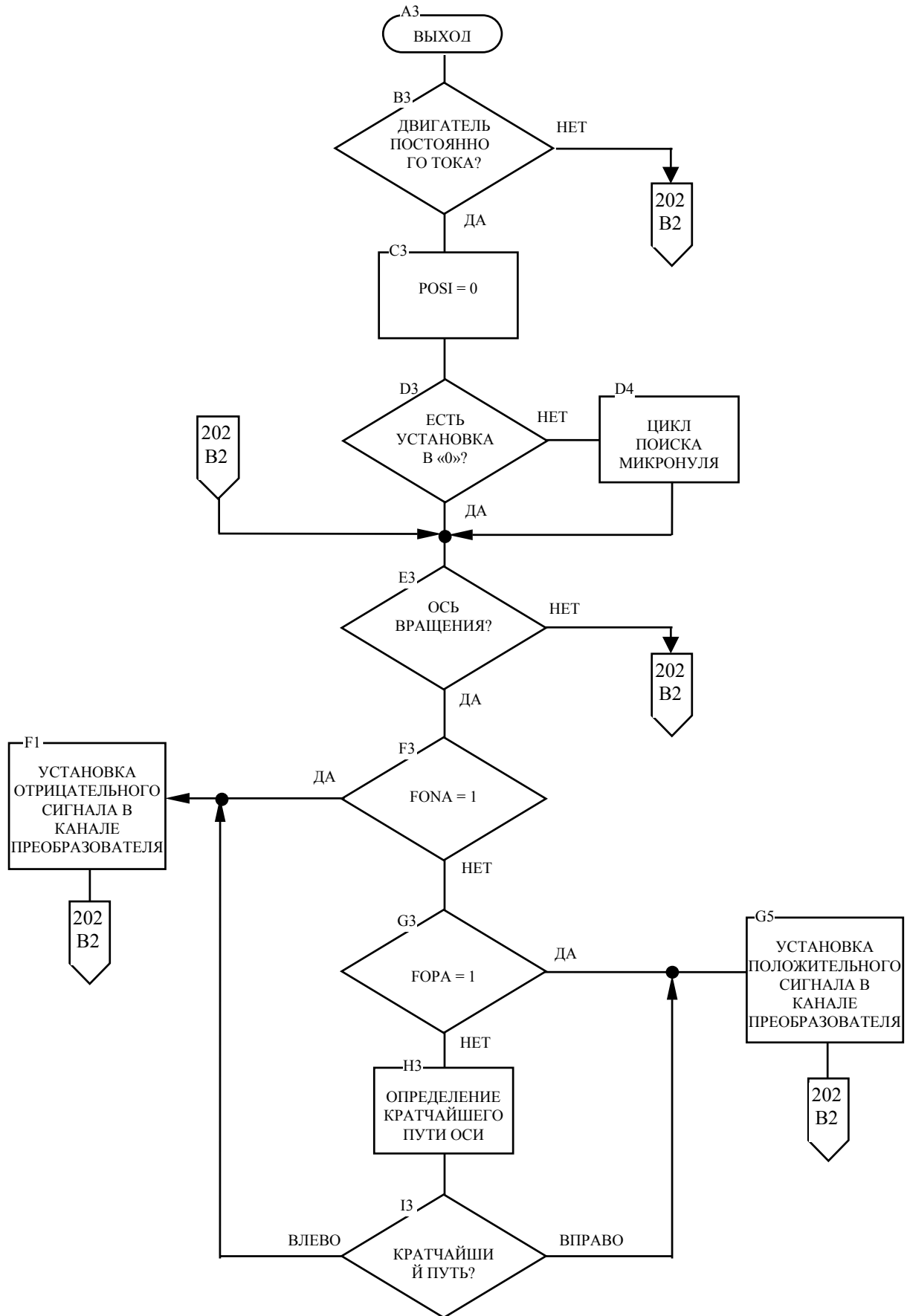


Рисунок А.27 (лист 1/2) - Цикл позиционирования осей с ЦАП

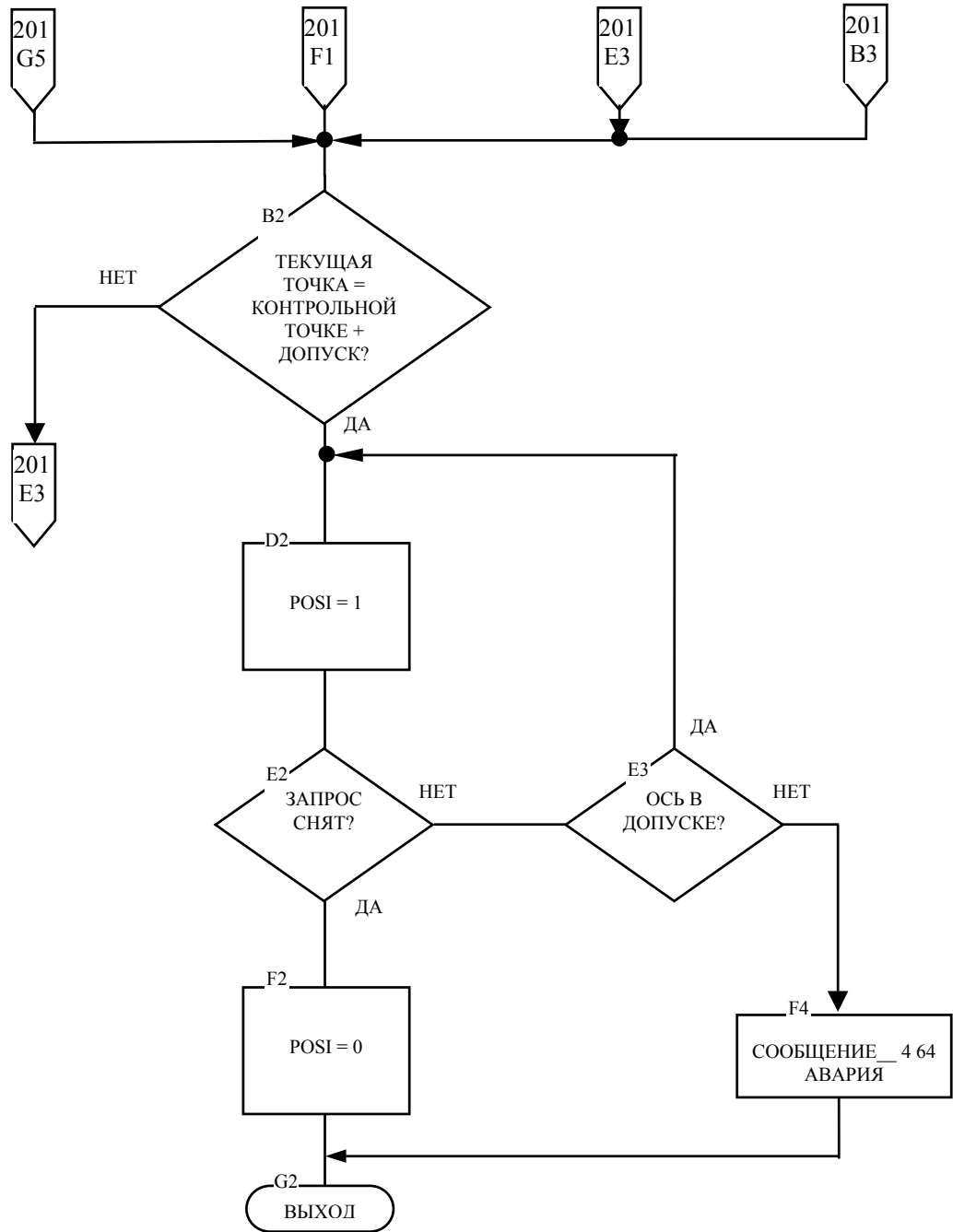


Рисунок А.27 (лист 2/2) - Цикл позиционирования осей с ЦАП

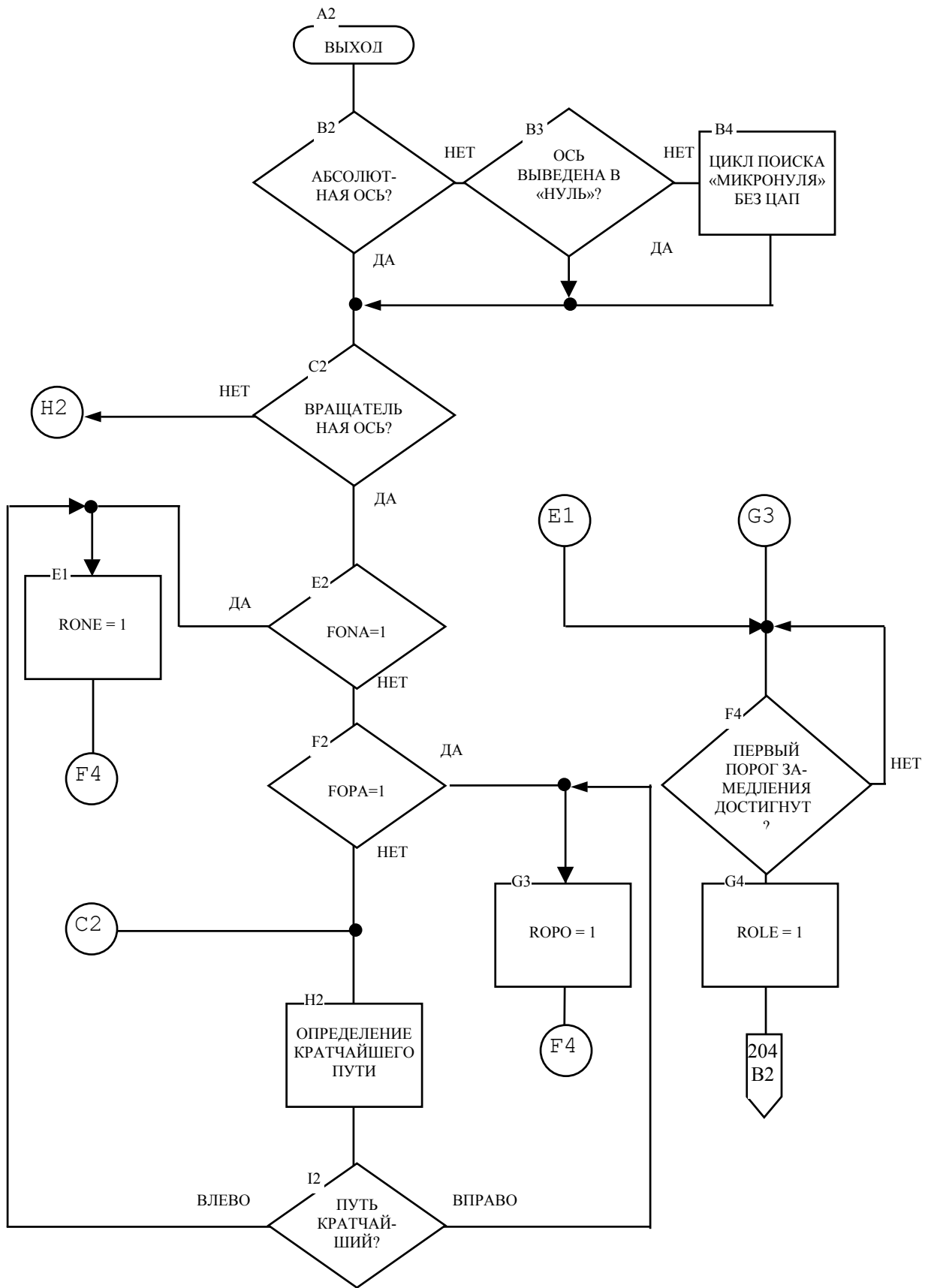


Рисунок А.28 (лист 1/2) - Цикл позиционирования осей без ЦАП

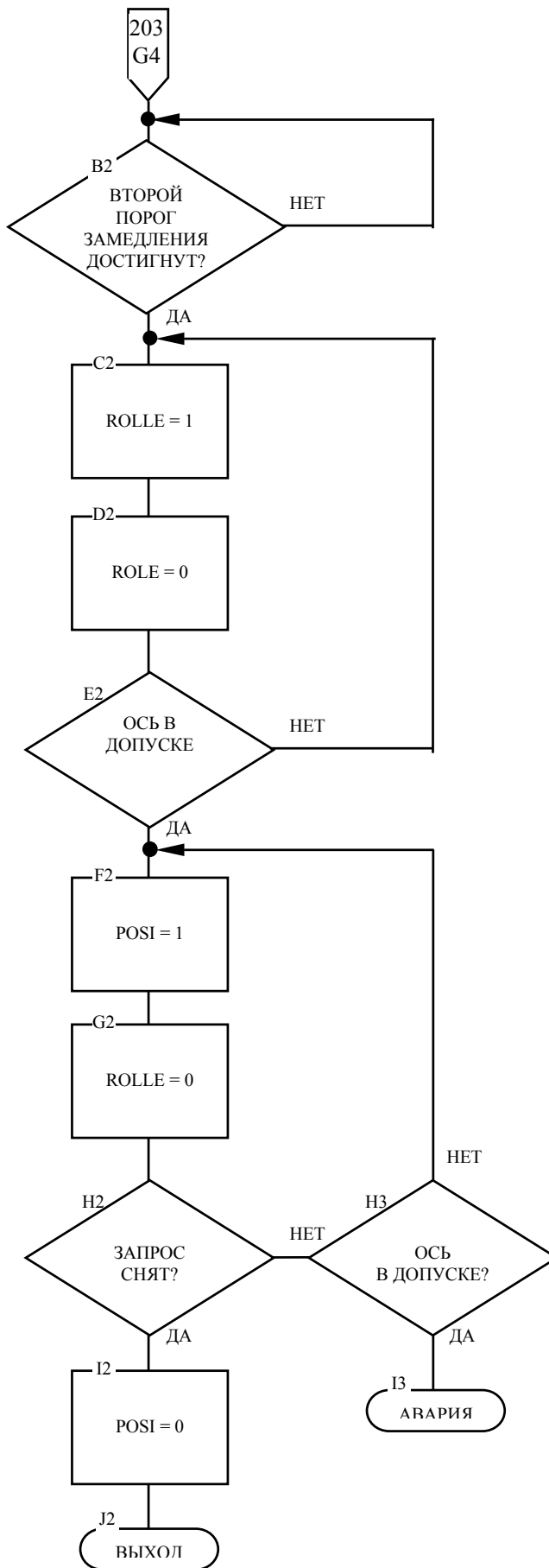


Рисунок А.28 (лист 2/2) - Цикл позиционирования осей без ЦАП

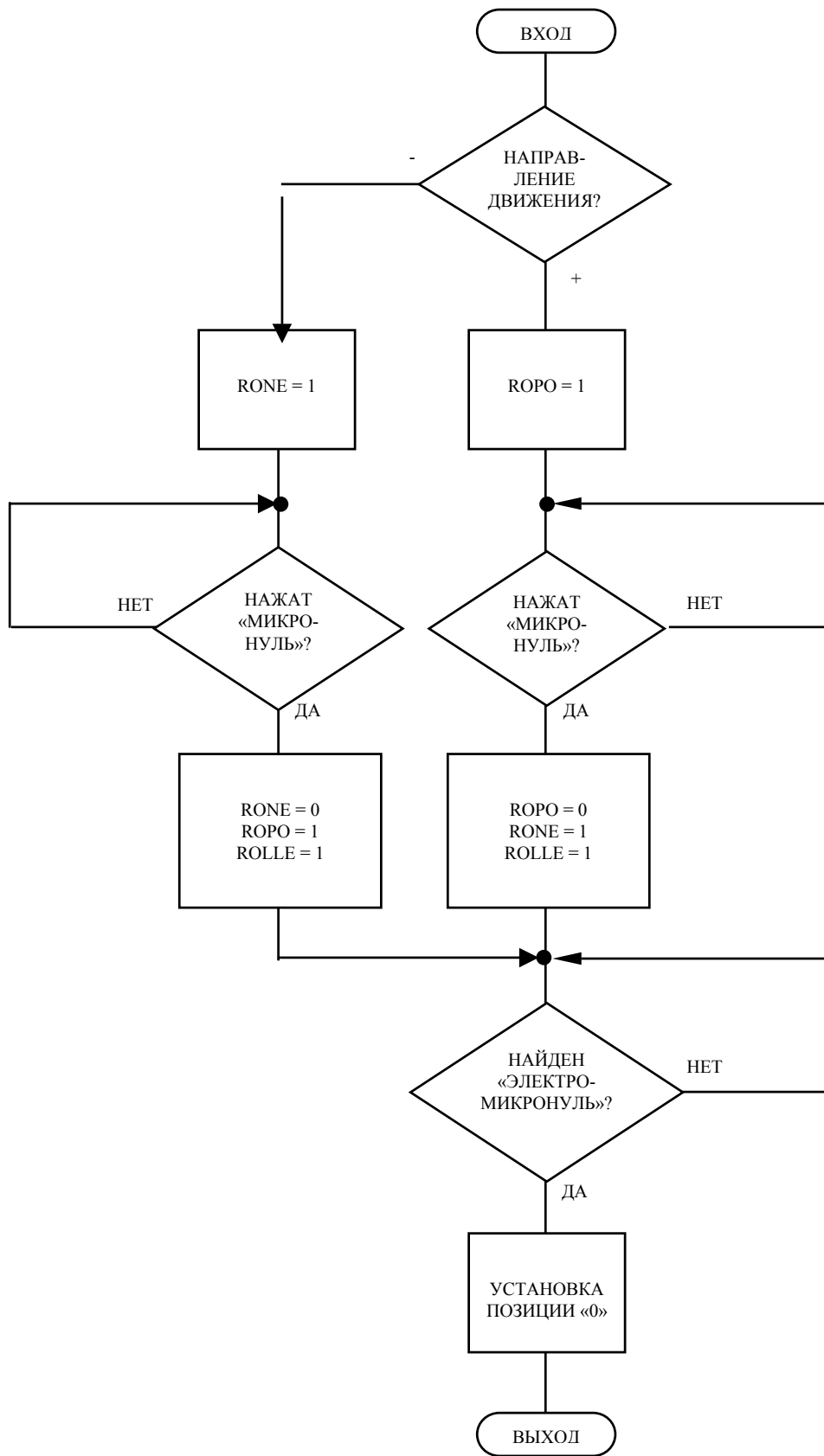


Рисунок А.29 - Цикл поиска микронуля осей «от точки к точке» без ЦАП