УСТРОЙСТВО
ЧИСЛОВОГО ПРОГРАММНОГО УПРАВЛЕНИЯ
NC-110, NC-310, NC-301, NC-302
NC-200, NC-201, NC-201M, NC-202, NC-210, NC-220, NC-230

Руководство программиста МС

Санкт-Петербург
2019г

АННОТАЦИЯ

Документ «Руководство программиста» (В2.1) предназначен для ознакомления с правилами и методами составления управляющих программ устройств числового программного управления типа NC-110, NC-200, NC-201, NC-201М, NC-202, NC-210, NC-220, NC-230, NC-301, NC-302 и NC-310 (в дальнейшем - УЧПУ), обеспечивающих управление металлообрабатывающим оборудованием, работающих как автономно, так и в составе гибких производственных модулей и гибких производственных систем.

СОДЕРЖАНИЕ

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Характеристики управления

1.1.1 Оси

- 8 управляемых осей: 8 осей в линейной интерполяции, 2 оси с перемещением «от точки к точке», 1 ось шпинделя;
- управление одновременно 8-ю осями, из которых 8 - непрерывных и скоординированных и 2 – «от точки к точке»;
- плоскость круговой интерполяции может быть применена к любой паре осей;
- винтовая интерполяция;
- сочетание круговой интерполяции с линейными и вращательными движениями;
- максимальный радиус - 99.9999 м;
- точность интерполяции в пределах одного микрона на метр радиуса;
- датчики установки положения: энкодеры (разрешающая способность 0,1 мкм), оптические линейки;
- автоматическое управление векторной скоростью на профиле;
- управление ускорением и замедлением при круговой интерполяции;
- автоматическое замедление на углах;
- динамическая оптимизация скорости на профиле;
- память конфигурируемого перехода (максимально 64 кадра) для непрерывной обработки.

1.1.2 Пульт управления

Пульт управления включает пульт оператора и станочный пульт/консоль, которые объединяют в себе все функции типа ввода/вывода в системе Оператор – УЧПУ - Станок.

Пульт включает алфавитно-цифровую клавиатуру, жидкокристаллический дисплей TFT 10.4”, замок с ключом для включения/выключения питания УЧПУ.

Станочный пульт/консоль включает корректоры для изменения скорости подачи, вращения шпинделя, для выбора направления и скорости ручных перемещений, а также клавиши/переключатель для выбора режима работы, кнопки «СТОП» и «ПУСК». В станочный пульт дополнительно могут быть установлены свободно программируемые клавиши и штурвал ручных перемещений.

1.1.3 Вывод алфавитно-цифровой информации

Вывод алфавитно-цифровой информации на экран дисплея осуществляется в режимах «КОМАНДА» и «УПРАВЛЕНИЕ СТАНКОМ».

Режим «КОМАНДА» используется при процедурах редактирования, визуализации списка программ, таблиц корректоров, исходных точек и срока службы инструмента.

Алфавитно-цифровая информация в режиме «УПРАВЛЕНИЕ СТАНКОМ» выводится на видеостраницы #1-#5 и #7, которые визуализируют:

- название программы;
- время обработки;
- сообщения оператору;
- реальные размеры осей;
- запрограммированные размеры осей;
- функции **G**, **T**, **S**, **M**;
- исходные точки;
- корректора;
- номер кадра;
- повтор циклов и подпрограмм.

1.1.4 Вывод графической информации

Графическая информация выводится в режиме «УПРАВЛЕНИЕ СТАНКОМ» на видеостраницу #6. Первые четыре строки видеокадра используются для воспроизведения

краткой информации, что и на видеокадре #1. В стадии обработки свободная часть видеоэкрана используется для воспроизведения декартовых осей, запрограммированных размеров, профилей и точек, на которых реализуются запрограммированные циклы и движения оси, перпендикулярной к плоскости обработки.

1.1.5 Запоминание программ и их модификация

Управляющие программы обработки детали должны быть занесены в память УЧПУ с клавиатуры или с периферийных устройств. Введённые символы в память программы могут быть воспроизведены на экране монитора и модифицированы посредством удаления, изменения или вставки кадров. Эти операции могут осуществляться во время обработки детали на станке.

1.1.6 Режимы работы

Режимы работы выбираются клавишами/переключателем со станочной панели/консоли:

- выполнение кадров, введённых с клавиатуры («MDI»);
- выполнение выбранной программы в автоматическом режиме («AUTO»);
- выполнение выбранной программы по кадрам («STEP»);
- выполнение безразмерных ручных перемещений («MANU»);
- выполнение фиксированных ручных перемещений («MANJ»);
- автоматический выход на профиль и продолжение работы после прерывания цикла обработки, за которым следовали ручные перемещения («PROF»);
- выход в «0» станка («HOME»).

1.1.7 Штурвал

Для выполнения ручных перемещений можно использовать штурвал. Существует два способа управления штурвалом.

1) Внутреннее управление. Управление максимум от одного штурвала. Включение/выключение штурвала выполняется командой VOL. Перемещение происходит с двумя различными шкалами:

   - 1 мм/об при безразмерных ручных перемещениях;
   - 0,1 мм/об при фиксированных ручных перемещениях.

2) Внешнее управление. Управление максимум от двух штурвалов. Шкала каждого штурвала назначается от ПЛ. Включение/выключение штурвала выполняется от ПЛ.

1.1.8 Проверка программ

При вводе команд с клавиатуры можно:

- проверить программы в памяти, без движения осей, используя графическую визуализацию или видеокадр «УПРАВЛЕНИЕ СТАНКОМ»;
- выполнить программу со скоростями обработки, равными скоростям быстрых перемещений.

1.1.9 Ноль станка

Один из ограничителей перемещения любой оси используется для автоматического выбора нуля системы отсчета. В рабочем состоянии системы при перемещении любой оси на этот ограничитель за точку абсолютного нуля станка принимается наиболее близкий шаг датчика.

1.1.10 Стоп

Система выполняет останов движения осей с контролируемым замедлением.

1.1.11 Компенсация люфта

Система обеспечивает автоматическую компенсацию люфта при изменении направления движения. Значение люфта устанавливается в памяти системы при характеризации.

1.1.12 Компенсация геометрических ошибок

Эта операция позволяет автоматически компенсировать (посредством вычислений, выполняемых системой) размеры, полученные датчиком положения. Компенсация геометрических ошибок может быть выполнена для любой оси. Количество точек компенсации устанавливается при характеризации (максимально 256 точек для каждой оси).

1.1.13 Датчики положения

Система работает с энкодером на оси шпинделя, энкодером и оптическими линейками на осях.

1.1.14 Абсолютные исходные точки

Вводя с клавиатуры трёхбуквенный код **ORA**, относящийся к сконфигурированным осям станка, можно определить максимально 100 абсолютных исходных точек (от 0 до 99). Например: **ORA, n, Z.., X...** Исходные точки можно активизировать из программы трёхбуквенным кодом **UAO**.

Исходные точки могут быть определены в той же системе измерения, в которой сконфигурирован станок, или в альтернативной системе установкой номера исходной точки с отрицательным значением. Например: **ORA, -n, Z..,X.. .**

1.1.15 Временные исходные точки

Кроме абсолютных исходных точек, используя трёхбуквенный код UOT, в программе можно определить бесконечное количество временных исходных точек, привязанных к любой из абсолютных.

1.1.16 Исходные точки в приращениях

Из программы можно определить, используя трёхбуквенный код UIO, бесконечное количество исходных точек в приращениях, т.е. относительно исходных точек (точки), существующих в момент определения.

1.1.17 Корректировки

Число коррекций не ограничено и определяется во время установки. Максимальное значение корректора Z=+(-)9999.999 мм по длине и K=999.999 мм для диаметров. Корректировка длины инструмента может быть применена для любой оси. Значения корректировки длины могут быть введены с клавиатуры или автоматически вычислены системой (при установке инструмента). Значения корректировки диаметра должны быть введены с клавиатуры.

Значение корректировки может быть воспроизведено и модифицировано в любой момент. Значение корректировки могут быть модифицированы программой после выполнения измерительного цикла.

1.1.18 Цикл контроля инструмента

Имеется возможность выполнить проверку инструмента с остановкой, ручными перемещениями и последующим возвращением в точку остановки. Возврат в точку остановки может быть выполнен вручную ось за осью по выбору оператора (RAP=0) или же автоматически, повторяя в обратном направлении порядок ручных перемещений, выполненных при отводе (RAP=1). Максимальное число перемещений - 32.

1.1.19 Управление головками для расточки и обточки

Головки расточки и обточки, установленные на шпинделе, управляются как одновременные и скоординированные оси. Ось, относящаяся к головке расточки и обточки, программируется в диаметрах.

1.1.20 Управление электронным щупом

Устройство измерения по всем направлениям, установленное на шпинделе, рассматривается как инструмент с коррекциями по длине и диаметру. Параметры измерения щупом: размер подхода, размер надёжности и скорость измерения - заносятся в память с клавиатуры посредством трёхбуквенного кода DPT. Параметры, не присутствующие в управлении, определяются во время конфигурации системы.

Посредством функций G72 и G73, которые могут быть внесены в программу обработки, щуп реализует:

- измерение координат точки в пространстве;
- измерение координат центра и радиуса окружности в плоскости.

Параметры, полученные измерением, накапливаются в памяти посредством параметра E, находящегося в кадре измерения.

С инструментом, установленным в шпинделе, зафиксированный щуп реализует измерение смещений от теоретических точек посредством функции G74, вставляемой в программу обработки. Этот цикл может быть использован для переквалификации или контроля целостности инструмента.

1.1.21 Цикл срока службы инструмента

Для каждого инструмента можно определить срок службы (использование в обработке), что позволяет контролировать состояние инструмента. Кроме того, можно запрашивать замену использованного или вышедшего из строя инструмента другим, пригодным для использования, с такими же характеристиками.

Управление циклом срока службы инструмента осуществляется посредством использования таблицы, содержащей характеристики инструмента:

- номер инструмента;
- номер альтернативного инструмента;
- корректор, который надо применить к альтернативному инструменту;
- максимальный теоретический срок службы;
- минимальный теоретический срок службы;
- остаточное время службы;
- состояние инструмента.

1.1.22 Поиск информации в памяти

Возможен поиск вперёд и назад до введенного слова (N18; G33; M5; X80.5 и т.д.). Кроме того, командами, введёнными с клавиатуры, можно:

- остановить обработку кадра с заданным порядковым номером;
- выполнить или исключить из выполнения кадры, разделённые дробной чертой.

1.1.23 Запомненный поиск

Система сохраняет в устройствах постоянной памяти некоторые параметры, которые однозначно определяют выполняемый кадр.

На основе этих параметров, таким образом, возможно автоматическое возобновление цикла с места прерывания. Это возможно даже в наиболее критических случаях, таких как повторяющиеся циклы, составные циклы, условные переходы, вызовы подпрограмм. Оператор должен только ввести с клавиатуры код автоматического поиска RCM и код конца поиска ERM. Система имитирует функционирование до кадра, выполненного полностью, вызывает требуемый инструмент, устанавливает коррекции и воспроизводит на дисплей координаты, на которых должен бы быть инструмент, и координаты его фактического нахождения.

Для возобновления обработки достаточно дать «ПУСК» после позиционирования осей.

1.1.24 Типы памяти

Инструкции, характеризующие поведение системы (которые, например, отличают управление фрезерными обрабатывающими центрами от управления токарным станком), находятся в ПрО.

Параметры металлорежущего станка (например: скорость, ускорение и т.д.), значения корректировки длины и диаметра инструмента, исходные точки, таблица срока службы инструмента и управляющие программы обработки детали занесены в память (**HDD, FLOPPY, FLASH**).

Данные временного использования находятся в памяти ОЗУ без сохранения содержимого при выключении питания.

1.1.25 Изменение скорости подачи и вращения

На пульте оператора расположены клавиши, которые могут изменять:

- скорость подачи от 0-125% («F»);
- скорость вращения шпинделя от 75-125% («S»).

1.1.26 Система защиты и автодиагностики

Осуществляется как контроль аппаратной части системы, подверженной повреждениям (центральная вычислительная система, кабельные связи, датчики положения и т.д.), так и контроль правильности функционирования (внутренняя температура, напряжение питания, паритет входных данных и переполнение памяти, команды с клавиатуры и т.д.). Ошибки сервомеханизмов также находятся под постоянным контролем вычислительной системы.

При неисправности аппаратной части или ошибке функционирования на экран выводится диагностическое сообщение с указанием причины обнаруженного дефекта (указывается модуль, в котором обнаружена неисправность, или аномальная ситуация, которую надо исправить). Диагностические сообщения хранятся в файле характеризации системы.

1.2 Характеристики программирования

1.2.1 Система измерения

Миллиметры или дюймы, выбираемые посредством функции G71/G70.

1.2.2 Программирование абсолютных размеров или по приращениям

Подготовительная функция: G90 - абсолютное программирование, G91 - программирование по приращениям.

1.2.3 Программирование относительно нуля станка

Перемещения, запрограммированные в кадре, могут быть отнесены к нулю станка заданием функции G79.

1.2.4 Программирование с десятичной точкой

Размеры программируются так, как читаются (без нулей в начале или в конце), с указанием точки разделения целой части от десятичной.

Пример: **X-20.275** .

1.2.5 Код ленты

EIA RS244, ISO 840 с автоматическим распознаванием.

1.2.6 Формат программирования

N4, G2, X/Y/Z/A/B/C/U/W/V/P/Q/D/5.4,R5.4,I/J/K5.4, F5.2, S5.2, T4.4, M2, H2.

1.2.7 Координаты осей

Координаты программируются в миллиметрах или дюймах от +/-0.0001 до +/-99999.9999.

1.2.8 Координаты I, J

Определяют координаты центра окружности в круговой интерполяции **I** по оси абсцисс и **J** по оси ординат. Программируемое значение: от +(-) 0.0001 до +(-) 99999.9999 миллиметров или дюймов.

1.2.9 Вращательные движения

Во время характеризации системы любая ось может быть определена как ось вращения. Программируемое значение: от +/-0.0001 до +/-99999.9999 градусов.

1.2.10 Функция F

Программируется от 0.01 до 99999.99.

- **G94** определяет скорость подачи осей в мм/мин или дюйм/мин; с помощью символа «**t**» можно программировать время в секундах, необходимое для отработки элемента, определённого в кадре («**F**» для кадра является отношением между длиной элемента и запрограммированным «**t**»);
- **G93** определяет обратное время, т.е. отношение скорость подачи/расстояние;
- **G95** определяет скорость осей в мм/оборот.

1.2.11 Функция S

Программируется от 0.01 до 99999.99. Может выражать:

- число оборотов/мин шпинделя (G97);
- скорость резания в м/мин (G96).

1.2.12 Функция T

Определяет требуемый для обработки инструмент и номер коррекции для данного инструмента. Программируемая величина: от 1.0 до 9999.9999. Цифры перед десятичной точкой определяют инструмент, после - номер корректора.

1.2.13 Подготовительные функции G

- G00 быстрое позиционирование;
- G01 линейная интерполяция;
- G02 интерполяция круговая по часовой стрелке;
- G03 интерполяция круговая против часовой стрелки;
- G04 выдержка времени, заданная в кадре;
- G06 сплайновая интерполяция;
- G09 замедление в конце кадра;
- G17 выбирает плоскость интерполяции, определенную конфигурируемыми осями 1 и 2;
- G18 выбирает плоскость интерполяции, определенную конфигурируемыми осями 3 и 1;
- G19 выбирает плоскость интерполяции, определенную конфигурируемыми осями 2 и 3;
- G20 закрывает среду программирования языка GTL;
- G21 открывает среду программирования языка GTL;
- G27 непрерывная обработка с автоматическим уменьшением скорости на углах;
- G28 непрерывная отработка без автоматического уменьшения скорости на углах;
- G29 позиционирование «от точки к точке»;
- G33 нарезание резьбы с постоянным или изменяющимся шагом;
- G34 нарезание резьбы с постоянным или изменяющимся шагом;
- G35 синхронизированное начало движения со шпинделем;
- G40 отмена корректировки на профиле;
- G41 приводит в действие корректировку на профиле (инструмент слева);
- G42 приводит в действие корректировку на профиле (инструмент справа);

- G70 программирование в дюймах;
- G71 программирование в миллиметрах;
- G72 измерение точки с компенсацией радиуса инструмента;
- G73 измерение параметров отверстия;
- G74 измерение отклонения от теоретической точки без компенсации радиуса инструмента;
- G79 программирование относительно нуля станка (действительно только в данном кадре);
- G80 отмена постоянных циклов;
- G81 цикл сверления;
- G82 цикл растачивания;
- G83 цикл глубокого сверления;
- G84 цикл нарезания резьбы метчиком;
- G85 цикл рассверливания;
- G86 цикл развертывания;
- G89 цикл развертывания с остановкой;
- G90 абсолютное программирование;
- G91 программирование по приращениям;
- G93 скорость подачи, выраженная в виде обратного времени выполнения;
- G94 скорость подачи осей, мм/мин или дюйм/мин;
- G95 скорость подачи осей, мм/оборот;
- G96 скорость вращения шпинделя, м/мин;
- G97 скорость вращения шпинделя, оборот/мин.

1.2.14 Вспомогательные функции М

- M00 остановка программы;
- M01 условная остановка программы;
- M02 конец программы;
- M03 вращение шпинделя по часовой стрелке;
- M04 вращение шпинделя против часовой стрелки;
- M05 остановка вращения шпинделя;
- M06 замена инструмента;
- M07 включение вспомогательного охлаждения;
- M08 включение основного охлаждения;
- M09 выключение охлаждения;
- M10 блокировка осей;
- M11 разблокировка осей;
- M12 блокировка вращающихся осей;
- M13 вращение шпинделя по часовой стрелке и охлаждение;
- M14 вращение шпинделя против часовой стрелки и охлаждение;
- M19 остановка вращения шпинделя с угловой ориентацией;
- M30 конец программы и возврат к первому кадру;
- M41 ¦
- M42 ¦ выбирает диапазон вращения шпинделя
- M43 ¦ 1-2-3-4
- M44 ¦
- M40 аннулирует диапазон вращения шпинделя;
- M45 автоматическая замена диапазона;
- M60 замена детали.

1.2.15 Постоянные циклы

С использованием подготовительных функций G81-G89 определения подготовительного цикла можно программировать ряд операций (сверление, нарезание резьбы метчиком, растачивание и т.д.) без повторения для каждой из них параметров отверстия, запрограммированную обработку которого надо осуществить. Последовательность движений циклов может быть установлена в следующем порядке:

- быстрое позиционирование к оси отверстия;
- быстрый подход к плоскости обработки (размер R);

- рабочая скорость подачи до запрограммированного размера Z;
- фиксации цикла на дне отверстия;
- ускоренное или со скоростью обработки возвращение к точке R.

Можно программировать размер возвращения R2, отличный от R (тогда два размера R в кадре).

Характеристики постоянных циклов приведены в таблице 1.1.

Таблица 1.1 - Характеристики постоянных циклов

Постоянный цикл	Подход	Функции на дне отверстия		Возврат
		выдержка времени	вращение шпинделя	
G 81 сверление	рабочая подача	нет	рабоч.скор.	быстрый ход
G 82 растачивание	рабочая подача	да	рабоч.скор.	быстрый ход
G 83 глубокое сверление (с разгрузкой стружки)	в прерывистой работе	нет	рабоч.скор.	быстрый ход
G 84 нарезание резьбы метчиком	рабочая подача, начало вращения шпинделя	нет	изменение направления	рабочая подача
G 85 рассверливание или tapmatic	рабочая подача	нет	рабоч.скор.	рабочая подача
G 86 развертывание	рабочая подача начало вращения шпинделя	нет	останов	быстрый ход
G 89 развертывание с растачиванием	рабочая подача	да	рабоч.скор.	рабочая подача

1.2.16 Постоянный цикл нарезания резьбы метчиком с датчиком на шпинделе

В этом цикле скорость подачи **F** не программируется, т.к. вычисляется автоматически в соответствии с числом оборотов шпинделя и шага **K** метчика нарезания резьбы.

1.2.17 Изменение скорости возвращения при нарезании резьбы метчиком

Определяя процентное содержание изменения посредством кода RMS, введенного в программу или накопленного в памяти с клавиатуры, можно модифицировать скорость возврата в цикле нарезания резьбы метчиком.

Пример
RMS=110 (+10%запрограммированного F)
RMS=10 (-90% запрограммированного F)

1.2.18 Выдержка времени

Выражается в секундах заданием кода TMR, введённого с клавиатуры.

Пример
TMR=2

1.2.19 Время обработки

Группа из трёхбуквенных кодов TIM позволяет пользователю определить время обработки в определенных точках программы. Трёхбуквенный код TOT позволяет дополнительно программировать 6 специальных времён в определенных точках обработки.

1.2.20 Сообщения программы

На втором видеокадре в зоне сообщений могут быть воспроизведены сообщения, переменные, константы, которые программируются посредством трёхбуквенного кода DIS.

Примеры
(DIS, «ИНСТРУМЕНТ=12»);
(DIS, E37);
(DIS, UOV).

1.2.21 Коэффициент масштабирования

Коэффициент масштабирования применяется для масштабирования заданного перемещения для определённых осей. Программируют трёхбуквенный код SCF и коэффициент масштабирования, который необходимо применить.

Примеры
(SCF,2) для всех осей,
(SCF,2,X) для оси X.

1.2.22 Нарезание резьбы

С функцией G33 программируется цикл цилиндрического или конического нарезания резьбы, с постоянным или переменным шагом. Параметры, запрограммированные в кадре, определяют тип нарезания резьбы.
Формат:

G33 Z..K.. - цилиндрическое нарезание резьбы с постоянным шагом,
G33 Z..U..K.. - коническое нарезание резьбы с постоянным шагом,
G33 Z..K..I+. - нарезание резьбы с увеличивающимся шагом,
G33 Z..K..I-. - нарезание резьбы с уменьшающимся шагом,

где:
G33 - подготовительная функция;
Z,U - координаты конечной точки;
K - шаг нарезания резьбы;
I+/- - изменение шага.

1.2.23 Векторная компенсация радиуса инструмента

Векторная компенсация радиуса инструмента позволяет осуществить программирование контуров профиля без учета радиуса инструмента. Корректировка радиуса действует в перпендикулярном направлении к запрограммированному профилю и приводится в действие при помощи функций:

- **G41** - корректировка слева от профиля,
- **G42** - корректировка справа от профиля.

Параметры корректировки, которую надо применить к паре осей для их коррекции, вычисляются автоматически. Корректировка отменяется функцией **G40**.

1.2.24 Определение припуска

Кодом UOV можно определить припуск в операциях контурной обработки. Заданный в программе или введённый с клавиатуры код UOV временно модифицирует значение корректировки на величину, равную установленному значению. Отмена припуска программируется установкой кода UOV=0.

Пример
UOV=1.5

1.2.25 Осепараллельные коррекции радиуса инструмента

С использованием в кадре обработки факторов корректировки **u, v, w** можно выполнить корректировку конечной точки, запрограммированной для декартовых осей станка. При этом конечная точка вычисляется следующим образом:

$$Pi = Qi + r * Fi \qquad (1.1),$$

где:
Qi - запрограммированные размеры для оси;
R - радиус инструмента;

Fi - фактор корректировки, может быть:

- **u** для оси 1 или ее замены;
- **v** для оси 2 или ее замены;
- **w** для оси 3 или ее замены.

1.2.26 Зеркальная обработка

Трёхбуквенный код MIR позволяет зеркальную обработку для всех скоординированных осей.

Пример
```
(MIR,X)
.........
(MIR,Y)
.........
(MIR,X,Y)
```

1.2.27 Вращение в плоскости

Программируя трёхбуквенный код URT, можно вращать в плоскости часть или всю запрограммированную деталь. Вращение происходит вокруг начальной точки, активной в этот момент.

Пример
```
(URT,45)
```

1.2.28 Повторение программ

Используя трёхбуквенный код RPT, можно повторять **n** раз программу или часть программы для создания специальных циклов. Максимальное количество повторений - 99. Внутри повторяющегося цикла можно создать другой цикл, а в нём - ещё один (до трёх уровней). Часть программы, которую необходимо повторить, закрывается трёхбуквенным кодом ERP.

Пример
```
(RPT,99)
..........
..........
(ERP)
```

1.2.29 Параметрическое программирование

С помощью кода **E** можно программировать параметрические, геометрические и технологические данные цикла обработки. При помощи параметров можно осуществлять математические и тригонометрические действия, вычисление выражений. Максимальное число параметров **E** не ограничено и определяется во время конфигурации системы. Параметры **E** предусматривают различные индексы для переменных различного формата. Параметры **E** для различных форматов приведены в таблице 1.2.

Таблица 1.2 – E-параметры

Формат	Параметры	Значение мин/макс
BY (байт)	E0….E9	От 0 до 255
IN (целое)	E10...E19	От минус 32768 до плюс 32768
LI (целое с двойной точностью)	E20...E24	От минус 2.147.483.647 до плюс 2.147.483.647
RE (действительное)	E25...E29	±7 целых или десятичных чисел
LR (действительное с двойной точностью)	E30...En	±16 целых или десятичных чисел

Арифметические действия:

1) + сложение;
2) - вычитание;
3) * умножение;
4) / деление.

Функции:

1) SIN(A) - вычисляет синус **A**;
2) COS(A) - вычисляет косинус **A**;
3) TAN(A) - вычисляет тангенс **A**;
4) ARS(A) - вычисляет арксинус **A**;
5) ARC(A) - вычисляет арккосинус **A**;
6) ART(A) - вычисляет арктангенс **A**;
7) SQR(A) - вычисляет квадратный корень **A**;
8) ABS(A) - вычисляет абсолютное значение **A**;
9) INT(A) - вычисляет целую часть **A**;
10) NEG(A) - инвертирует значение **A**;
11) MOD(A/B) - вычисляет остаток отношения между **A** и **B**;
12) FEL(A,B) - вычисляет элемент индекса **B** (1,2,3) из геометрического элемента (прямая линия) индекса **A**.

Индексы **A** или **A,B** могут быть параметрами **E** или цифровыми значениями.

Геометрические и технологические данные (G, F, S, X, Z, Y, начальные точки и т.д.), определяющие цикл обработки, могут быть представлены параметрами, значение которых определяется в основной программе до вызова данной подпрограммы.

1.2.30 Вычисление выражений

Можно выполнять выражения, содержащие постоянные, параметры, функции.

Пример
```
N1 E37=E31*SIN(E30)+123.4567/SQR(16).
```

Пример кадров назначения для вычисления переменных:

```
"LAB 1" E51 = -0.00000124 +5/E35 = FEL(37,1)
E7 = 81
E10 = 1
E25 = E25 + 30
```

Параметры **E** могут быть использованы как внутри программы, так и внутри подпрограммы, и могут быть воспроизведены на экране дисплея.

1.2.31 Параметрические подпрограммы

Под подпрограммой понимается последовательность кадров, определяющая пользовательский цикл обработки, которая может быть вызвана из основной программы. Подпрограмма может вызывать только одну подпрограмму (2 уровня вложенности). Подпрограммы хранятся в памяти пользователя, их количество зависит только от их длины и от объёма используемой памяти. Подпрограмма вызывается трёхбуквенным кодом CLS.

Пример
```
N35 (CLS,PROG1)
```

1.2.32 Переходы в программе

Внутри программы можно программировать переходы посредством программирования инструкций, содержащих метку для передачи управления. Метка - это алфавитно-цифровая последовательность, состоящая из 6 символов, заключённых в знак « » (кавычки), которая должна быть запрограммирована перед номером кадра и после знака «/» в случае, если кадр разделён дробной чертой.

Пример
```
/"НАЧАЛО"N125
```

Переходы могут быть условными и безусловными. Коды переходов и соответствующие им форматы приведены в таблице 1.3.

VAR1 и VAR2 являются сравниваемыми переменными, могут быть параметрами, сигналами логики станка, цифровыми величинами или последовательностью символов.

Таблица 1.3 - Коды переходов

Формат	Функции
(BNC, метка)	Безусловный переход к метке
(BGT,VAR1,VAR2,метка)	Переход в случае, если VAR1 больше VAR2
(BLT,VAR1,VAR2,метка)	Переход в случае, если VAR1 меньше VAR2
(BEQ,VAR1,VAR2,метка)	Переход в случае, если VAR1 равен VAR2
(BNE,VAR1,VAR2,метка)	Переход в случае, если VAR1 отличен от VAR2
(BGE,VAR1,VAR2,метка)	Переход в случае, если VAR1 больше или равен VAR2
(BLE,VAR1,VAR2,метка)	Переход в случае, если VAR1 меньше или равен VAR2

Пример

N10 (BGT,E1,123,END) — переходит к END, если значение переменной E1 больше 123.

N20 (BEQ,SA3,1,LAB) — переходит к LAB, если булевская переменная SA3 включена.

N30 (BNE,E1,E5,START) — переходит к START, если значение переменной E1 отлично от значения E5.

N40 (BEQ,SYVAR1.CH,OK,END) — переходит к END, если символы SYVAR1.CH = OK

1.2.33 Измерительные циклы

С помощью щупа представляется возможным выполнить три цикла измерения, программируя функции G.

G72 - осуществляет измерения координат точки в пространстве с линейным движением и запоминает их в последовательности параметров E, первый из которых объявлен в кадре. Измерение выполняется с компенсацией радиуса щупа.

Пример

G 72 Z200 X50 E32

В E32 и E33 заносятся соответственно вычисленные величины для Z и X.

G73 - осуществляет измерение параметров отверстия в данной плоскости интерполяции и заносит эти величины в параметры E, первый из которых объявлен в кадре.

Полученные параметры являются координатами центра и радиуса отверстия. Измерение выполняется с компенсацией радиуса щупа.

Пример

G73 R100 E35

В E35-E36-E37 заносятся соответственно абсцисса, ордината и радиус окружности.

G74 - осуществляет измерение отклонений от теоретических точек инструментом, установленным в шпинделе, относительно закреплённого щупа, а также заносит данные измерений в параметры E аналогично функциям **G72** и **G73**. Этот цикл может быть использован для переквалификации и контроля целостности инструмента.

Пример

G74 X50 E40 (максимально 3 оси в кадре). Разница между измеренными и теоретическими значениями записывается в параметр E40. Значения параметров E, запомненные при выполнении измерительных циклов, могут быть использованы для программирования переквалификации начальных точек, инструмента и определения целостности инструмента.

1.2.34 Выполнение частей программы

Программируя трёхбуквенный код EPP, можно выполнить часть программы, заключенной между двумя метками.

Пример

"НАЧАЛО"

....

....

```
"КОНЕЦ"
(ЕРР,НАЧАЛО,КОНЕЦ)
........
```

После выполнения части программы программа продолжается от кадра, следующего после команды ЕРР.

1.2.35 Модификация исходных точек

Посредством программирования трёхбуквенного кода RQO и использования параметров **E**, накопленных в памяти при выполнении измерительных циклов (**G72-G73**), можно выполнить переквалификацию начальных точек.

Пример
(RQO,0,XE35)
E35 - разница между измеренным и теоретическим размерами.

1.2.36 Переквалификация инструмента

Переквалификация инструмента осуществляется при программировании трёхбуквенного кода RQU. Значения переквалификации обычно запоминаются в параметрах **E**, используемых в измерительных циклах.
Формат следующий:

(RQU,Nut,Ncorr,ZEn,KEm) ,

где:

Z - ось, к которой присоединяется корректор длины;
K - диаметр инструмента;
Nut - номер инструмента;
Ncorr - номер коррекции.

Номер инструмента определяется при управлении сроком службы инструмента, т.к. корректор, который надо модифицировать, может быть тем, который присоединён к альтернативному инструменту. Если таблица корректоров была составлена для запоминания также и значения измеренной корректировки, то команда RQU обновляет её, объявляя инструмент непригодным, если эти значения превышают максимальные допустимые значения. Программируя код RQP, вместо кода RQU, система модифицирует только корректоры длины и диаметра без обновления значения внесенной корректировки.

1.2.37 Целостность инструмента

Целостность инструмента в шпинделе может быть проверена посредством измерения при помощи цикла измерения **G74**. Сравнение запрограммированной величины допустимого допуска и величины отклонения, накопленной в памяти при цикле измерения, даёт возможность объявить инструмент пригодным или неисправным посредством трёхбуквенного кода TOF.

Пример
(TOF,12) - инструмент 12 неисправен.

1.2.38 Канал между программой и логикой станка

Обмен данными между пользовательской программой и интерфейсом логики возможен путем определения в пользовательской программе параметров входа/выхода через переменные интерфейса логики пакета «**K**» (переменные SK) и пакета «**A**» (переменные SA).

Системные структуры данных:

- пакет «**A**», определяющий все электрические сигналы типа включен/выключен, которые соединяют УЧПУ с оборудованием;
- пакет «**K**», определяющий все переменные связи между прикладным ПрО и интерфейсом логики станка.

Примеры присвоения:

SA12=SK.BL - придаёт биту N12 пакета «А» значение первого бита структуры пакета «К».

SK5=SK7 - придаёт байту N5 пакета «К» значение байта N7 этого же пакета.

SA128=1 - устанавливает сигнал (бит) N128 пакета «А».

SK7.3CH="RIF" - записывает инструкцию RIF, начиная с байта N7 пакета «К».

SA3.BY=255 - придаёт значение 255 байту N3 пакета «А».

1.2.39 Программные ограничители хода

Система следит за тем, чтобы запрограммированные движения не выходили за пределы рабочего поля станка (как линейные, так и круговые движения), подавая сигнал ошибки в случае, если это произошло. Контроль осуществляется перед началом движения. Предельные значения рабочего поля запоминаются в файлах характеризации системы и могут быть временно модифицированы посредством трёхбуквенного кода DLO внутри программы. В случае ручных перемещений сигнал ошибки подаётся в момент перемещения за пределы рабочего поля.

1.2.40 Ограничение рабочего поля

При помощи трёхбуквенного кода DLO из программы можно менять по любой оси пределы рабочего поля, запомненные в файлах системы.
Формат программирования:

(DLO,X-X+) ,
(DLO,Z-Z+) ,

где:

X- - нижний предел по X;
X+ - верхний предел по X;
Z- - нижний предел по Z;
Z+ - верхний предел по Z.

Запрограммированные пределы относятся к данным начальным точкам.

Пример

```
N20 (DLO,X-50 X100)
N21 (DLO,Z-60 Z20)
```

1.2.41 Программирование защищенных зон

Посредством трехбуквенного кода DSA из программы можно определить до трёх защищённых зон, т.е. три зоны, в которые запрещается вход инструмента. Контроль осуществляется до начала движения.

Формат:

(DSA,n,Z-Z+,X-X+) ,

где:

n - номер зоны для защиты (1 до 3);
Z- - нижний предел по Z;
Z+ - верхний предел по Z;
X- - нижний предел по X;
X+ - верхний предел по X.

Контроль защищённых зон приводится в действие посредством трёхбуквенного кода ASC и отменяется трёхбуквенным кодом DSC.

Формат:

(ASC,n) ,
(DSC,n) ,

где:

n - номер защищённой зоны.

Пример

```
(DSA,1,Z0,Z50,X5 X100)
(DSA,2,Z-100 Z-50,X-20 X150)
(ASC,1)
(ASC,2)
..........
(DSC,1)
..........
```

1.2.42 Геометрическое программирование высшего уровня

С этим видом программирования предоставляется возможность описать любой геометрический профиль на плоскости, состоящий из прямых линий и окружностей, с использованием информации, данной на рисунке. Система сама вычисляет точки касания и точки пересечения геометрических элементов. Определение профиля с использованием языка геометрического программирования высшего уровня основано на использовании четырёх типов геометрических элементов и на определении автоматических пересечений между элементами профиля. Используются следующие типы геометрических элементов:

1) точки начала отсчёта;
2) точки;
3) прямые линии;
4) окружности.

Максимальное число элементов определяется во время цикла конфигурации. Элементы могут иметь индекс в виде цифрового значения или параметра **E**. Геометрические элементы определяются параметрами, необходимыми для установки позиции на плоскости, а также направлением движения. Функции G21 и G20 определяют профиль, т.е. ряд геометрических элементов, соединенных конкретным образом. Сначала геометрические элементы должны быть занесены в память системы. Профиль может быть либо открытым, либо закрытым. Открытый профиль начинается с одной точки и кончается другой точкой, отличной от первой; закрытый профиль начинается и заканчивается с одной и той же точкой. Имеется возможность перемещать любую ось, не участвующую в контурном движении, в любую точку на профиле. Список возможных определений геометрических элементов представлен в таблице 1.4.

1.2.43 Виртуальные оси

Виртуальные оси используются в следующих случаях:

1) для расчёта и выполнения профилей в полярных координатах;
2) для расчёта и выполнения профилей в цилиндрических координатах;
3) для преобразования косоугольной системы реальных координат в декартовую систему виртуальных координат;
4) для разворота плоскости обработки в пространстве под любым углом.

1.2.44 Программный интерфейс (PLC)

Интерфейс УЧПУ - металлорежущий станок программируется с использованием системного модуля PLC. Вставляя в систему модуль PLC, система позволяет записывать, исправлять и проверять непосредственно на УЧПУ в реальных условиях программу логики, разрабатываемую для конкретного станка. После того как осуществлена проверка программы логики станка, можно записать её в постоянную память, содержащую системное ПрО.

Этот тип программирования интерфейса позволяет очень быстро и просто модифицировать и обновлять сам интерфейс, делая, таким образом, УЧПУ более надёжным.

Используя интерфейс, можно воспроизводить на мониторе сообщения оператору для выявления аномальных ситуаций.

1.2.45 Сплайновая интерполяция

Сплайновая интерполяция применяется для объединения последовательности отдельных точек в гладкий непрерывный контур.

В ПрО УЧПУ сейчас реализован «С»-сплайн. «С»-сплайн обеспечивает гладкий контур с точным прохождением через все точки сплайна, с непрерывной кривизной и возможностью задания условий на его краях.

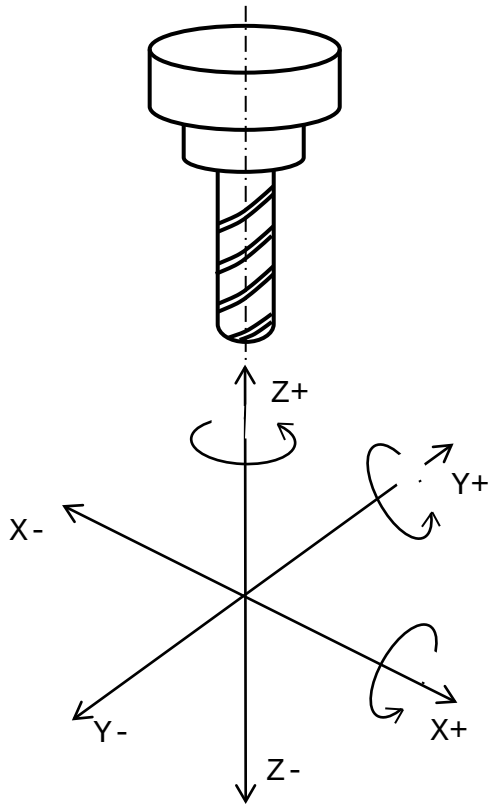
Таблица 1.4 - Геометрические элементы

Элемент	**Определение**	**Описание**
Точки начала отсчета	on=ZXa	
Точки	pn=(om) X Y pn=(om) m a pn=+lm,+lp pn=+lm,+cp(,s2) pn=+cm,+lp(,s2) pn=+cm,+cp(,s2)	точка в декартовых координатах; точка в полярных координатах; точка пересечения двух прямых линий; точка пересечения линия/окружность; точка пересечения окружность/линия; точка пересечения двух окружностей.
Прямые линии	ln=(om)I J r,(op)I J r ln=(om)X Y,(op)X Y ln=(om)I J r,(op)X Y ln=(om)X Y,(op)I J r ln=(om)I J r,a ln=(om)X Y,a ln=+cm,+cp ln=+cp,pm ln=pm,+cp ln=pm,pq ln=+cm,a ln=pm,a ln=+lm,d	линия, касательная к двум окружностям; линия, проходящая через две точки; линия, касательная к одной окружности и проходящая через одну точку; линия, проходящая через одну точку и касательная к одной окружности; прямая, касательная к одной окружности и образующая угол с абсциссой; прямая, проходящая через точку и образующая угол с осью абсциссы; линия, касательная к двум окружностям; линия, касательная к одной окружности и проходящая через одну точку; линия, проходящая через одну точку и касательная к одной окружности; линия, проходящая через две точки; линия, касательная к одной окружности и образующая угол с осью абсциссы; линия, проходящая через одну точку и образующая угол; линия, параллельная другой на расстоянии **«d».**
Окружности	cn=(om)I J r cn=(om)m a r cn=+lm,+lp,r cn=+lm,+cp,r cn=+cp,+lm,r cn=pm,+lp,r cn=+lp,pm,r cn=+cm,+cp,r cn=pm,+cp,r cn=+cp,pm,r cn=pm,pq,r cn=pm,+lp cn=pm,pa,pr cn=pm,r cn=+cm,+d cn=pm,+cp(,s2)	окружность в декартовых координатах; окружность в полярных координатах; окружность определенного радиуса, касательная к двум прямым линиям; окружность, касательная к одной прямой и к одной окружности определенного радиуса; окружность определенного радиуса, касательная к одной окружности и одной прямой; окружность данного радиуса, проходящая через точку и касательная к одной прямой; окружность данного радиуса, касательная к одной прямой и проходящая через одну точку; окружность данного радиуса, касательная к двум окружностям; окружность данного радиуса, проходящая через точку и касательная к окружности; окружность данного радиуса, касательная к окружности и проходящая через точку; окружность данного радиуса, проходящая через две точки; окружность с центром в одной точке и касательная к одной прямой; окружность, проходящая через три точки; окружность данного радиуса с центром в одной точке; концентрические окружности с данными величинами расстояния; окружность с центром в одной точке и касательная к одной окружности.

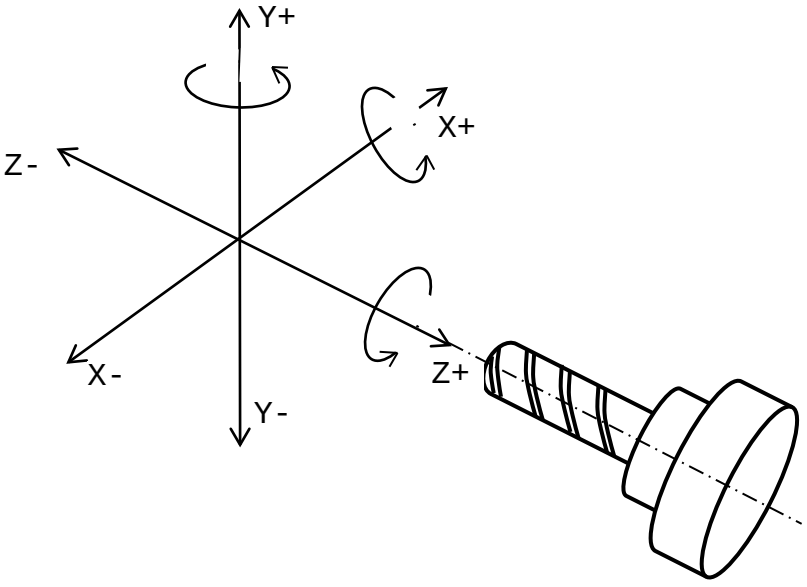
2 ФУНКЦИИ ПРОГРАММИРОВАНИЯ

2.1 Движение осей

Направление движения осей станка предусмотрено стандартом EIA RS 267. Это направление определяется движением инструмента относительно детали независимо от того, что из них будет двигаться (см. рисунок 2.1).



а) вертикальный шпиндель



б) горизонтальный шпиндель

Рисунок 2.1 - Направление движения осей станка

2.2 Подготовительный этап программирования

Подготовка всей необходимой геометрической и технологической информации для осуществления предусмотренного цикла обработки требует от программиста проведения подготовительной работы, которая состоит из следующих операций:

1) определить на чертеже начальную точку осей (ноль детали), относительно которой должны быть измерены все перемещения. Этот выбор должен быть осуществлен в соответствии с фактическими размерами чертежа. Надо иметь в виду, что, если чертеж был выполнен с учетом одной точки, будет возможно выбрать ноль детали, совпадающий с этой точкой. В обратном случае, выбирается точка, которая позволяет осуществить наиболее легкий переход от данного измерения к новому измерению;
2) определить на чертеже детали точки отсчета и точки зажима самой детали;
3) убедиться в том, что все операции, которые необходимо выполнить, находятся в пределах рабочего поля станка;
4) составить список требуемых инструментов в строгой последовательности, необходимой для выполнения программы;
5) определить технологические условия резания (скорость вращения шпинделя и скорость подачи) для каждого инструмента; вышеуказанные данные заносятся программистом в карточку инструмента.

2.3 Ввод программ

Программа, которую необходимо выполнить, должна быть занесена в память системы. Ввод программы в память может осуществляться с клавиатуры, с дискеты или с ПК по последовательному каналу. В последнем случае в ПК должна быть загружена программа **comnc.exe**.

2.4 Информация управляющих программ

2.4.1 Символ

Символ - это число, буква или знак, используемые для выражения информации.

Пример
I,G,%,3,X,LF...

Используемые в УЧПУ символы должны соответствовать символам, которые описаны в таблице 2.1.

Таблица 2.1 - Символы, используемые в УЧПУ

Описание	Символы
Заглавные буквы	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Строчные буквы	a b c d l m o p r s t u v w
Десятичные цифры	от 0 до 9
Математические действия 1	+ -
Математическое действие 2	*
Математическое действие 3	/
Десятичная точка	.
Сепаратор	"
Открытая скобка	(
Закрытая скобка	)
Пояснительный знак	;
Разделительный знак	,
Знак	=
Конец или начало ленты	%(ISO)
Терминатор	L.F. (ISO)
Особые символы	:
Символы-приставки	# (запрос синхронизации) & (аннулирует синхронизацию)

2.4.2 Адрес

Адрес представлен буквой, которая определяет тип инструкции.

Пример
G,Z,X,F

2.4.3 Слово

Слово состоит из адреса, за которым следует цифровое значение.

Пример
G1 X50.5 Y-3.15 F200 T1.1

Все цифровые значения, которые записаны за адресным словом, выражены своей системой измерения. В общем случае нули в начале и в конце могут быть опущены. Если величины имеют десятичную часть, она должна быть записана после десятичной точки.

2.4.4 Кадр

Программа состоит из последовательности кадров, которые позволяют описать цикл обработки. Каждый кадр - это последовательность слов, определяющих операции, которые необходимо выполнить. Каждый кадр должен заканчиваться символом LF (ISO). Максимальная длина кадра - 128 символов. Все кадры, кроме комментирующего, который будет описан далее, могут иметь в начале три дополнительных поля, независимо от класса, к которому принадлежит кадр. А именно:

1) поле подтверждения кадра и выведения его из рабочего состояния (символ 1);
2) поле метки;
3) поле номера кадра.

Они могут присутствовать в кадре по одиночке или одновременно. В случае если они присутствуют одновременно, последовательность расположения одиночных полей должна быть в обязательном порядке следующей: 1), 2), 3). Поле подтверждения кадра и выведения его из рабочего состояния позволяет включить в программу кадры, выполнение которых зависит от параметра системы, названного USB (см. кадры назначения в п.2.5). Если параметр является активным (USB=1), кадр выполняется, в противном случае (USB=0) кадр рассматривается как комментирующий. Формат устанавливается знаком «/» в первой позиции кадра.

Пример
/N100G00X100

Поле метки позволяет дать символическое название кадру, которому оно принадлежит. Метка служит для возможности вызова кадра из различных точек программы при помощи инструкций перехода. Метка - это алфавитно-цифровая последовательность символов, заключенная в знак « » (кавычки), максимальная длина которой - шесть символов Метка должна быть запрограммирована сразу же после поля «/», если оно присутствует.

Пример
"START"
/"END"

Поле номера кадра служит для нумерации одиночных кадров программы. Номер кадра устанавливается символом «N» , за которым следует число, и должен быть запрограммирован в начале каждого кадра, но после символа «/» и метки.

Пример
N125
"INIZIO" N 125
/"FINE" N 125

2.5 Типы кадров

В языке можно определить четыре типа кадров:

1) комментирующие кадры;
2) кадры ISO;
3) кадры назначения;
4) кадры с трехбуквенными кодами.

Комментирующий кадр даёт возможность программисту вводить в программу фразы, описывающие функции, которые он должен выполнить, делая, таким образом, программу более легко читаемой. Такой кадр не выдаёт посылки оператору и не учитывается в стадии выполнения программы. Формат состоит из последовательности алфавитно-цифровых символов, из которых первым элементом в обязательном порядке должен быть символ «;».

Пример

```
;ЭТО - ПРИМЕР
```

Кадры ISO - это кадры, операторы которых предусмотрены стандартом ISO.

Пример

```
G1 X500 Y20 F200
```

Кадры назначения непосредственно из программы пользователя позволяют определить величину нескольких глобальных параметров системы. Впоследствии эти параметры могут быть использованы в других кадрах того же или другого класса. В зависимости от типа переменных кадры назначения могут быть подразделены на три класса:

1) кадры назначения с переменными вычисления; например: E30 = 28.5;
2) кадры назначения с геометрическими переменными; например: p2 = X10 Y25;
3) кадры назначения с глобальными переменными системы; например: UOV=1.5 .

Кадры с трёхбуквенными кодами - это кадры, в которых тип операции, выполнение которой предусмотрено, определён трёхбуквенной командой (кодом), согласованной со стандартом EIA 1177 B.

Пример

```
(URT,45)
```

Оператор G может быть запрограммирован либо неявным способом при помощи параметров E, либо явным. В неявном программировании используемый тип параметра - байт. Все элементы, заключенные в квадратные скобки [ ], должны рассматриваться как необязательные. Все элементы, заключенные в фигурные скобки { }, должны рассматриваться как альтернативные.

2.6 Начало и конец программы

В первом кадре обычно программируется информация о замене инструмента (T...M06). В конце обработки необходимо установить оси в позиции, удобной для демонтажа детали. Затем следует остановить вращение шпинделя и охлаждающий поток и осуществить управление автоматической установкой («СБРОС») программы при помощи функции M30.

```
N1    (DIS,".............")
N2    T1.1 M6 S800
N3    G Z80 X80 M13
...............
...............
N236  G Z250 X50 M5
N237  M30
```

Можно вставить во внутрь программы сообщение, заключённое в кавычки и предназначенное для оператора станка. Это сообщение программируется трёхбуквен-

ным кодом следующим образом: **(DIS, "текст сообщения").** Текст сообщения не должен превышать 32 символа.

2.7 Адресные слова управляющей программы

Подготовительные функции (G), допустимые для программирования в УЧПУ, представлены в таблице 2.2.

Таблица 2.2 - Подготовительные функции G

Код	Группы модальных функций	Действительна только в кадре	Функция	Присутствует при включении
G00	a	нет	Быстрое позиционирование осей	да
G01	a	нет	Линейная интерполяция	нет
G06	a	нет	Сплайновая интерполяция	нет
G02	a	нет	Круговая интерполяция по часовой стрелке	нет
G03	a	нет	Круговая интерполяция против часовой стрелки	нет
G33	a	нет	Нарезание резьбы с постоянным или переменным шагом	нет
G34	a	нет	Нарезание резьбы с постоянным или переменным шагом	нет
G17	b	нет	Функция задания плоскости XY(1-2 оси)	да
G18	b	нет	Функция задания плоскости ZX(3-1 оси)	нет
G19	b	нет	Функция задания плоскости YZ(2-3 оси)	нет
G27	c	нет	Непрерывный режим обработки с автоматическим замедлением скорости на углах	да
G28	c	нет	Непрерывный режим обработки без замедления скорости на углах	нет
G29	c	нет	Перемещение от точки к точке	нет
G21	d	нет	Вход в программу GTL	нет
G20	d	нет	Выход из программы GTL	да
G40	e	нет	Отмена компенсации радиуса инструмента	да
G41	e	нет	Компенсация радиуса инструмента (инструмент слева)	нет
G42	e	нет	Компенсация радиуса инструмента (инструмент справа)	нет
G70	f	нет	Программа в дюймах	нет
G71	f	нет	Программа в мм	да
G80	g	нет	Отмена постоянных циклов	да
G81	g	нет	Постоянный цикл сверления	нет
G82	g	нет	Постоянный цикл расстачивания	нет
G83	g	нет	Цикл глубокого сверления (с разгрузкой стружки)	нет
G84	g	нет	Постоянный цикл нарезания резьбы метчиком	нет
G85	g	нет	Постоянный цикл рассверливания	нет
G86	g	нет	Постоянный цикл развертывания	нет
G89	g	нет	Постоянный цикл развертывания с остановкой	нет
G90	h	нет	Абсолютное программирование	да
G91	h	нет	Программирование в приращениях	нет
G79	k	да	Программирование относительно нуля станка	нет
G04	i	да	Выдержка времени в конце кадра	нет
G09	i	да	Замедление в конце кадра	нет
G72	j	да	Измерение точки с компенсацией радиуса	нет
G73	j	да	Измерение параметров отверстия	нет
G74	j	да	Измерение теоретического смещения от точки без компенсации радиуса	нет
G93	l	нет	Скорость подачи выражена как обратное время выполнения элемента	нет
G94	l	нет	Скорость подачи в мм/мин или дюйм/мин	нет
G95	l	нет	Скорость подачи в мм/об или дюйм/об	да
G96	m	нет	Скорость резания в м/мин или фут/мин	да
G97	m	нет	Скорость вращения шпинделя выражена в об/мин	нет
G35	n	да	Синхронизация начала движения со шпинделем	нет

Примечания

1. Выдержка времени программируется трёхбуквенным кодом TMR:

   TMR=n,

   n - выражено в секундах при G94 и в количестве оборотов шпинделя при G95.
2. Представляется возможным программировать несколько функций G в одном и том же кадре, с учётом их совместимости.

2.7.1 Адресные слова координатных осей A, B, C, U, V, W, X, Y, Z, P, Q, D

Координаты программируются в миллиметрах или дюймах от +/- 0.0001 до +/- 99999.9999.

Любая ось в фазе характеризации системы может быть объявлена осью вращения. Программируемая величина от +/- 0.0001 до +/- 99999.9999 градусов.

2.7.2 Адресное слово R

Определяет в постоянном цикле величину перемещения до точки начала обработки отверстия или величину возврата к этой точке. Программируемая величина от +(-) 0.0001 до +(-) 99999.9999 миллиметров или дюймов. В кадре нарезания резьбы R представляет сдвиг фаз, относительно угловой позиции нуля шпинделя (для многозаходной резьбы).

2.7.3 Адресные слова I J

Выражают координаты центра окружности в круговой интерполяции, соответственно **I** - абсцисса и **J** - ордината. Программируемая величина от +/- 0.0001 до +/- 99999.9999 миллиметров или дюймов. Используемыми символами всегда являются **I** и **J**, независимо от плоскости интерполяции. Символы **I** и **J** используются также в постоянном цикле сверления (G83). Символ **I** в кадре нарезания резьбы определяет изменение шага нарезания резьбы с изменяющимся шагом: **(I+)** - для увеличивающихся шагов, **(I-)** - для уменьшающихся шагов.

2.7.4 Адресное слово K

Определяет коэффициент умножения для обработки глубины отверстия **I** в G83 (постоянный цикл глубокого сверления с разгрузкой стружки). Определяет шаг резьбы, который необходимо выполнить в G33 (нарезание резьбы) и в G84 (нарезание резьбы метчиком). Определяет в винтовой интерполяции шаг винта. Определяет величину корректировки диаметра инструмента. Программируемая величина от +/- 0.0001 до +/- 99999.9999 миллиметров или дюймов.

2.7.5 Функция F

Программируется от 0.01 до 99999.99.

- Функция **G94** - определяет скорость подачи осей в мм/мин (если в G71) или в дюйм/мин (если в G70). Имеется возможность программирования посредством символа **«t»** времени в секундах, необходимого для прохождения участка, определённого в кадре (**F** кадра является отношением между длиной участка и запрограммированным **t**). Функция **t** действительна только в кадре, в котором она запрограммирована.
- Функция **G95** - определяет скорость подачи осей в мм/оборот (G71) или в дюймах/оборот (G70), если это предусмотрено в характеризации.
- Функция **G93** - определяет в минутах обратное время выполнения участка, определённого из отношения: скорость подачи/расстояние. Функция **F** в G93 действительна только в одном кадре.

2.7.6 Функция S

Программируется от 0.01 до 99999.99.

Определяет скорость вращения шпинделя в об/мин, при G97 или скорость резьбы в м/мин при G96 (когда это предусмотрено при характеризации).

2.7.7 Функция T

Определяет инструмент, необходимый для обработки, и номер соответствующей коррекции. Программируемая величина от 1.0 до 9999.9999. Цифры до десятичной точки определяют инструмент, после - номер коррекции. Число коррекций определяется в фазе установки. Коррекция приводится в действие при помощи функции M06. Величины коррекции относятся к длине и диаметру (**K**) инструмента. Корректировка длины инструмента может быть применена к любой оси станка. Выбор зависит от названия оси, к которой присоединена корректировка длины.

Пример
X55, Y20

Корректировка длины приводится в действие без использования других подготовительных функций. Корректировка диаметра инструмента, вызванная одновременно с корректировкой длины, приводится в действие при помощи функций компенсации радиуса инструмента G41/G42 (см. функции программирования G).

2.7.8 Обычно используемые вспомогательные функции M

Описание функций **M** представлено в таблице 2.3.

Таблица 2.3 - Функции M

Функция	Активная функция		Функция или операции, которые её отменяют	Значение
	начало движения	конец движения		
M00		x	ПУСК	Остановка программы
M01		x		Условная остановка программы
M02		x		Конец программы
M03	x		M4-M5-M14-M19	Вращение шпинделя по часовой стрелке
M04	x		M3-M5-M13-M19	Вращение шпинделя против часовой стрелки
M05		x	M13-M4-M13-M14	Остановка вращающегося шпинделя
M06		x		Замена инструмента
M07	x		M9	Включение дополнительного охлаждения
M08	x		M9	Включение основного охлаждения
M09		x	M7-M8	Выключение охлаждения
M10	x		M11	Блокировка осей
M11	x		M11	Разблокировка осей
M12	x			Блокировка вращающихся осей
M13	x		M4-M5-M14-M19	Вращение шпинделя по часовой стрелке и охлаждение
M14	x		M3-M5-M13-M19	Вращение шпинделя против часовой стрелки и охлаждение
M19	x		M3-M4-M5-M13-M14	Остановка вращения шпинделя и угловая ориентация
M30		x		Конец программы и установка на 1-ом кадре
M41	x		M42-M43-M44-M40	Форсирует 1 диапазон вращения шпинделя
M42	x		M41-M43-M44-M40	Форсирует 2 диапазон вращения шпинделя
M43	x		M41-M42-M44-M40	Форсирует 3 диапазон вращения шпинделя
M44	x		M41-M42-M43-M40	Форсирует 3 диапазон вращения шпинделя
M40		x	M41-M42-M43-M44	Отменяет форсированный диапазон вращения шпинделя
M45	x		M41-M42-M43-M44	Автоматическая замена диапазона вращения шпинделя
M60		x		Замена детали

M00 - останавливает выполнение программы после выполнения операций, содержащихся в кадре. Останавливает вращение шпинделя и охлаждающий поток. Сохраняет всю информацию, накопленную в памяти.
M01 - условная остановка программы: если трёхбуквенный код USO=1 занесён с клавиатуры, функция M01 интерпретируется управлением как M00; если трёхбуквенный код USO=0 подтвержден, функция M01 не учитывается.
M02 - определяет конец программы без перемотки ленты на начало.
M03 - вращение шпинделя по часовой стрелке.
M04 - вращение шпинделя против часовой стрелки.
M05 - остановка шпинделя и подачи охлаждения. Осуществляется после выполнения операций, содержащихся в кадре.
M06 - замена инструмента. Останавливает вращение шпинделя, подачу охлаждения и выполнение программы. Подтверждает корректировки, выбранные функцией T. Осуществление становится возможным после выполнения информации, содержащейся в кадре. Не стирает M03, M04, M08, M13, M14.
M07 - подача вспомогательного охлаждения.
M08 - подача основного охлаждения.
M09 - остановка охлаждения. Осуществляется после выполнения операций, содержащихся в кадре.
M10 - блокировка линейных и вращающихся осей. При помощи этой функции осуществляется блокировка осей, не участвующих в процессе обработки.
M11 - отмена M10.
M12 - блокировка вращающихся осей. При помощи этой функции осуществляется блокировка осей, не участвующих в процессе обработки.
M13 - вращение шпинделя по часовой стрелке и подача охлаждения.
M14 - вращение шпинделя против часовой стрелки и подача охлаждения.
M19 - остановка вращения шпинделя с угловой ориентацией осуществима после операций, содержащихся в кадре. Отменяется функциями M03, M04, M13, M14.
M30 - автоматический СБРОС в конце программы. При помощи функции M30 стирается вся информация, находящаяся в динамическом буфере системы. Подтверждаются автоматически: начальная точка 0 и возобновление выбранной программы. Корректировка инструмента в шпинделе не стирается.
M40 - отмена диапазона вращения шпинделя.
M41-M42-M43-M44 - активизирует диапазон вращения шпинделя 1-2-3-4.
M45 - автоматическая смена диапазона вращения шпинделя.
M60 - замена детали.

Примечание - Функции **M**, описанные в этом параграфе, являются чисто указательными.

При помощи программы логики представляется возможным определить эти функции другим образом, добавляя или сокращая их. В каждом кадре можно программировать до четырёх функций **M**.

Все функции **M** стираются при помощи выполнения режима «СБРОС».

2.8 Кадры программирования с функциями G

Эти кадры определены подготовительными функциями G. Оператор G определяется символом «G», за которым следуют две цифры (макс.). Этот оператор должен быть запрограммирован после номера кадра (если таковой имеется) и до какого-либо операнда в кадре. Теоретически существуют 100 операторов типа G (G00 - G99), но только часть из них декодируется системой УЧПУ. В одном кадре можно программировать несколько операторов G в случае, если они конгруэнтны. Таблица 2.4 демонстрирует разделения операторов G на классы с точки зрения конгруэнтности и совместимости внутри одного и того же кадра. В этой таблице величина «1» означает «НЕСОВМЕСТИМОСТЬ». С функциональной точки зрения функции G подразделены на классы и занесены в таблицу 2.5.

Оператор **G** может быть запрограммирован либо неявным способом при помощи параметров **E**, либо явным. Параметр, используемый в неявном программировании, является типа - байт. При описании формата кадра будут встречаться следующие знаки:

1) все элементы, заключённые в квадратные скобки [ ], должны рассматриваться как необязательные;
2) все элементы, заключённые в фигурные скобки { }, должны рассматриваться как альтернативные.

Таблица 2.4 - Конгруэнтность операторов G в кадре

G	00	01	02 03	33	81 86 89	80	72 73 74	21	20	41 42	40	27 28	29	04	09	90 91	79	70 71	17 18 19
G00	1	1	1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1
G01	1	1	1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1
G02	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1
G03	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1
G04	0	0	0	1	1	0	1	0	0	0	0	1	0	1	1	0	0	0	1
G06	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1
G09	0	0	0	1	1	0	1	0	0	0	0	1	0	1	1	0	0	0	1
G17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
G18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
G19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
G20	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	0	1
G21	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	0	1
G27	0	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0	0	0	1
G28	0	0	0	1	1	0	1	0	0	0	0	1	1	1	0	0	0	0	1
G29	0	0	0	0	1	0	1	0	0	0	0	1	1	0	0	0	0	0	1
G33	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1
G34	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1
G35	1	0	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	1
G40	0	0	0	1	1	1	1	0	0	1	1	0	0	0	0	0	1	0	1
G41	0	0	0	1	1	1	1	0	1	1	1	0	0	0	0	0	1	0	1
G42	0	0	0	1	1	1	1	0	1	1	1	0	0	0	0	0	1	0	1
G70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
G71	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
G72	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
G73	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
G74	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
G79	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	1	1	0	1
G80	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0	1
G81	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
G82	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
G83	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
G84	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
G85	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
G86	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
G89	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
G90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
G91	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
G93	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
G94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
G97	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Примечание – «1» означает несовместимость.

Таблица 2.5 – Деление функций G на функциональные классы

Класс	Функции	Описание
a	G00-G01-G02-G03-G06-G33-G34	Определение типа движения
b	G17-G18-G19	Определение плоскости интерполяции
c	G27-G28-G29	Определение динамического режима («от точки к точке» или непрерывный)
d	G21-G20	Открыть и закрыть среду программирования GTL
e	G40-G41-G42	Активизация компенсации радиуса инструмента и её отмена
f	G70-G71	Программирование в альтернативной системе измерения
g	G81..G86-G89-G80	Постоянные циклы обработки отверстия
h	G90-G91	Программирование абсолютное/в приращениях
i	G79	Программирование относительно нуля станка
j	G04-G09	Свойства динамического типа
k	G72-G73-G74	Циклы измерения
l	G93-G94-G95	Скорость подачи
m	G96-G97	Скорость вращения шпинделя

2.8.1 Тип движения

Тип движения определяется функциями:

G00 - быстрое позиционирование осей;
G01 - линейная интерполяция;
G02 - интерполяция круговая по часовой стрелке;
G03 - интерполяция круговая против часовой стрелки;
G06 - сплайновая интерполяция;
G33 - нарезание резьбы с постоянным или переменным шагом;
G34 - нарезание резьбы с постоянным или переменным шагом.

2.8.1.1 Быстрое позиционирование осей (G00)

Быстрое позиционирование осей (G00) определяет линейный тип движения, скоординированный по всем осям, запрограммированным в кадре с быстрым ходом.
Формат:

G00 [ДРУГИЕ G] [ОСИ] [ОПЕРАНДЫ КОРРЕКТИРОВКИ] [СКОРОСТЬ ПОДАЧИ] [ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ] ,

где:

[ДРУГИЕ G] - все другие функции **G**, совместимые с **G00** (см. таблицу 2.4);
[ОСИ] - представлены символом оси, за которым следует числовое значение в явной или неявной форме (параметр E). Могут присутствовать 8 осей (максимально), они не должны быть заменимыми между собой. Для неявного определения осей необходимо вначале определить точку согласно текущей абсциссе и ординате;
[ОПЕРАНДЫ КОРРЕКТИРОВКИ] - коэффициенты коррекций на плоскости (u, v, w);
[СКОРСТЬ ПОДАЧИ] - рабочая подача для скоординированных перемещений. Она запоминается, но не определяет движение осей, определенных в кадре с функцией G00. Скорость подачи в кадре с функцией G00, для программируемых осей, определяется на базе скоростей быстрого хода. Скорости быстрого хода определяются в файлах характеризации УЧПУ;
[ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ] - вспомогательные функции M, S и T. В одном кадре можно программировать: до 4 - функций M, по 1 - функции S и T.

2.8.1.2 Линейная интерполяция (G01)

Линейная интерполяция (G01) определяет линейное одновременное движение, скоординированное по всем осям, которые запрограммированы в кадре, с заданной скоростью обработки.
Формат:

G01 [ДРУГИЕ G] [ОСИ] [ОПЕРАНД КОРРЕКТИРОВКИ] [СКОРОСТЬ ПОДАЧИ] [ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ] ,

где:

[СКОРОСТЬ ПОДАЧИ] - выражает рабочую скорость (**F**), с которой выполняется движение. В случае отсутствия используется ранее запрограммированная скорость. Это означает, что в предшествующих кадрах должна быть запрограммирована скорость. В противном случае подаётся сигнал ошибки.

Описание остальных полей изложено в предыдущем пункте.
Пример линейной интерполяции приведён на рисунке 2.2.

```
N79 X10 Y10
N80 G01 X90 Y40 F200
```

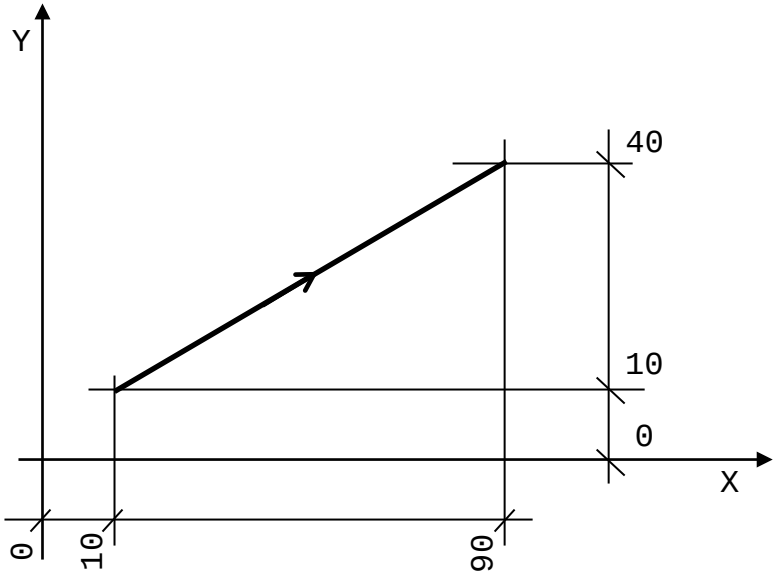


Рисунок 2.2

2.8.1.3 Сплайновая интерполяция

Сплайновая интерполяция применяется для объединения последовательности отдельных точек в гладкий непрерывный контур.

В ПО сейчас реализован «С»-сплайн. «С»-сплайн обеспечивает гладкий контур с точным прохождением через все точки сплайна, с непрерывной кривизной и возможностью задания условий на его краях.

Точки сплайна должны быть определены между операторами **BSP** – начало сплайна и **ESP** – конец сплайна. В первый кадр после оператора **BSP** обязательно записывается функция **G00** или **G01,** координата начала сплайна и, если необходима компенсация радиуса инструмента, то функция **G41** или **G42**. Далее при определении координат точек сплайна, принадлежащих профилю (до кадра предшествующего оператору **ESP**), записывать функции **G00** или **G01** нельзя. Во второй кадр после оператора **BSP** записывается функция **G06** и координата первой точки профиля. В кадр, предшествующий оператору **ESP,** обязательно записывается функция **G00** или **G01,** координата выхода с профиля и функция отмены компенсации радиуса инструмента **G40.**

Формат:

```
(BSP, Имя оси абсциссы, Имя оси ординаты, n)
G01/G00[Другие G][Оси][Подача][Вспомогательные функции]
G06    [Другие G][Оси][Подача][Вспомогательные функции]
       [Другие G][Оси][Подача][Вспомогательные функции]
       [Другие G][Оси][Подача][Вспомогательные функции]
.............
G01/G00[Другие G][Оси][Подача][Вспомогательные функции]
(ESP)
```

где:

BSP - начало определения сплайна, где:

Имя оси абсциссы, Имя оси ординаты - имена осей, определяющих уже установленную плоскость интерполяции;

N - параметр, задающий различные условия на краях сплайна **(n=0÷3),** где:

n=0:

- кривизна в первой точке сплайна равна «0»;
- кривизна в последней точке сплайна равна «0».

n=1:

- движение в первой точке сплайна направлено по касательной;

- кривизна в последней точке равна «0».

n=2:

- кривизна в первой точке сплайна равна «0»;
- движение в последней точке сплайна направлено по касательной.

n=3:

- движение в первой точке сплайна направлено по касательной;
- движение в последней точке сплайна направлено по касательной.

Примечания

1. Направление касательной на входе сплайна определяется в кадре с перемещением, содержащим функции **G01** или **G00** и записанным между кадром с оператором **BSP** и кадром с функцией **G06**.
2. Направление касательной на выходе сплайна определяется в кадре с перемещением, содержащим функции **G01** или **G00** и записанным сразу перед кадром с оператором **ESP**.

G06 - определяет тип движения по сплайновой интерполяции. После функции **G06** программируются координаты точек сплайнового профиля. Функция **G06** отменяется функцией **G01** или **G00**. Программирование других функций, задающих тип движения, внутри сплайна запрещено (**Сообщение_4 77 «Неконгруэнтный профиль»**).

ESP - конец определения сплайна.

На рисунке 2.3 а) представлен «С» – сплайн, когда движение в первой и/или последней точке сплайна P0 направлено по касательной. Направление касательной определяет направление отрезка (P;P0), если точка P0 - первая точка сплайна или направление отрезка (P0;P), если точка P0 - последняя точка сплайна. Точки P0, P1, P2, P3, ,P4 – точки сплайна. Точка P не является точкой сплайна.

На рисунке 2.3 б) представлен «С» – сплайн, когда кривизна в первой и/или последней точке сплайна P0 равна «0» и не зависит от направления отрезков (P;P0) или (P0;P).

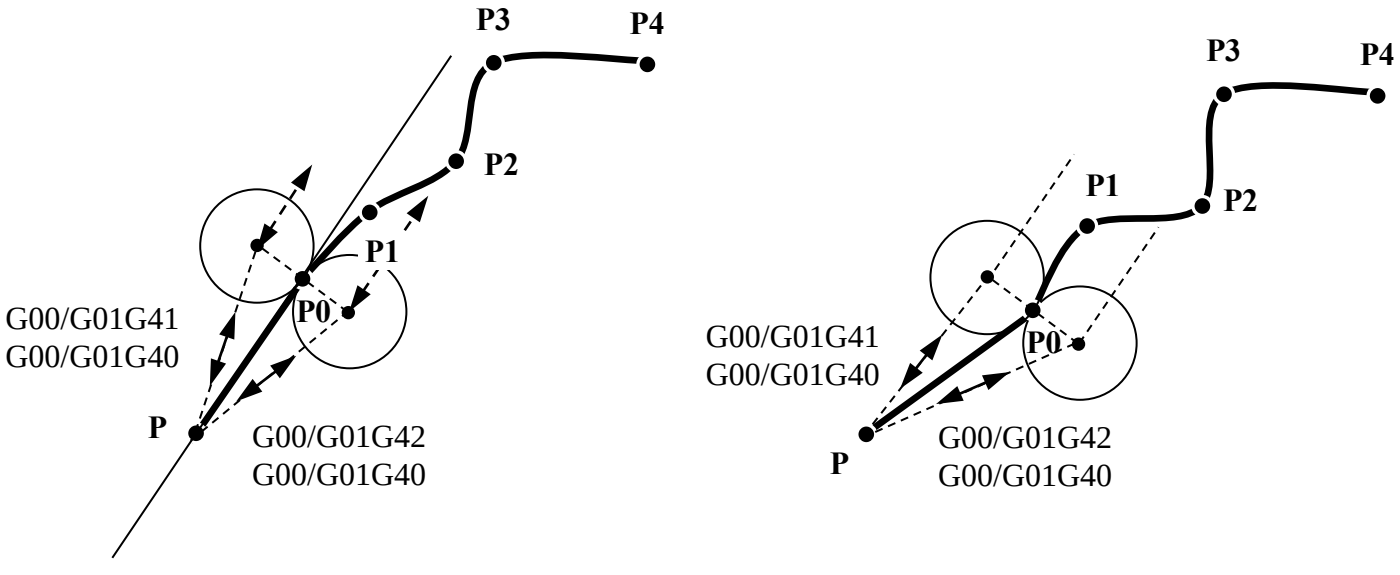


Рисунок 2.3

2.8.1.4 Круговая интерполяция (G02-G03)

Круговая интерполяция (G02-G03) определяет круговое движение по часовой стрелке (G02) или против часовой стрелки (G03). Это движение является скоординированным и одновременным по всем осям, запрограммированным в кадре с заданной скоростью обработки.

Формат:

{G02,G03} [ДРУГИЕ G] [ОСИ] I J [СКОРОСТЬ ПОДАЧИ] [ОПЕРАНДЫ КОРРЕКТИРОВКИ] [ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ] ,

где:

[G] - смешанные операторы и вспомогательные функции; имеют те же значения, что и в предыдущих случаях;

[СКОРОСТЬ ПОДАЧИ] - скорость подачи;

[ОСИ] - представлены символом оси и цифровым значением в явной или неявной форме (параметр E). Если ни одна ось не запрограммирована, то выполняемым движением будет полное круговое движение в плоскости интерполяции. Оси могут быть определены неявным образом посредством геометрического элемента - точки. Если координата прибытия равна координате отправления, она может быть опущена;

I и J - являются адресными словами, выражающими координаты центра окружности, цифровая часть которых может быть выражена в явной или неявной форме (параметр E). Используемыми символами всегда являются I и J независимо от плоскости интерполяции и всегда присутствуют.

Пример круговой интерполяции приведён на рисунке 2.4.

```
N10 G1 X-20 Y60 F200
N20 G3 X-40 Y80 I-40 J60
N30 G1 X-45
N40 G2 X-55 Y90 I-45J90
N50 G1 Y...
```

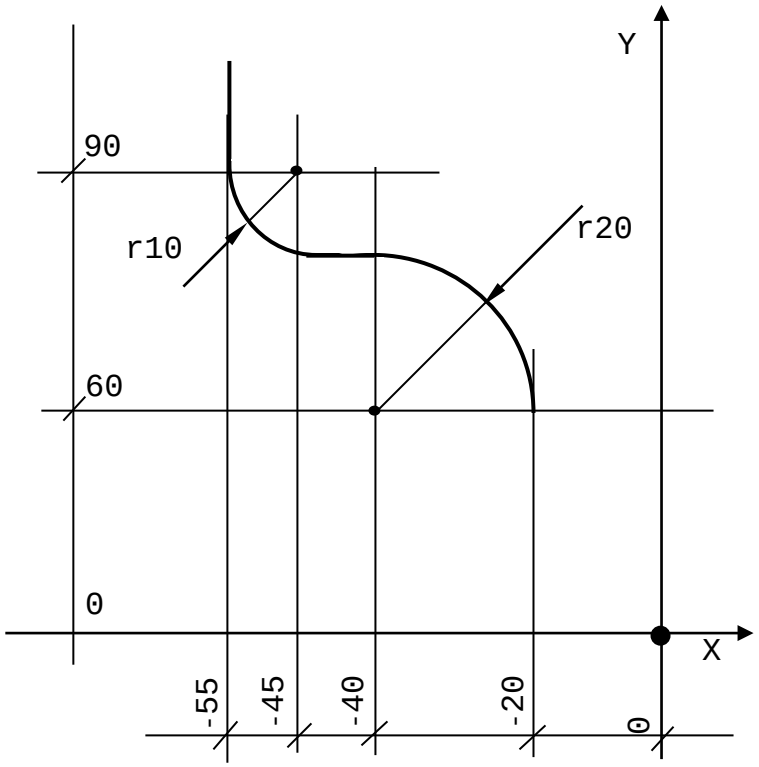


Рисунок 2.4

Максимальная программируемая дуга - 360 градусов.

Координаты начальной точки, запрограммированные в предшествующем кадре, конечной точки и центра окружности должны быть вычислены таким образом, что бы разница между начальным и конечным радиусом не превышала бы 0,01 мм. Если разница превышает это значение, воспроизводится запись: «Профиль не конгруэнтен»,- и окружность не выполняется.

Круговая интерполяция может быть также запрограммирована в приращениях, т.е. с координатами конечной точки и точки центра окружности относительно начальной точки, запрограммированной в предшествующем кадре.

Пример соответствует рисунку 2.4:

```
N10 G1 X-20 Y60 F200
N20 G3 G91 X-20 Y20 I-20 J0
N30 G1 X-5
N40 G2 X-10 Y10 I0J10
N50 G1 Y...
```

Направление кругового движения (по часовой или против часовой стрелки) определяется, глядя на плоскость интерполяции со стороны положительной полуоси, перпендикулярной к плоскости, в соответствии с рисунком 2.5.

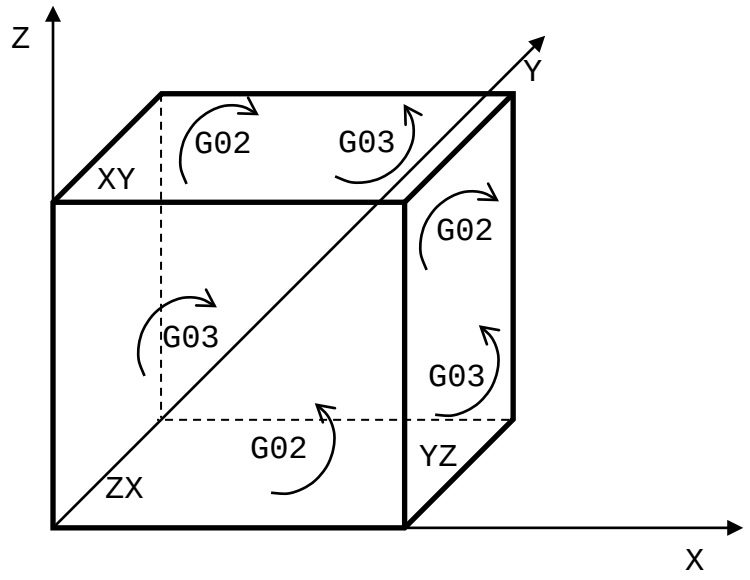


Рисунок 2.5

Программирование дуги менее 360 градусов через задание координат конечной точки и радиуса.

Формат:

{G02,G03} [ДРУГИЕ G] [ОСИ] R+ [СКОРОСТЬ ПОДАЧИ] [ОПЕРАНДЫ КОРРЕКТИРОВКИ] [ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ] ,

где:

[G] - смешанные операторы и вспомогательные функции; имеют те же значения, что и в предыдущих случаях;

[СКОРОСТЬ ПОДАЧИ] - скорость подачи;

[ОСИ] - представлены символом оси и цифровым значением в явной или неявной форме (параметр E); оси могут быть определены неявным образом посредством геометрического элемента – точки;

R - адресное слово, выражающее радиус дуги окружности, цифровая часть которой может быть выражена в явной или неявной форме (параметр E); знак «+» или «-» перед адресным словом R выбирает одно из двух возможных решений:

- «+» - для дуги до 179.999^0;
- «-» - для дуги от 180^0 до 359.999^0.

Пример программирования дуги менее 360 градусов через задание координат конечной точки и радиуса приведён на рисунке 2.6.

```
1     G02 X20 Y20 R+20 F100

2     G02 X20 Y20 R-20 F100
```

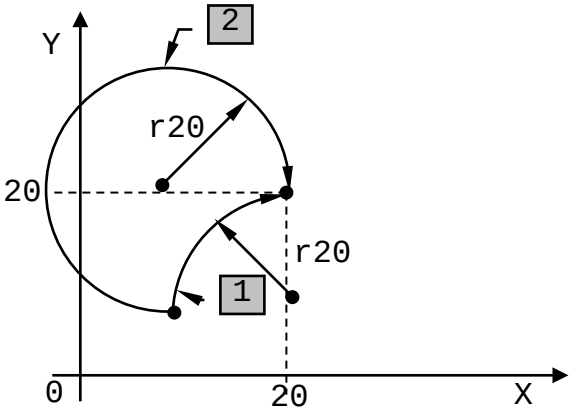


Рисунок 2.6

2.8.1.5 Плоскость интерполяции

Плоскость интерполяции устанавливается при характеризации системы и может быть переопределена при помощи функций **G17-G18-G19** или же посредством определения плоскости, образованной парой осей, установленных трёхбуквенным кодом DPI (G17 в любом случае присутствует при включении).

Формат:

{G17}
{G18}
{G19} ,

где:

G17 - определяет плоскость круговой интерполяции и компенсации радиуса инструмента, образованную осями 1-2 (XY);
G18 - определяет плоскость круговой интерполяции и компенсации радиуса инструмента, образованную осями 1-3 (XZ);
G19 - определяет плоскость круговой интерполяции и компенсации радиуса инструмента, образованную осями 2-3 (YZ).

Примечание - Направление круговой интерполяции (по часовой стрелке или против) определяется при взгляде на плоскость с позитивной стороны перпендикулярной к ней полуоси.

2.8.1.6 Винтовая интерполяция

Для получения перемещения по винтовой линии необходимо запрограммировать в одном и том же кадре круговую интерполяцию на плоскости интерполяции и линейное перемещение, перпендикулярное к этой плоскости.

Формат:

[G02] [другие коды G] I J K [F] ,
[G03] [вспомогательные функции] .

Пример
G02(или)G03 X..Y..Z..I..J..K..F.. ,

где:

G02(или)G03 X..Y..I..J..- параметры окружности;
Z - глубина винта;
K - шаг винта (**K** может быть пропущен, если глубина винта меньше шага винта в соответствии с рисунком 2.7).

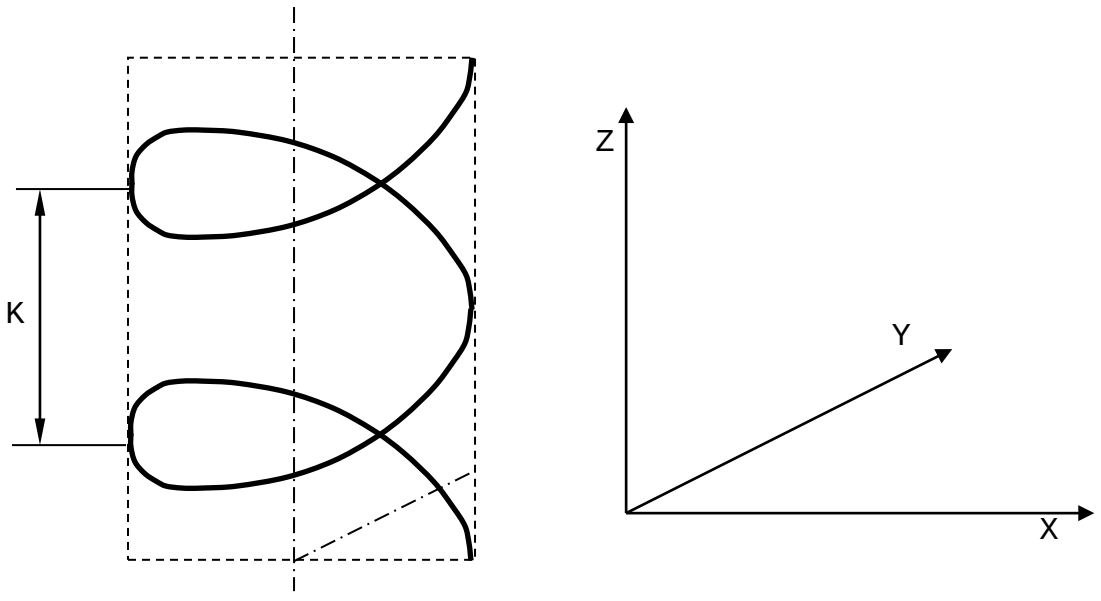


Рисунок 2.7

Длина винта, выраженная через шаг винта, равна **nK**, где **n** – целое или десятичное число.

Если **n** – целое число, необходимо запрограммировать полный круг (360 градусов).

Если **n** – десятичное число, то необходимо запрограммировать дугу, пропорциональную **n**. Например, если длина винта равна 2.7К, необходимо запрограммировать следующую дугу: 360*0,7=252 градусов.

Как и при круговой интерполяции, плоскость винтовой интерполяции можно определить при помощи функций **G17**, **G18**, **G19** или с помощью трёхбуквенного кода DPI.

2.8.1.7 Нарезание резьбы с постоянным или переменным шагом (G33)

Нарезание резьбы с постоянным или переменным шагом (G33) определяет цикл цилиндрического или конического нарезания резьбы с постоянным или переменным шагом. Это движение координируется с вращением шпинделя. Запрограммированные в кадре параметры определяют тип резьбы, которую следует осуществить.

Формат:

G33 [ОСИ] К [I] [R] ,

где:

[ОСИ] - представлены символом оси и цифровым значением в явной или неявной форме (параметр **E**);

К - представляет шаг резьбы; в случае переменного шага, представляет начальный шаг; должен присутствовать всегда;

[I] - представляет изменение шага; для нарезания резьбы с возрастающим шагом **I** должна быть положительной, для нарезания резьбы с уменьшающимся шагом должна быть отрицательной;

[R] - представляет отклонение по отношению к угловой позиции нуля шпинделя (в градусах); используется при многозаходной резьбе для того, чтобы не сдвинуть начальную точку.

Во время нарезания резьбы выведены из состояния работы команда «СТОП» и коррекции подачи и скорости вращения шпинделя.

Функция G33 программируется только с датчиком в шпинделе.

Примеры нарезания резьбы с постоянным шагом приведены на рисунке 2.8:

- цилиндрическое нарезание резьбы;
- коническое нарезание резьбы;
- цилиндрическо-коническое нарезание резьбы.

Примечания

1. Ось U – диаметральная.
2. Все параметры могут быть выражены цифровым значением в явной или неявной форме.

Примеры нарезания резьбы с переменным шагом приведены на рисунке 2.9:

- цилиндрическое нарезание резьбы с возрастающим шагом;
- коническое нарезание резьбы с возрастающим шагом;
- цилиндрическое нарезание резьбы с уменьшающимся шагом.

Во время нарезания резьбы с уменьшающимся шагом начальный шаг, изменения шага и длина нарезания резьбы должны быть такими, чтобы шаг не становился равным нулю до достижения конечного размера. Для проверки применяется формула:

$$I <= \frac{K2}{(ZK - ZN)}, \quad (2.1)$$

где:

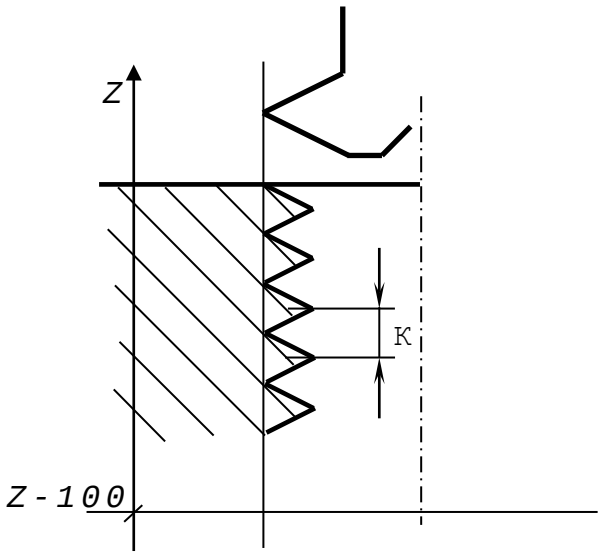
I - максимальное изменение шага;

К - начальный шаг;

ZK - координата конечной точки;

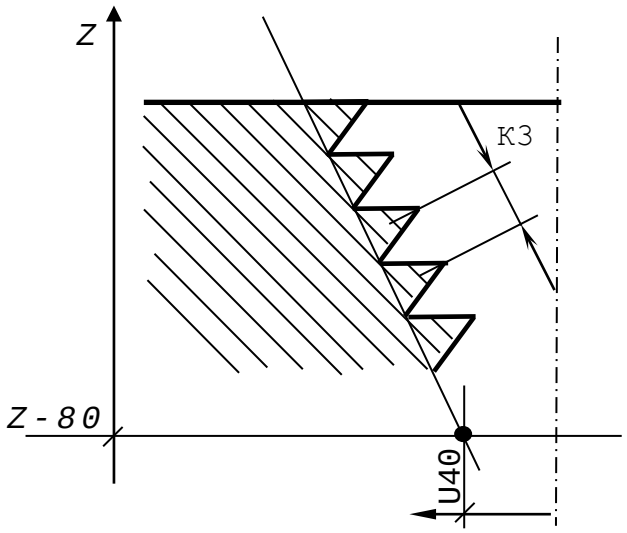
ZN - координата начальной точки;

(ZK-ZN) - длина нарезания резьбы.



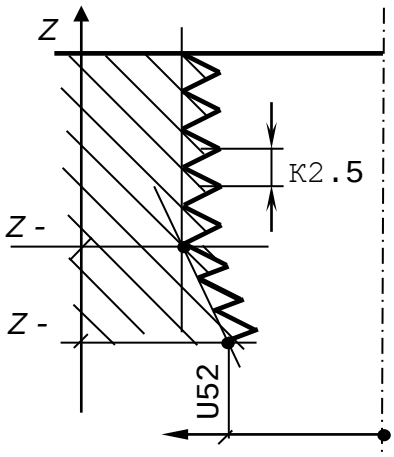
N22 G33 Z-100 K2

а) цилиндрическая резьба



N22 G33 X-80 U40 K3

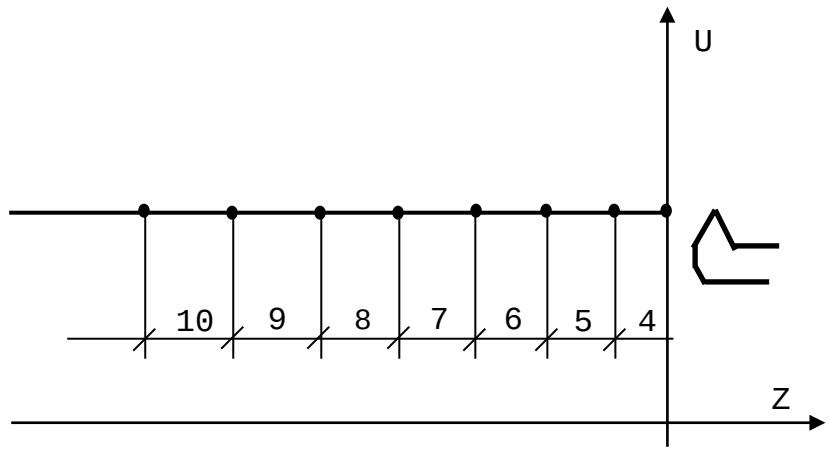
б) коническая резьба



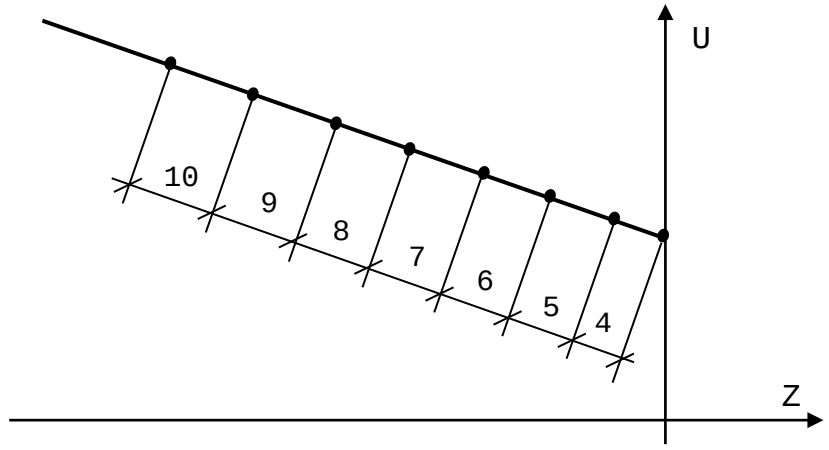
N35 G33 Z-95 K2.5
N36 Z-100 U52 K2.5

в) цилиндрическо-коническая резьба

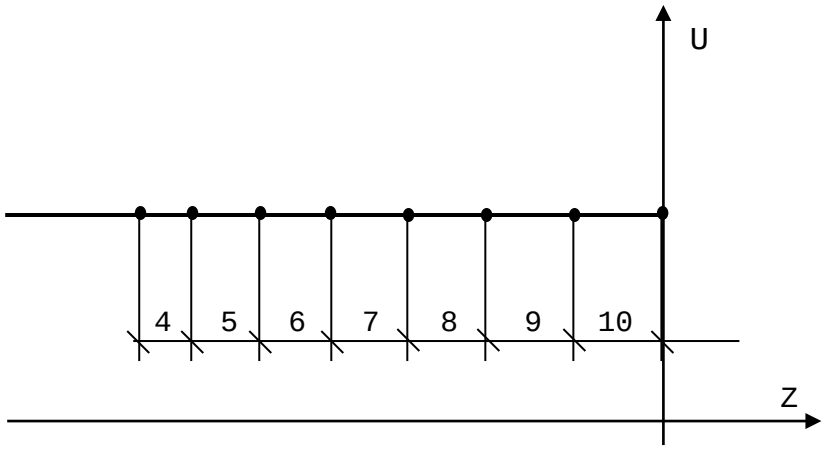
Рисунок 2.8 – Примеры нарезания резьбы с постоянным шагом



а) цилиндрическая резьба с увеличивающимся шагом



б) коническая резьба с увеличивающимся шагом



в) цилиндрическая резьба с уменьшающимся шагом

Рисунок 2.9 - Примеры нарезания резьбы с переменным шагом

Пример нарезания резьбы с 3 заходами:

```
.............
N37 G33 Z3 K6          первый заход
.............
N41 G33 Z3 K6 R120     второй заход
.............
N45 G33 Z3 K6 R240     третий заход
.............
```

Функция R дает команду системе для размещения осей в угловой позиции, которая меняется в зависимости от запрограммированной величины R. Таким образом, представляется возможным программировать одну начальную точку для различной нарезки в отличие от других систем, в которых для осуществления многозаходной резьбы необходимо сместить начальную точку каждой нарезки на величину, равную шагу, разделённому на количество заходов.

2.8.2 Программирование угловых перемещений

Во время характеризации системы любая ось может быть определена как ось вращения. Программирование так называемого «непрерывного поворотного стола», движение которого является одновременным и скоординированным с движением других осей, запрограммированных в том же кадре, очень простое. Необходимо учитывать следующие моменты:

- программирование должно быть выполнено в градусах и десятичных дробях градуса от $\pm$0.00001 до $\pm$99999.9999 градусов, начиная с предварительно установленной начальной точки;
- перемещение может быть осуществлено на быстром ходу с функцией **G00** или с рабочей подачей при функции **G01**, программируя скорость вращения в град/мин (с 2 десятичными макс.) посредством функции **F**.

Например, при программировании **F75.5** ось будет вращаться со скоростью 75.5 град/мин. Если необходимо выполнить фрезерование вдоль окружности с использованием поворотного стола, для вычисления скорости угловой подачи, которую надо запрограммировать, следует использовать следующую формулу:

$$F = \frac{360}{\pi} * \frac{A}{D} = 144.64 * \frac{A}{D}, \qquad (2.2)$$

где:

F - угловая скорость, град/мин;
A - линейная скорость вдоль окружности, мм/мин;
D - диаметр, на котором выполняется фрезерование, (мм).

Когда вместе с вращательными осями движутся также и линейные оси, для вычисления скорости подачи, которую надо запрограммировать, нужно использовать следующую формулу:

при **G94**

$$F = A * \frac{\sqrt{X^2+Y^2+Z^2+B^2+C^2}}{L}; \qquad (2.3)$$

при **G93**

$$F = \frac{A}{L}, \qquad (2.4)$$

где:

F - скорость подачи, которую надо запрограммировать;
A - скорость подачи, требуемая для обработки детали, мм/мин;
X Y Z B C - фактическое перемещение, выполненное каждой осью (мм – для линейных осей, градусы – для вращательных);
L - длина результирующей траектории, мм.

Траекторией будет:

- дуга окружности в случае, когда движется только ось вращения;
- дуга Архимеда, цилиндрическая спираль или же сложные кривые, если ось вращения движется вместе с одной или несколькими линейными осями.

2.8.3 Управление вращением индексного поворотного стола

Управление вращением индексного поворотного стола осуществляется программированием названия оси, определённой на стадии характеризации системы, за которым следует число индексов, на которое необходимо поворачивать стол. Программирование может быть выполнено в абсолютной системе (G90) или по приращениям (G91).

Пример

```
.....
N24 P10
.....
N41 G91 P2
.....
G90 P0
```

Управление вращением индексного поворотного стола выполняется в начале движения. Таким образом, при программировании кадра N..G..X..Y..P, сначала будет осуществлено вращение, а потом после углового размещения, линейное перемещение X..Y..

При каждом включении УЧПУ стол следует установить в нулевое положение, с клавиатуры для этого вводится команда: «B0».

Символы, используемые для названия осей, аналогичны символам, используемым для координатных осей станка.

Система управляет одной индексной осью.

Число индексов программируется явным образом или посредством параметра E и не должно превышать значения 99999.

2.8.4 Оси вращения с увеличенным диапазоном поворота

Оси вращения могут иметь увеличенный диапазон поворота (более 360^0). В этом случае при характеризации они объявляются как оси с увеличенным диапазоном поворота. При этом позиция, которая запрограммирована или введена с пульта, при выводе на экран остается всегда в диапазоне от 0 до 359.9999^0. программирование можно осуществлять как в абсолютных размерах (G90), так и в приращениях (G91). В случае программирования в абсолютных размерах возможный диапазон задания перемещений от ± 0 до $\pm$ 359.9999^0. При этом запрограммированный угол соответствует желаемой позиции, и знак указывает направление вращения. Примеры приведены на рисунках 2.10-2.12.

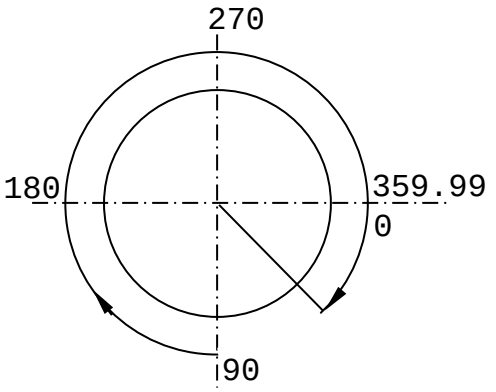


Рисунок 2.10

Пример 1 соответствует рисунку 2.10.

Допустим, что ось вращения **B** должна быть позиционирована на отметку 90 градусов, тогда кадр имеет вид: **B45**. В этом случае ось поворачивается на 315^0, позиционируется на отметке 45^0, и на экране появится: **B45**.

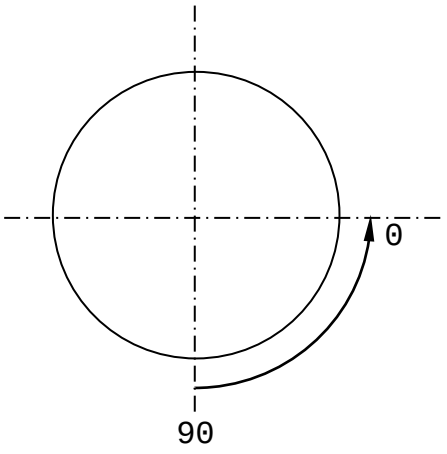


Рисунок 2.11

Пример 2 соответствует рисунку 2.11.

Допустим, что ось вращения позиционирована на отметке 90 градусов, и программируется кадр: **B0.**

В результате ось поворачивается на 90 градусов и становится на отметку 0 градусов, а на экране появится: **B0.**

При программировании в приращениях G91 диапазон программирования представляет величину +0.0001-99999.9999, направление вращения определяется знаком приращения.

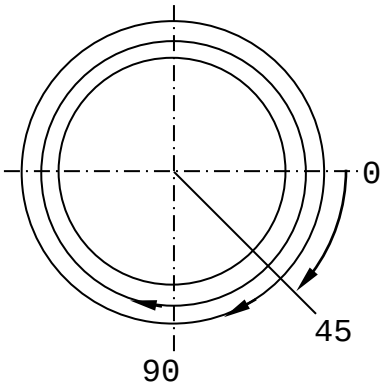


Рисунок 2.12

Пример 3 соответствует рисунку 2.12

Если ось вращения **B** находится на отметке 0 градусов, и программируется кадр: **G91 B765,** ось **B** поворачивается на два оборота и еще на 45 градусов, а на экране появится размер: **B45.**

2.8.5 Определение режима динамики (G27-G28-G29)

Функции определения режима динамики определяют скорость выхода из элементов профиля, т.е. режим движения. К этому классу принадлежат три функции: G27, G28, G29.

Формат:

```
{G27}[ДРУГИЕ G] [ОПЕРАНДЫ]    ,
{G28} ,
{G29} ,
```

где:

[ОПЕРАНДЫ] - указывает все возможные классы, определённые для операндов с функциями «G»;

G27 - обеспечивает непрерывное движение с автоматическим уменьшением скорости на углах; это значит, что скорость выхода из элементов профиля вычисляется автоматически в соответствии с геометрической формой профиля и установленными значениями переменных ERF и MCD;

G28 - обеспечивает непрерывное движение без автоматического уменьшения скорости на углах; это означает, что скорость выхода из элементов профиля равна запрограммированной скорости;

G29 - обеспечивает движение в режиме «от точки к точке», т.е. скорость выхода из элементов профиля установлена равной «0».

Тип позиционирования, который осуществляется со скоростью обработки G1, G2, G3 установлен функциями G27, G28, G29, в то время как быстрое позиционирование G00 осуществляется всегда «от точки к точке», т.е. со сведением скорости к нулю и точным позиционированием, независимо от состояния, в котором находится система (G27,G28,G29). Во время включения и после включения каждого «СБРОСА» функция G27-G0 автоматически приводится в действие.

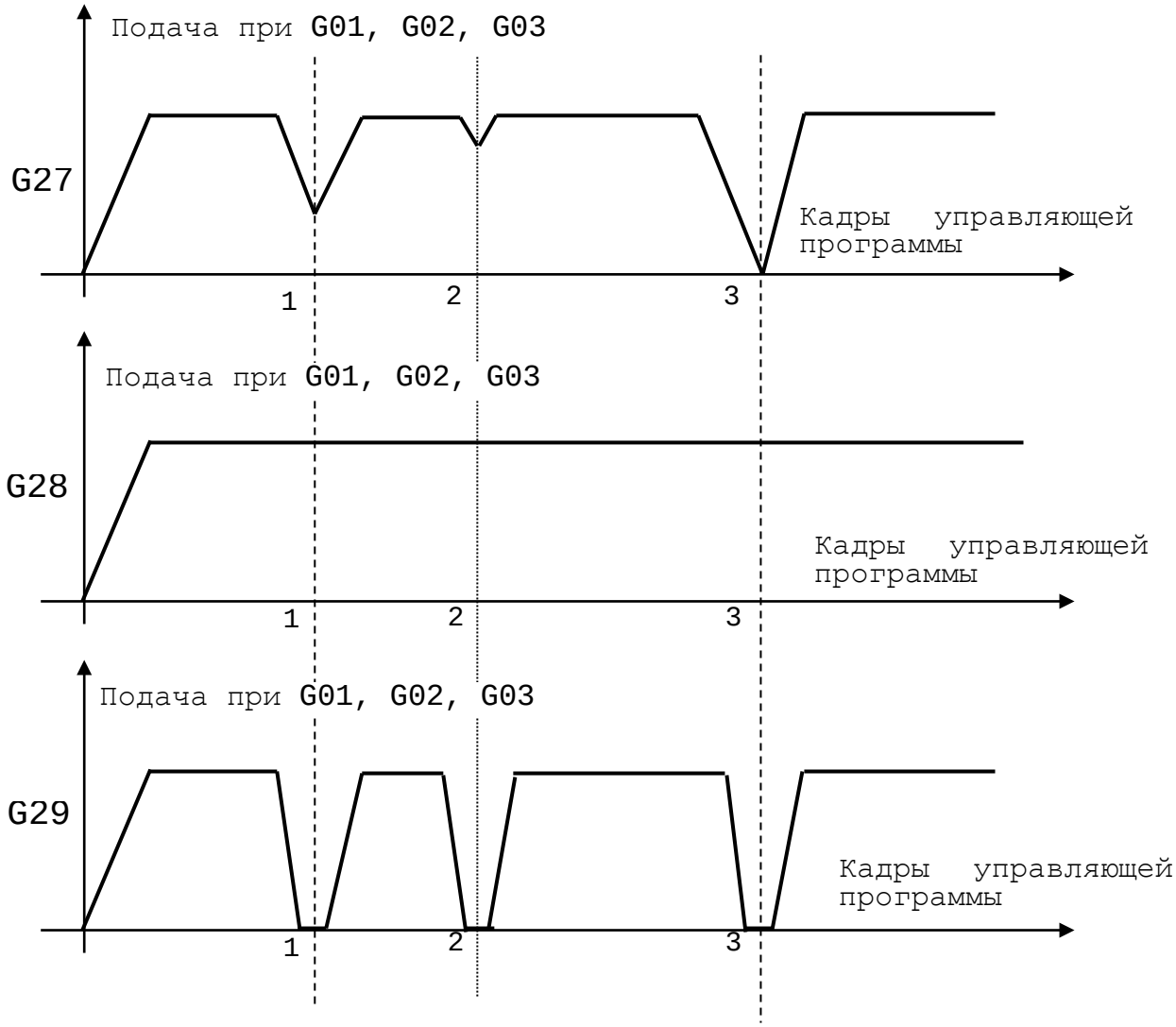


Рисунок 2.13 - Графическое изображение режимов динамики движения

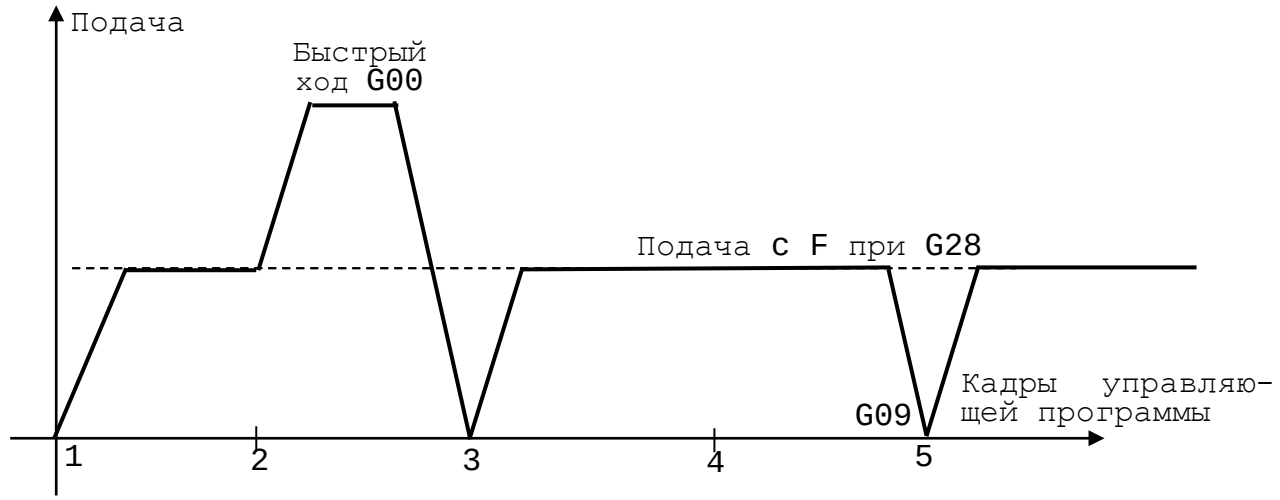


Рисунок 2.14 - Замедление в конце кадра внутри непрерывной обработки G28

Во время непрерывного движения (G27-G28), система запоминает профиль, который должен быть реализован, поэтому элементы профиля выполняются как один кадр. По этой причине во время прохождения профиля с G27-G28 применение вспомогательных функций **M, S** и **T** недопустимо. Непрерывное функционирование временно прекращено движением по G00, которое является частью профиля. Если необходимо запрограммировать вспомогательные функции **M, S, T,** программирование осуществляется в кадре, следующем после G00. Примеры режимов динамики при непрерывном режиме и в режиме «от точки к точке» приведены на рисунке 2.13.

Внутри непрерывной обработки G28 можно программировать замедление в конце кадра при помощи G9, как показано на рисунке 2.14.

Примеры

1) Непрерывная обработка:

```
.............
N9 (DIS,”ФРЕЗА D=16”)
N10 S800 T4.4 M6
N11 G X-235 Y230 M13
N12 Z-10
N13 G1 X75 F500              начало непрерывной обработки G27
N14 Y
N15 G3 X-70.477 Y25.65 I J
N16 G1 X-187 Y-295
N17 G Z5                     временная отмена непрерывной
N18 M5                       обработки (M) для остановки
N19 (DIS,”24”)               шпинделя, замена инструмента,
N20 T5.5 M6 S1200            функции S
N21 G.. Y.. M13
N22 Z-.
N23 G1 X.. Y..
.............
```

Примечание - Если функция G29 была запрограммирована в кадре N17, непрерывная обработка отменяется, и последующие движения в G2, G3 будут осуществлены «от точки к точке».

2) Обработка в режиме «от точки к точке»:

```
.............
N9 (DIS,”ФРЕЗА D=16”)
N10 S800 T4.4 M6
N11 G29 G X-235 Y230 M13     Начало обработки «от точки к точке»
N12 Z-10
N13 G1 X75 F500 M5           остановка шпинделя
N14 Y S1200 M13              изменение S, вращения шпинделя
N15 G3 X-70.477 Y25.65 I J
N16 TMR=2
N17 G1 G4 X-187 Y-295        останов в конце кадра
N18 G Z5
N19 M5
N20 (DIS,”24”)
N21 T5.5 M6 S1200
N22 G.. Y.. M13
N23 Z-..
N24 G1 X.. Y..
.............
```

Примечание - Находясь в режиме «от точки к точке», установленным функцией G29 в кадре N11, можно программировать функции **M** и **S** в профиле (кадры N13, N14).

2.8.6 Геометрическое определение профиля на базе языка GTL (G21-G20)

Функции геометрического определения профиля определяют профиль, запрограммированный с использованием языка GTL. К этому классу принадлежат две функции:

G21 - устанавливает начало геометрического профиля на базе GTL;
G20 - устанавливает конец геометрического профиля на базе GTL.

Формат:

{G20} {pn} ,
{G21} [ДРУГИЕ G] {ln} [s2] [ОСИ] [СКОРОСТЬ ПОДАЧИ] [ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ] {cn} ,

где:

pn,ln,cn - обозначают точку, прямую линию и окружность индекса n, определённую ранее; если запрограммировано pn, это означает, что профиль открыт; pn не может быть запрограммировано внутри профиля;

s2 - обозначает вторую точку пересечения между двумя элементами прямая линия - окружность (s1 не программируется); данные оси могут быть только осями, не принадлежащими плоскости интерполяции; другие поля имеют значения, аналогичные значениям, описанным для функции G1;

[оси] могут быть только осями, не принадлежащими плоскости интерполяции.

Другие поля имеют значения, аналогичные значениям, описанным для функции **G1**.

Примечание - Примеры и спецификацию см. в п.2.11 («Геометрическое программирование высшего уровня»).

2.8.7 Компенсация радиуса инструмента (G41-G42-G40)

Для включения/выключения компенсации радиуса инструмента программируются следующие функции:

- G41 - включение компенсации, инструмент слева от детали;
- G42 - включение компенсации, инструмент справа от детали;
- G40 - отмена компенсации.

Формат:

G41 ,
G42 [другие коды G] [операнды] ,
G40 .

До программирования компенсации радиуса инструмента необходимо определить плоскость интерполяции. Как известно, плоскость интерполяции может быть определена при помощи G17 (устанавливается при включении устройства), G18, G19 или при помощи трёхбуквенного кода DPI. Вектор компенсации радиуса инструмента, оставаясь перпендикулярным к обрабатываемому профилю вдоль всего профиля, обеспечивает точное позиционирование инструмента в точках пересечения прямых и окружностей, смещенных относительно профиля (см. рисунок 2.15). Инструмент должен позиционироваться к стартовой точке на профиле при помощи линейной интерполяции. Однако, первый элемент (кадр) профиля может быть как линейным (см. рисунок 2.16), так и круговым (см. рисунок 2.17).

Для отмены компенсации радиуса инструмента необходимо запрограммировать функцию G40.

Действие функций G41, G42 прекращается в первом же кадре движения после кадра с функцией с G40. На рисунке 2.18 изображено использование функции G40 в случае, когда последним элементом профиля является линейный кадр. На рисунке 2.19 последним кадром профиля является кадр круговой интерполяции.

При программировании профиля с компенсацией радиуса инструмента следует помнить, что:

1) первое перемещение должно быть линейным, т.е. на быстром ходу или при скорости обработки (G00- G01);
2) блоки с функциями **M,H,S** и **T** не могут программироваться внутри цикла;
3) профиль может обрабатываться в непрерывном режиме (G27-G28) или в режиме «от точки к точке» (G29) в автоматическом или кадровом режиме;
4) компенсация радиуса инструмента деактивизируется при помощи функции G40, которая должна программироваться в последнем кадре профиля;
5) G00 не исключает компенсацию;
6) на первой и последней точке профиля центр инструмента позиционируется перпендикулярно профилю на программируемой точке.

При программировании выпуклого пути перемещением против часовой стрелки радиус (r), связывающий линии, должен иметь положительную величину; при перемещении по часовой стрелке программируется отрицательный радиус. Радиус r=0 оптимизирует путь инструмента путем генерирования радиуса, равного нулю на детали (см. рисунок 2.21).

Чтобы программировать наклон (в) с компенсацией инструмента, вводят величину наклона без знака. Устройство управления считывает наклон как расстояние от точки пересечения между линиями (см. рисунок 2.22).

В профиле GTL вы можете программировать компенсацию радиуса инструмента, включив операторы G21 и G41/G42 в тот же кадр. В этом случае вы также должны программировать коды отмены (G20 и G40) в одном кадре.

Примеры программирования обхода угла приведены на рисунках 2.20 и 2.21.

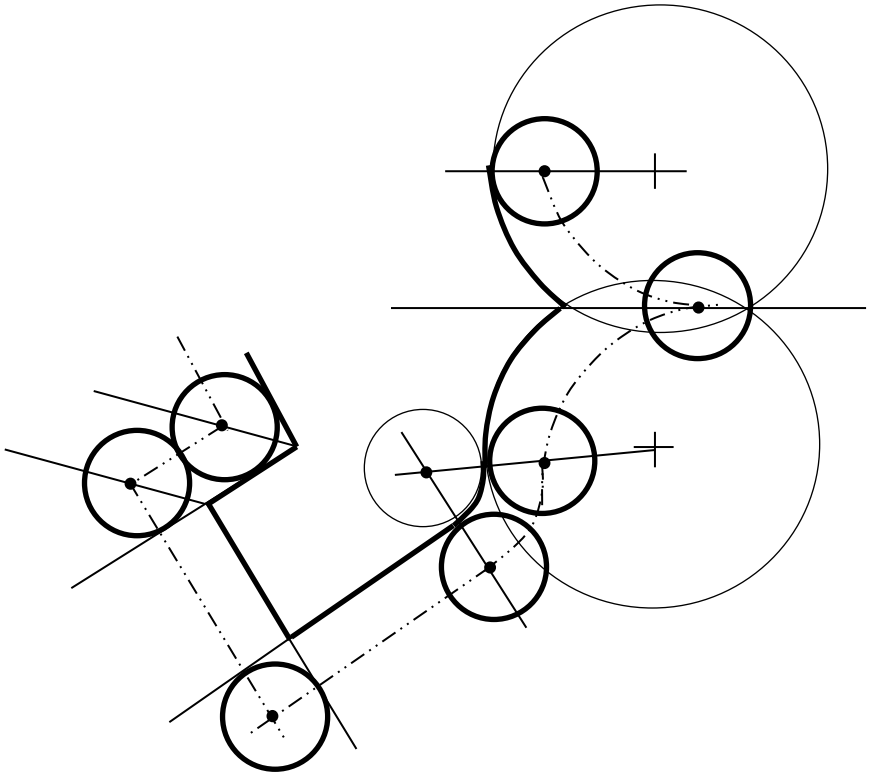


Рисунок 2.15 - Компенсация радиуса инструмента

Инструмент справа от детали

```
G1 G42 X-50 Y15 F200
X-20 Y45
```

Инструмент справа от детали

```
G1 G41 X-50 Y15 F200
X-20 Y45
```

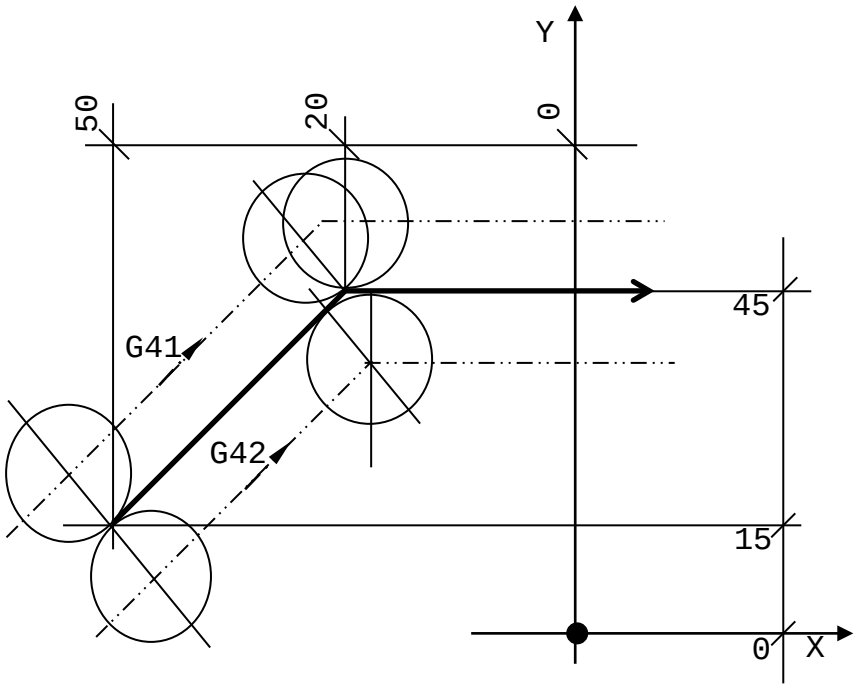


Рисунок 2.16 - Первый элемент профиля - линейный

```
G1 G42 X-31.622 Y40 F200
G2 X33.541 Y35 I J25

G1 G41 X-31.622 Y40 F200
G2 X33.541 Y35 I J25
```

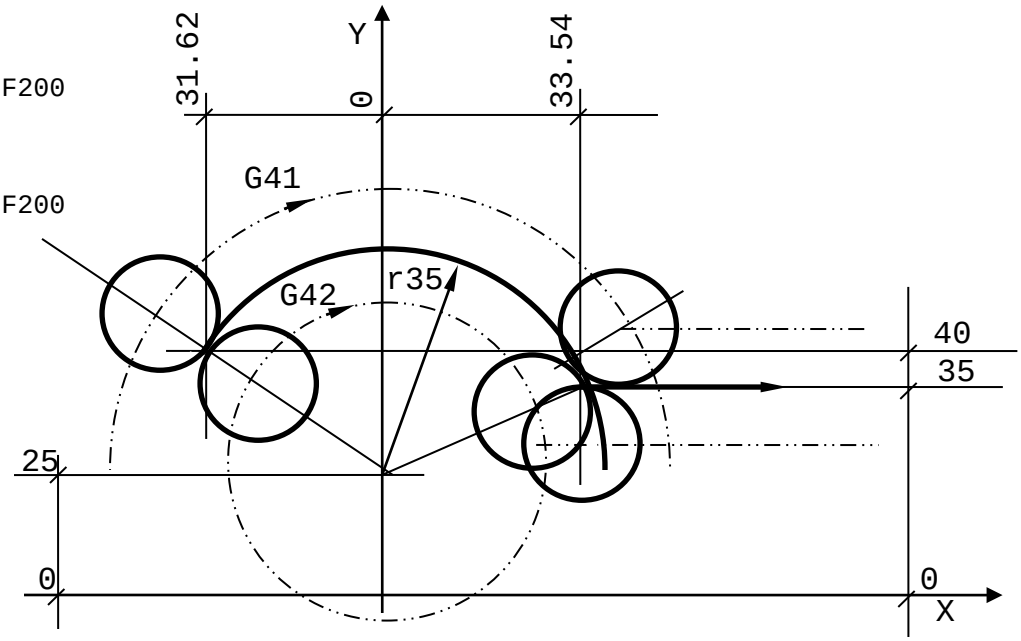


Рисунок 2.17 - Первый элемент профиля - круговой

```
N88 G1 G40 X50 Y15
N90 X.. Y..
```

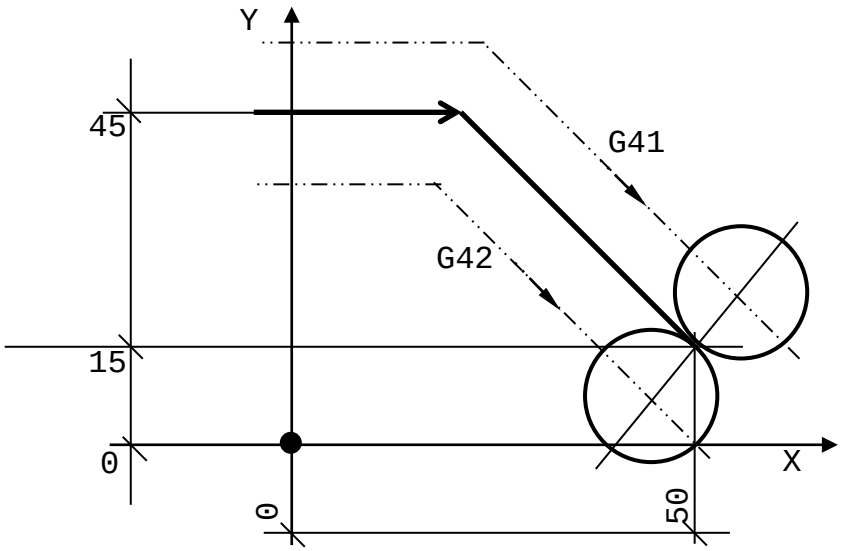


Рисунок 2.18 - Последний элемент профиля – линейный кадр

```
N99 G2 G40 X-31.622 Y40 I J25 F200
N100 G X.. Y..
```

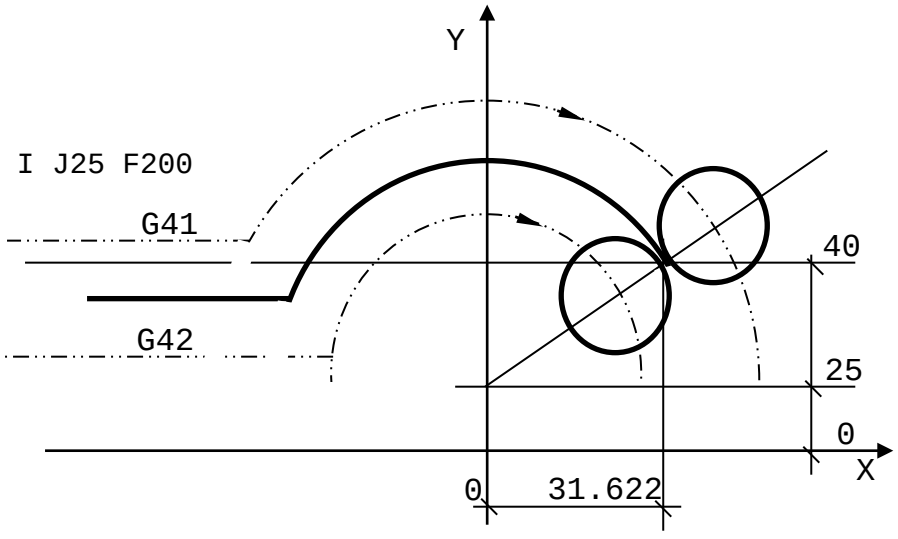
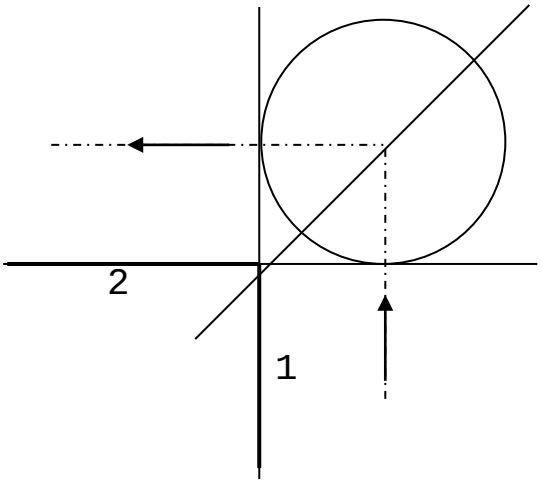
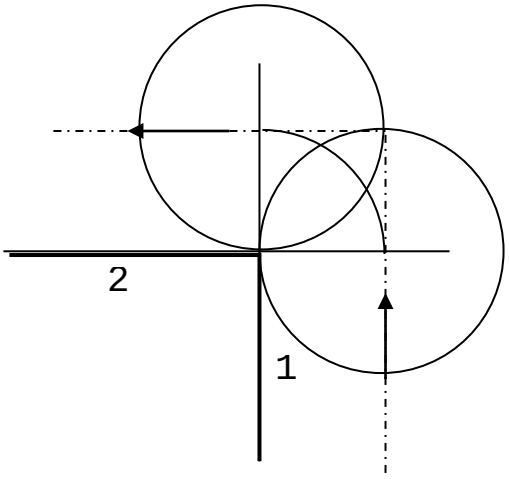


Рисунок 2.19 - Последний элемент профиля – кадр круговой интерполяции



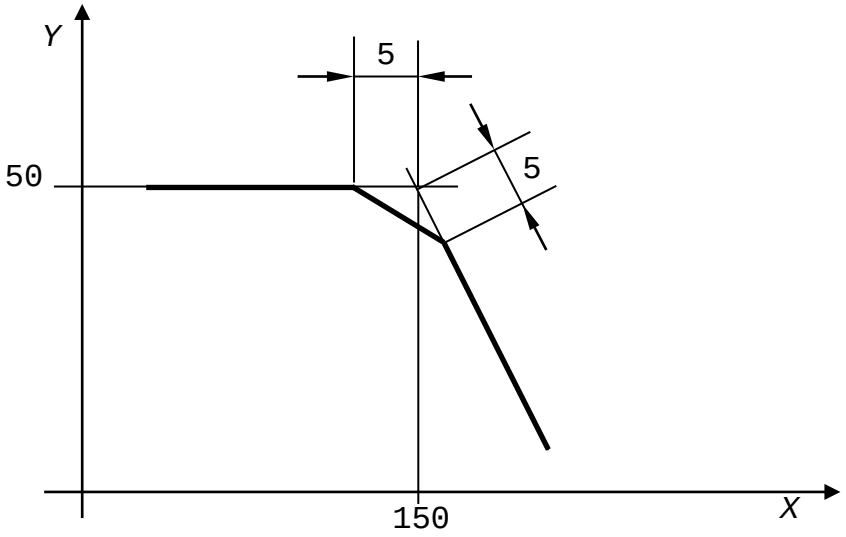

```
1  N20 G1 X100

2  N21 Z-100
```

Рисунок 2.20




```
1  N20 G1 X100
   N21 r0
2  N22 Z-100
```

Рисунок 2.21




```
.............
N10 G1 X50 Y
N11 X150 Y50
N12 b5
N13 X100 Y50
```

Рисунок 2.22 - Пример программирования скоса (фаски)

Пример программирования функций G41/G42/G40 приведён на рисунке 2.23

```
N1 S1500 T1.1M6
N2 G X85 Y60 M3
N3 Z-12
N4 G1 G41 X85 Y50 F220
N5 X100 Y23
N6 r0
N7 G3 Y-23 I110 J F180
N8 r-2
N9 G1 X85 Y-50
N10 X31.441 Y-31.803
N11 G3 X20 Y-40 I28.657 J-40 F100
N12 G2 X-20 I J-40 F250
N13 G1 Y F220
N14 G2 X-6.433 Y18.937 I J F250
N15 G1 G40 X85 Y50 F220
N16 G X85 Y60
N17 Z2
```

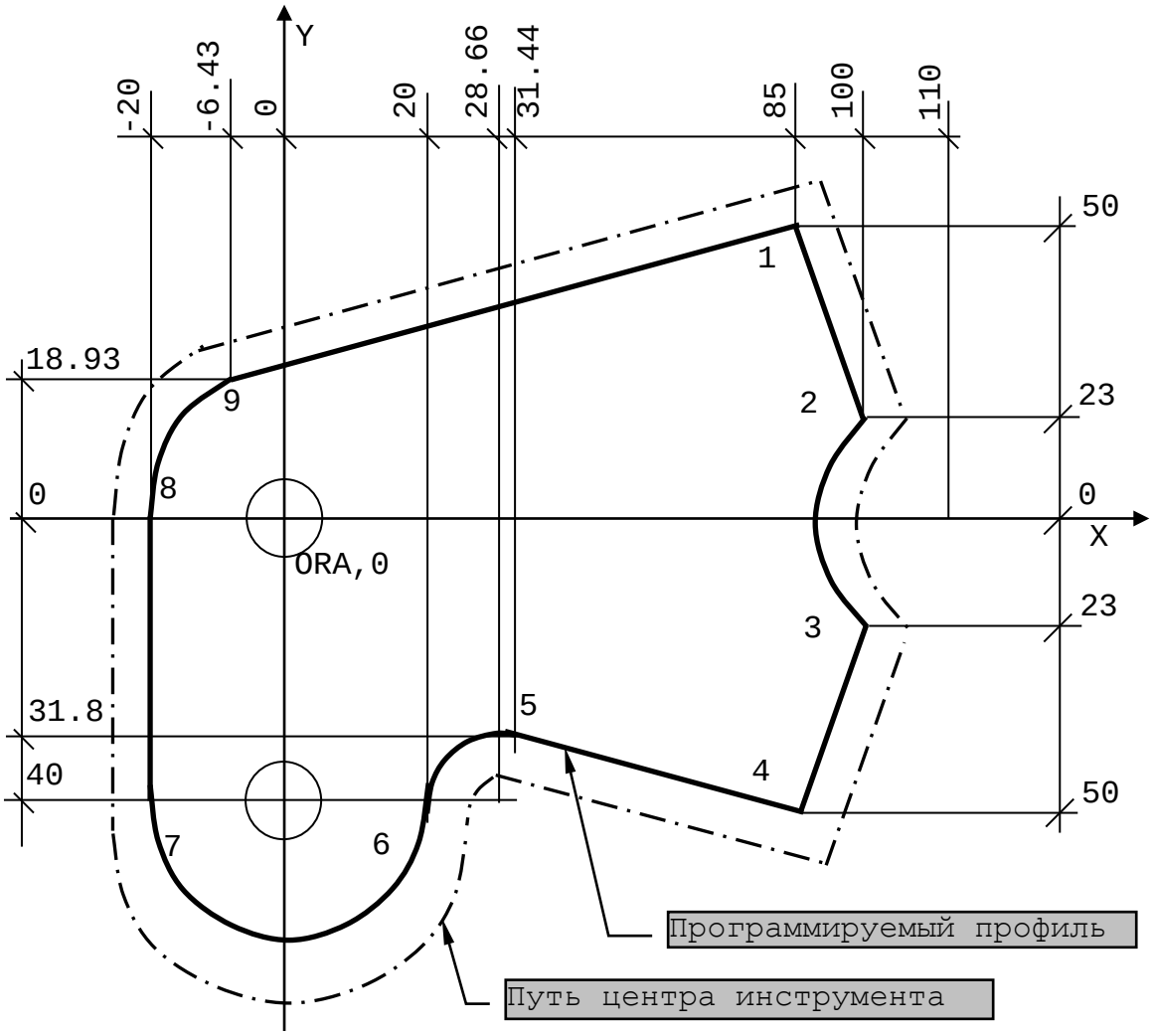


Рисунок 2.23 - Пример программирования функций G41/G42/G40

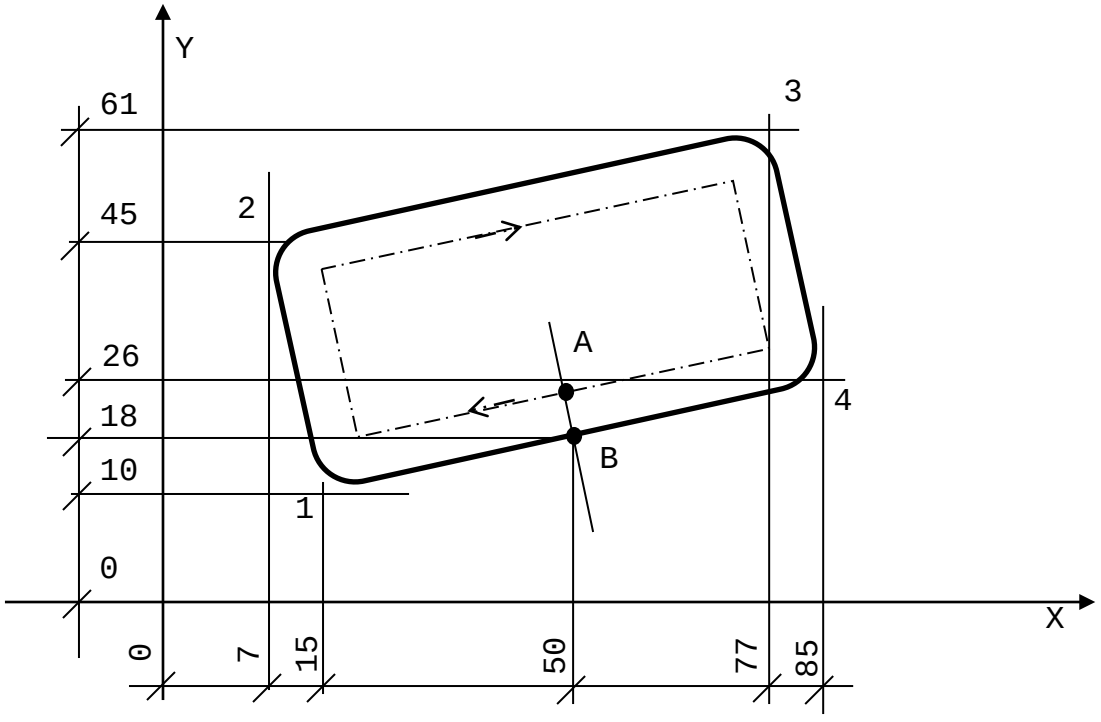


Рисунок 2.24 - Пример программирования паза с компенсацией радиуса инструмента

Пример программирования паза с компенсацией радиуса инструмента приведён на рисунке 2.24.

```
      N84 (DIS,”END MILL MD D=12”)
      N85 S1100 F170 T9.9 M6
A     N86 G X50 Y32 M3
      N87 Z-305
B     N88 G1 G42 X50 Y18 F170
1     N89 X15 Y10
2     N90 X7 Y45
3     N91 X77 Y61
4     N92 X85 Y26
B     N93 G40 X50 Y18
A     N94 G X50 Y32
      N95 Z M5
```

2.8.8 Система измерения (G70-G71)

Функции системы измерений **G** определяют единицу измерения. К этому классу принадлежат следующие функции:

G70 - программирование в дюймах;
G71 - программирование в миллиметрах.

Формат:

{G70} ,
{G71} [ДРУГИЕ G] [ОПЕРАНДЫ] .

Примечание - Если не запрограммированы ни G70, ни G71, то за единицу измерения принимается по умолчанию та, которая была определена в стадии конфигурации системы.

2.8.9 Постоянные циклы (G80-G89)

Функции постоянных циклов **G81** - **G89**, позволяют программировать ряд операций, таких как сверление, нарезание резьбы метчиком, растачивание и т.д., без повторения для каждой из них размеров отверстия. Характеристики постоянных циклов приведены в таблице 2.6.

Формат кадра постоянного цикла:

G8X[ДРУГИЕ G] [R1[R2]] КООРДИНАТА ЦИКЛА [ДОПОЛНИТЕЛЬНЫЕ ОПЕРАНДЫ] [СКОРОСТЬ ПОДАЧИ] [ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ] ,

где:

[ДРУГИЕ G] - это подготовительные функции, которые разрешаются программировать в кадре постоянного цикла;

[R1[R2]] - это координаты, определенные в явном или неявном виде (параметр E), относящиеся к оси шпинделя; они определяют координаты быстрого позиционирования в плоскости обработки в точке начала обработки и координаты возврата в конце обработки; если R2 отсутствует, то R1 считается конечной координатой;

КООРДИНАТА ЦИКЛА - определяет координату глубины отверстия, значение которой выражено в явном или неявном виде (параметр E), и ось, вдоль которой выполняется цикл;

[СКОРОСТЬ ПОДАЧИ] - определяется символом «F»; выражает скорость подачи, с которой выполняется обработка отверстия; если отсутствует, то скоростью подачи будет последняя запрограммированная «F»;

[ДОПОЛНИТЕЛЬНЫЕ ОПЕРАНДЫ] - являются операндами, определяющими параметры частных операций (например, I, J, K для глубокого сверления);

[ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ] - определяют функции S, M, T, H; последовательность движений при постоянных циклах можно представить следующим образом:

- быстрое позиционирование к оси отверстия;

- быстрый подход к плоскости обработки (размер R1);
- перемещение со скоростью рабочей подачи до запрограммированного размера (Z);
- функции цикла на дне отверстия;
- возвращение на быстром ходу или со скоростью рабочей подачи к координате R1 (R2), если координата возврата отличается от координаты подхода R1.

Таблица 2.6 - Характеристики постоянных циклов

Постоянный цикл	Подход	Функция на дне отверстия		Возврат
		выдержка времени	вращение шпинделя	
G81 сверление	рабочая подача	нет	нормальное	Ускоренное перемещение к R1 или R2, если программируется
G82 растачивание	рабочая подача	да	нормальное	Ускоренное перемещение к R1 или R2, если программируется
G83 глубокое сверление (с разгрузкой стружки)	в прерывистой работе (подход с рабочей скоростью с промежутком во время быстрого возврата или остановки)	да/нет	нормальное	Ускоренное перемещение
G84 нарезание резьбы метчиком	рабочая подача; начало вращения	нет	инверсное вращение	Рабочая подача к R1 ускоренное перемещение к R2, если программируется
G85 рассверливание или нарезание резьбы метчиком	рабочая подача	нет	нормальное	Рабочая подача к R1 ускоренное перемещение к R2, если программируется
G86 развертывание	рабочая подача; начало вращения шпинделя	нет	остановка	Ускоренное перемещение
G89 развёртывание с растачиванием	рабочая подача	да	нормальное	Рабочая подача к R1 ускоренное перемещение к R2, если программируется
G80 отмена постоянных циклов				

Для изменения значения R2 необходимо программировать R1 и R2 в одном и том же кадре.

Фаза ускоренного возврата производится, как движение с рабочей скоростью (G01) с быстрым ускорением.

Примечания общего характера, имеющие отношение ко всем постоянным циклам:

1. В кадре, содержащем функцию G постоянного цикла, не программируется никакое дополнительное движение осей, кроме самого цикла; цикл не приводится в действие, а кадр заносится в память системы. Цикл стартует координатами, запрограммированными сразу после кадра, содержащего постоянный цикл; (после выполнения первого цикла для того, чтобы выполнить последующие циклы, идентичные первому, достаточно запрограммировать координаты точек отверстия).
2. Продолжительность выдержки времени программируется трёхбуквенным кодом TMR.
3. Не представляется возможным программировать G8X, если профиль запрограммирован на языке GTL и/или внутри G41/G42 – G40.
4. Функции G8X являются модальными. Невозможно программировать новый постоянный цикл без закрытия предыдущего постоянного цикла с G80.

Пример

Постоянный цикл с R2=R1 и R2 не равно R1 приведён на рисунке 2.25.

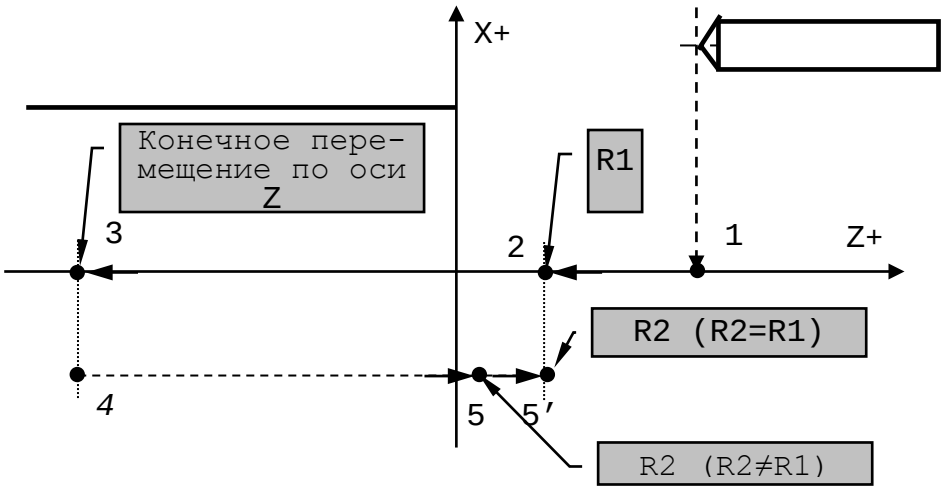


Рисунок 2.25 - Постоянный цикл с R2=R1 и R2 не равно R1

2.8.9.1 Постоянный цикл сверления (G81)

Кадр программирования: **G81 [R1..[R2..]] Z..**

Постоянный цикл G81 может быть также использован для операций растачивания, развертывания и центровочного сверления. Программирование постоянных циклов G82, G85, G86, G89 идентично программированию G81. В кадры, предшествующие постоянным циклам G82 и G89, вводится при необходимости выдержка времени через команду TMR.

Пример

```
N33 TMR=2
N34 G82 R3 Z-100 T6 M13
N35 X35 Y150
N36 G80
```

В данном примере выдержка времени равна 2 секунды.

Пример 1. Постоянный цикл сверления G81 приведён на рисунке 2.26.

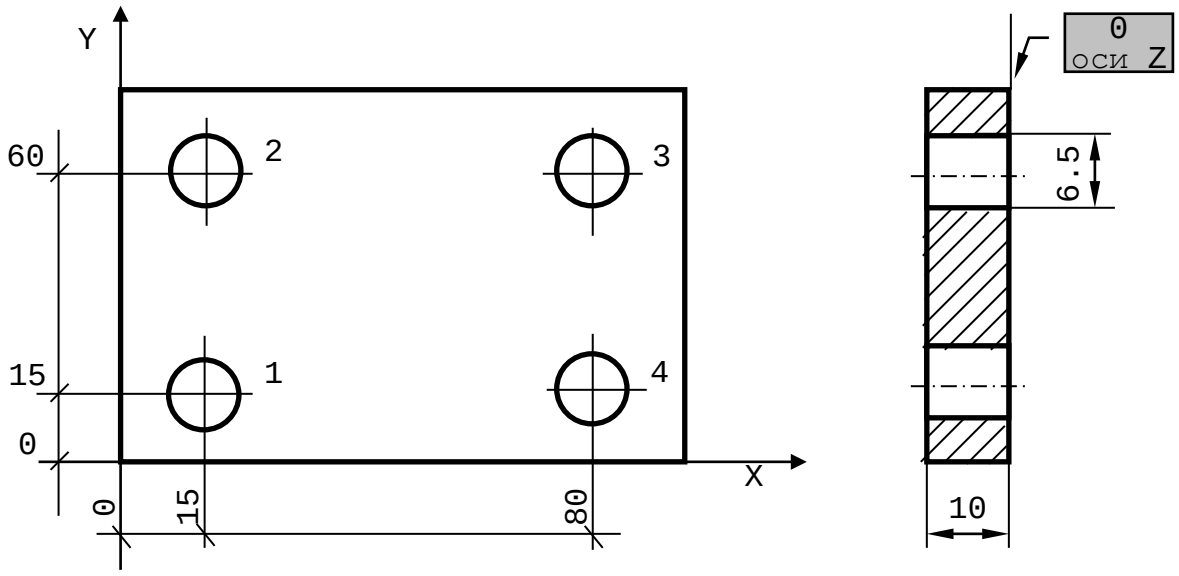


Рисунок 2.26- Пример программирования постоянного цикла сверления G81

```
.....................
N32 S1100 F95 T3.3 M6
N33 G81 R3 Z-15 M3 - задание параметров постоянного цикла сверления
N34 X15 Y15        - движение к точке 1 и выполнение цикла
N35 Y60            - движение к точке 2 и выполнение цикла
```

```
N36 X80            - движение к точке 3 и выполнение цикла
N37 Y15            - движение к точке 4 и выполнение цикла
N38 G80 Z50 M5     - отмена действия цикла
```

Пример 2. Программирование постоянного цикла сверление G81 приведено на рисунке 2.27.

```
N31 (DIS,"TWIST DRILL D=6.5)
N32 G97 S1000 T4.4 M06 M3 M7
N33 G81R5 Z-70 F45
N34 X0
N35 G80
N36 G.. X.. Z..
```

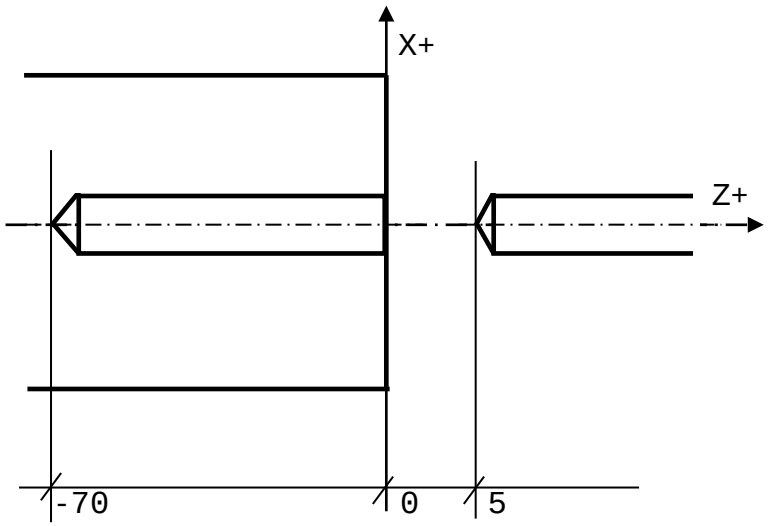


Рисунок 2.27 - Пример программирования постоянного цикла G81

Пример программирования постоянного цикла с двумя параметрами приведён на рисунке 2.28.

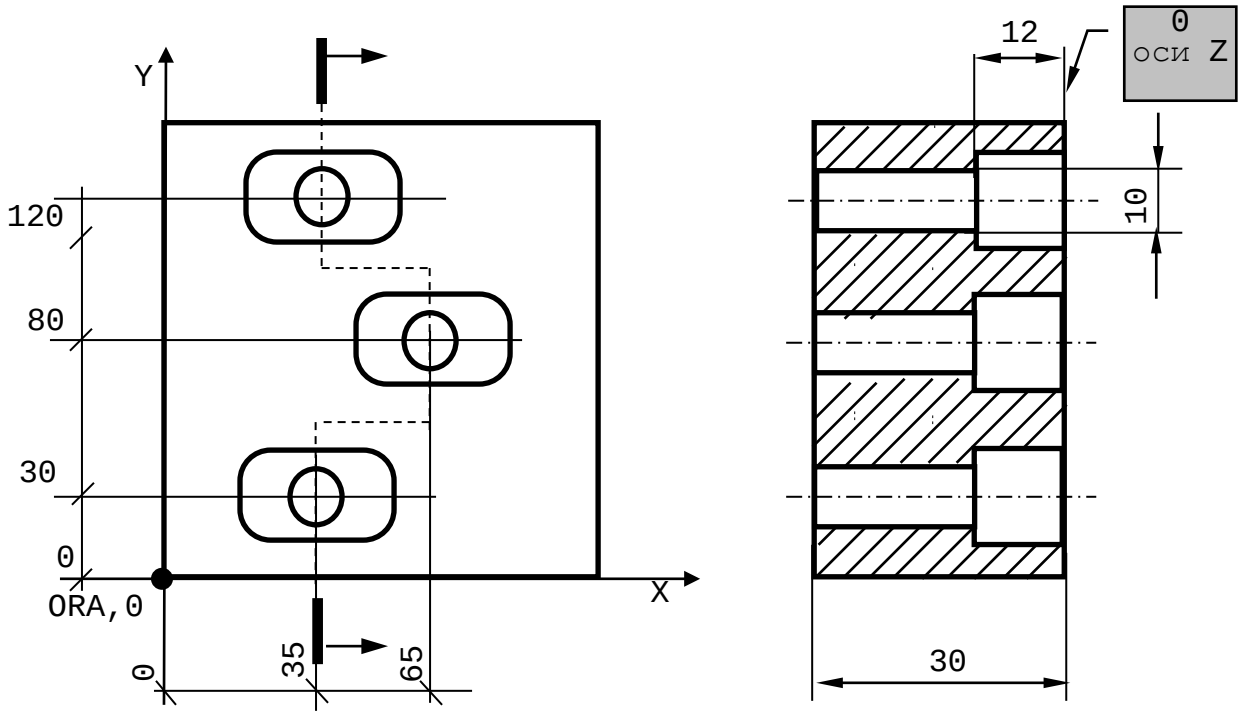


Рисунок 2.28 - Пример программирования постоянного цикла с двумя параметрами

```
N43 S850 F100 T4.4 M6
N44 G81 R-10 R2 Z36 M3
N45 X35 Y30
N46 X65 Y80
N47 X35 Y120
N48 G80 Z50 M5
```

2.8.9.2 Постоянный цикл глубокого сверления (G83)

Формат кадра:

G83 [R1..[R2..]] Z..I..[K..] [J..] ,

где:

[R1] - начальная координата отверстия (как для G81);
[R2] - координата точки возврата (как для G81);
Z - координата дна отверстия (как для G81);
I - приращение размера Z после каждого цикла разгрузки стружки;
[J] - минимальное приращение цикла разгрузки стружки; после достижения программированного значения следуют постоянные приращения;
[K] - коэффициент уменьшения параметра **I**(до достижения величины **J**).

Присутствие или отсутствие этих параметров определяет два разных цикла:

1) в случае, при котором были запрограммированы **I, K, J**, цикл имеет следующие шаги:

   - быстрый подход к оси отверстия для обработки;
   - быстрый подход к точке R1;
   - подход с рабочей подачей к точке R1+I;
   - быстрый возврат к точке R1 (разгрузка стружки);
   - вычисление нового значения R1: R1=R1+I-1;
   - вычисление нового значения I:

     I=I * K , если **I * K >= J.**
     I=J , если **I * K < J**.

   Шаги, начиная со второго, выполняются один за другим до получения запрограммированного размера глубины сверления.

Примечание - Для сохранения параметра **I** неизменным (постоянное приращение) запрограммировать **K=1** в отсутствии параметра **J**.

2) случай, при котором не были запрограммированы **K** и **J** (дробление стружки без разгрузки) - подача с постоянным приращением и выдержка времени при любом приращении, обеспечивается следующими шагами:

   - быстрый подход к центру отверстия для обработки;
   - быстрый подход к размеру R1;
   - рабочая подача к точке R1+I;
   - выдержка времени, запрограммированная с TMR;
   - подход по другой величине I;
   - (три последних шага следуют один за другим до достижения запрограммированного размера глубины);
   - быстрый выход из отверстия к точкам R1 или R2, если R2 запрограммирован.

Пример (рисунок 2.29):

```
N66 S930 F65 T6.6 M6
N67 G83 R3 Z-55 I20 R.8 J6 M13      - задание цикла
N68 X-15.81 Y-22.2                  - сверление в точке 1
N69 X23                             - сверление в точке 2
N70 X9 Y35.8                        - сверление в точке 3
N71 G80 Z50 M5                      - подъем инструмента и отмена цикла
```

Приведенный пример УП обеспечивает обработку трёх отверстий в абсолютной системе координат в точках:

   - 1 (в кадре N68);
   - 2 (в кадре N69);
   - 3 (в кадре N70).

С помощью функции G83 в кадре N67 объявляется цикл, который будет выполняться в точках, заданных последующими кадрами УП. Отменяется функцией G80 в кадре N17. Таким образом, в точках 1, 2, 3 обеспечивается выполнение цикла глубокого сверления с периодическим выводом инструмента.

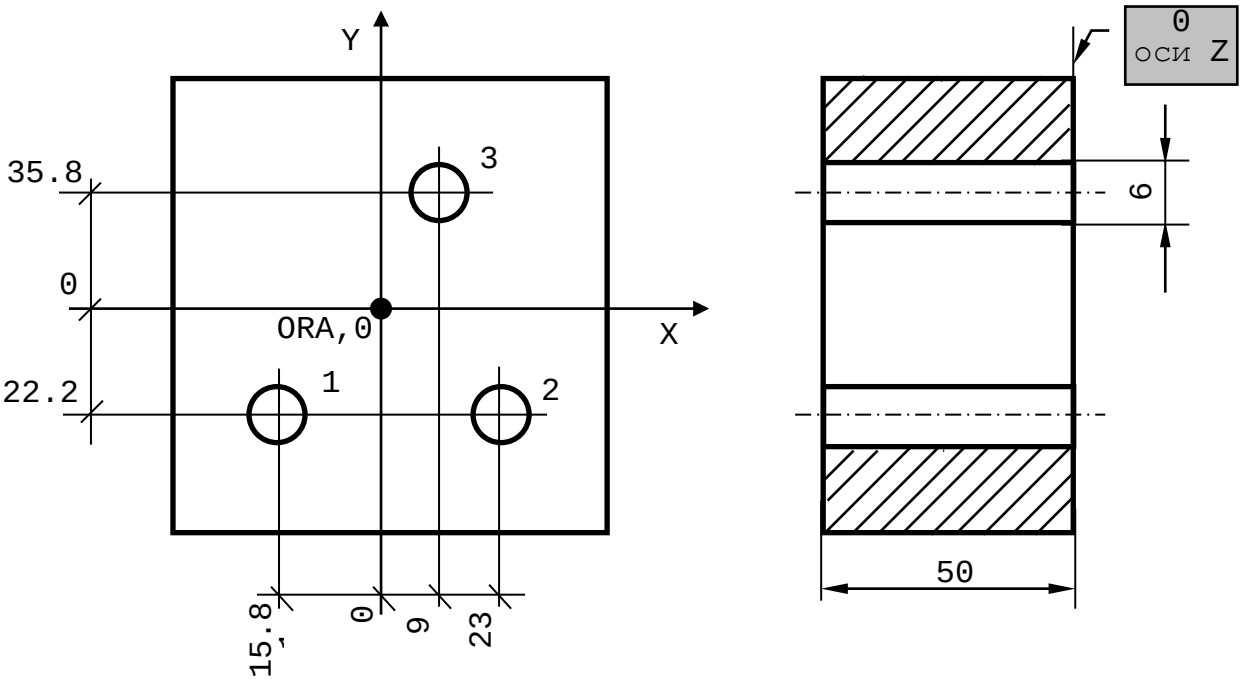
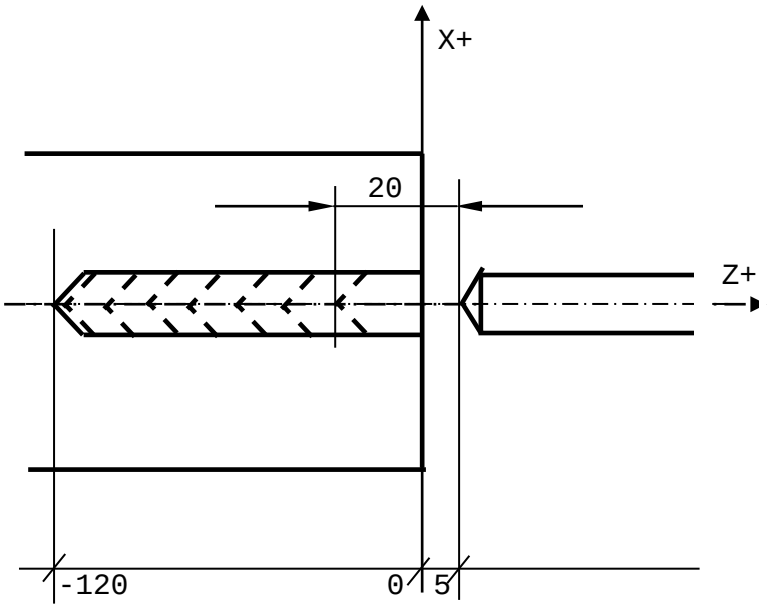


Рисунок 2.29

Пример программирования постоянного цикла G83 приведён на рисунке 2.30.




```
N61 (DIS,”TWIST DRILL D=6)
N62 G97 S1000 T3.3 M6 M3 M7
N63 G83R5 Z-120 I20 K.8 J6
N64 X0 Y0
N65 G80
N66 G.. X.. Z..
```

Рисунок 2.30 - Пример программирования постоянного цикла G83(глубокое сверление с разгрузкой стружки)

2.8.9.3 Постоянный цикл нарезания резьбы метчиком (G84)

Постоянный цикл нарезания резьбы метчиком (G84)может быть выполнен двумя способами.

1) Шпиндель без датчика.

Формат кадра цикла G84:

G84 [R1..][R2..] Z.. ,

где:

G84 - код цикла нарезания резьбы метчиком;
[R1] - координата точки начала обработки и конца обработки, если R1=R2 (размер быстрого подхода и возврата на рабочей скорости);
[R2] - координата точки конца обработки, если R1 ≠ R2;
Z - конечная координата нарезания резьбы.

При программировании необходимо учитывать следующее:

- размер перемещения быстрого хода инструмента к детали в операциях нарезания резьбы метчиком должен всегда заканчиваться на расстоянии от детали, равном пяти шагам резьбы, если глубина ≤ 3 диаметров, или семи шагам, если глубина > 3 диаметров;
- скорость подачи F, которую следует запрограммировать, вычисляется следующим образом:

F = S * p * 0.9 ,

где:

S - скорость вращения шпинделя;
P - шаг резьбы;
0,9 - коэффициент уменьшения скорости для сохранения упругости пружинного компенсатора резцедержателя.

Посредством кода RMS, задаваемого в программе или введенного с клавиатуры, можно изменять скорость возврата инструмента, определяя это изменение в процентах. Окончательный размер Z должен быть уменьшен на величину, равную 10% от фактического рабочего хода метчика.

Пример приведён на рисунке 2.31.

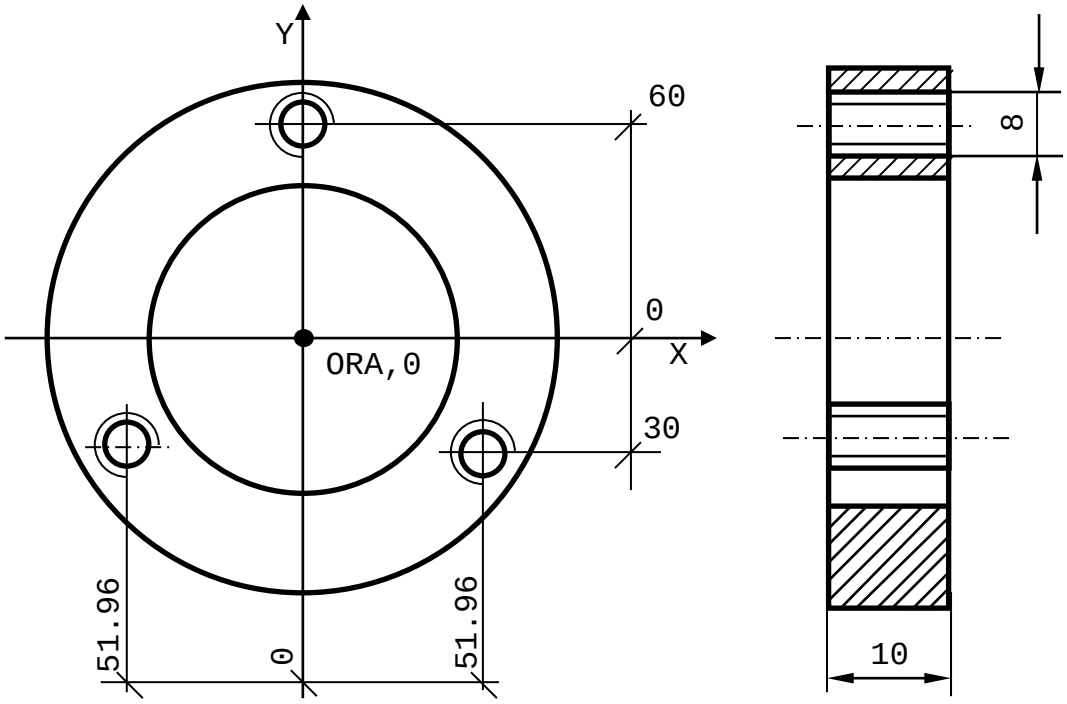


Рисунок 2.31

```
N91 S280F315 T8.8 M6 M13
N92 G84 R7 Z-15
N93 X-51.96 Y-30            - нарезание резьбы в точке 1
N94 X51.96                  - нарезание резьбы в точке 2
N95 X Y60                   - нарезание резьбы в точке 3
N96 G80 Z50 M5
```

В рассматриваемом примере с помощью функции G84 в кадре N92 объявляется цикл нарезания резьбы метчиком, который будет выполняться в точках 1, 2, 3 (рисунок 30), заданных следующими кадрами УП:

- точка 1 (в кадре N93)
- точка 2 (в кадре N94)
- точка 3 (в кадре N95)

Кадр N92 обеспечивает быстрый подвод инструмента в точку (R1=7мм), нарезание резьбы метчиком на глубину, определяемую координатой «Z» (Z-15), и возврат к точке R1 на рабочей скорости. Цикл отменяется функцией G80 в кадре N96.

Эта программа используется для нарезания правосторонней резьбы метчиком (вращение вправо), что обеспечивается функцией M13, запрограммированной в кадре N91. Если необходимо запрограммировать нарезание резьбы влево, достаточно запрограммировать функцию M14 (M04) вместо M13 (M03).

Если рабочий путь недостаточен для выполнения разгона/торможения, подаётся сигнал ошибки.

2) Шпиндель с датчиком.

В данном случае существует два способа программирования функции **G84**:

- использование программирования скорости подачи F, как в случае для шпинделя без датчика;
- использование программирования шага резьбы **K**; в этом случае система автоматически вычисляет подачу, умножая шаг K на число оборотов шпинделя.

Формат кадра цикла **G84**:

G84 [R1..][R2..] Z.. K ,

где:

G84 - код цикла нарезания резьбы метчиком;
[R1] - координата точки начала обработки и конца обработки, если R1=R2 (размер быстрого подхода и возврата на рабочей скорости;
[R2] - координата точки конца обработки, если R1 ≠ R2;
Z - конечная точка нарезания резьбы метчиком;
[K] - шаг резьбы.

Пример приведён на рисунке 2.31:

```
N91 S280 T8.8 M6 M3
N92 G84 R7 Z-15 K1
N93 X-51.96 Y-30
N94 X51.96
N95 X Y60
N96 G80 Z50 M5
```

Данный фрагмент программы задает обработку трёх отверстий. В кадре N92 задан шаг резьбы **K1**. Система автоматически рассчитывает величину подачи на основе информации по адресам **S** и **K**.

2.8.9.4 Особенности постоянных циклов

1) Если внутри постоянного цикла программируется кадр типа **X,Y,R** или же **X,Y,R** и/или **Z**, размеры **R** и/или **Z** постоянного цикла будут изменены, и движения осей будут выполнены в следующем порядке:

- X и Y;
- R обновлённая;
- Z обновлённая.

Это позволяет изменять глубину отверстия и переходить от обработки на одной плоскости к обработке на плоскости ниже без отмены постоянного цикла функцией G80.

Пример приведён на рисунке 2.32:

```
          N35 (DIS,".............")
          N36 S1000 F100 T4.4 M6
          N37 G81 R3 Z-42 M3
1         N38 X15 Y15
2         N39 X65
3         N40 Y85 R-13
4         N41 X15
          N42 G80 Z50 M5
```

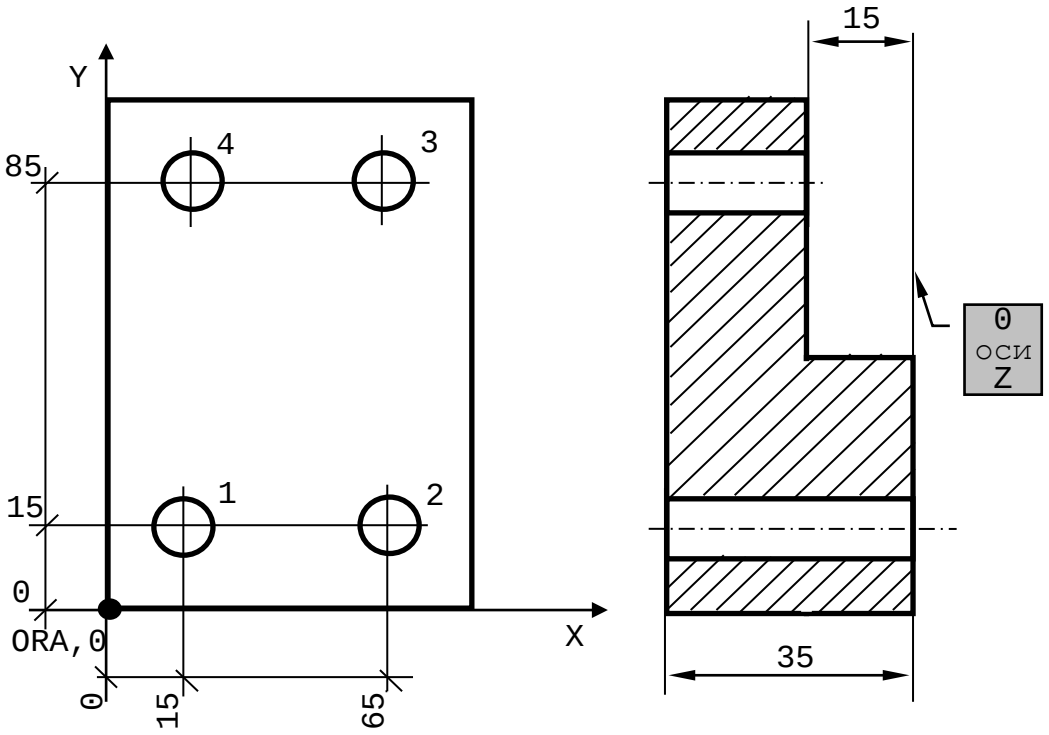


Рисунок 2.32

В рассматриваемом примере обработка отверстий в точках 1 и 2 (кадры N38, N39) осуществляется в соответствии с параметрами цикла сверления по функции G81, заданными в кадре N37(R3,Z-42). Кадр N40, кроме координат точки 3, задает новое значение координаты точки начала обработки и возврата в конце обработки отверстия (R-13). Таким образом, отверстия в точках 3 и 4 обрабатываются с параметрами цикла R-13, Z-42 при выполнении кадров N40 и N41 соответственно.

2) Внутри постоянного цикла, если программируется кадр типа **X Y R1 R2** (последнее отверстие нижней плоскости), размеры **R1** и **R2** постоянного цикла будут обновлены, и движения будут выполнены в следующем порядке:

- X и Y перемещения к точке;
- выполнение постоянного цикла с обновлёнными R1 и R2 (в конце цикла шпиндель подходит с быстрой скоростью к новой точке возврата R2), что даёт возможность перейти от обработки нижней плоскости к обработке более высокой плоскости без отмены постоянного цикла. Для обработки первого отверстия на более высокой плоскости следует запрограммировать кадр типа X Y R2.

Пример приведён на рисунке 2.33.

```
          N42 (DIS,".............")
          N43 S1000 F100 T5.5 M6
          N44 G81 R-18 Z-46 M13
1         N45 X25 Y25
2         N46 X60 R-18 R-8
3         N47 Y75 R-8 R2 Z-25
4         N48 Y175 R-3 Z-46
5         N49 X95
          N50 G80 Z50 M5
```

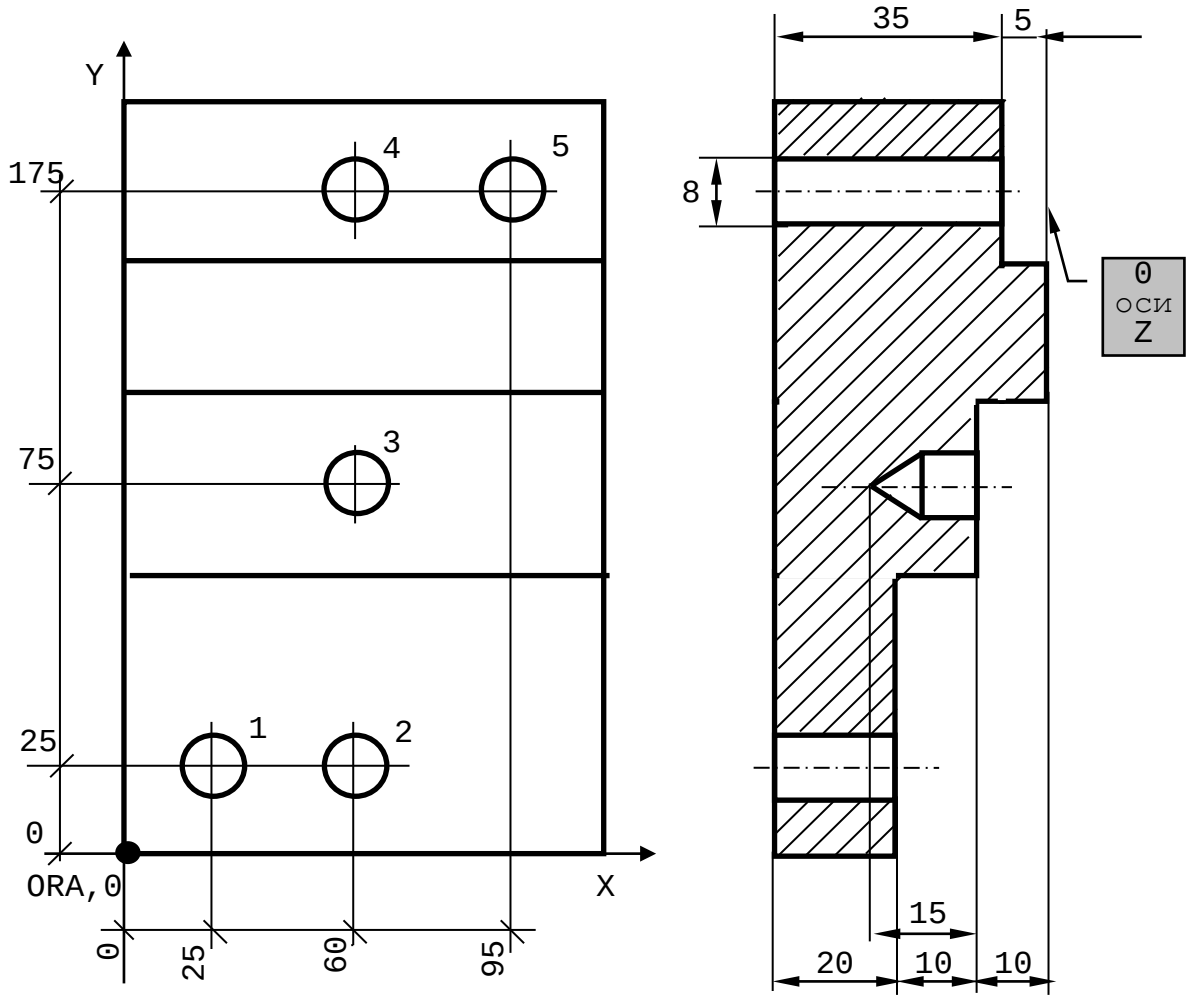


Рисунок 2.33

При выполнении этой программы в точке 1 (кадр N45) обеспечивается выполнение постоянного цикла с параметрами, заданными в кадре N44 (R-18,Z-46), где параметр R1=R2. В кадре N46 указаны новые значения параметров цикла, где R1 не равно R2, а глубина сверления осталась прежней. Параметр R-8 обеспечивает выход инструмента после обработки отверстия в новую позицию для последующего перемещения инструмента в точку 3.

В кадре N47, кроме значений положения инструмента в начальной и конечной точках цикла, указана новая глубина сверления (Z-25) в точке 3. Новое значение параметра R2 обеспечивает дальнейшее перемещение инструмента в позицию точки 4. При обработке кадров N48 и N49 выполняется сверление отверстий в точках 4 и 5 с одинаковыми параметрами цикла на глубину (Z-46). Положение точки начала и конца цикла определяется параметром R=3 (R1=R2).

2.8.10 Программирование в абсолютной системе, по приращениям и относительно нуля станка (G90-G91-G79)

Функциями, определяющими тип программирования (в абсолютной системе, по приращениям, относительно нуля станка) являются:

G90 - программирование в абсолютной системе (движения относительно фактической начальной точки);

G91 - программирование в системе по приращениям (движения относительно последнего местоположения);

G79 - программирование относительно нуля станка.

Примечание - Функция G79 действительна только в том кадре, в котором запрограммирована.

Формат:

{G90} ,
{G91} [ДРУГИЕ G] [ОПЕРАНДЫ] ,
{G79} .

Если ни одна из этих функций не запрограммирована, автоматически осуществляется программирование в абсолютной системе в отношении объявленных начальных точек.

Функции G90 и G91 являются модальными, в то время как G79 - нет. После программирования кадра с G79 система возвращается в состояние программирования (G90/G91), которое было активным в предыдущем кадре (G90/G91);

Программирование по приращениям несовместимо с программированием на языке GTL.

Пример приведён на рисунке 2.34.

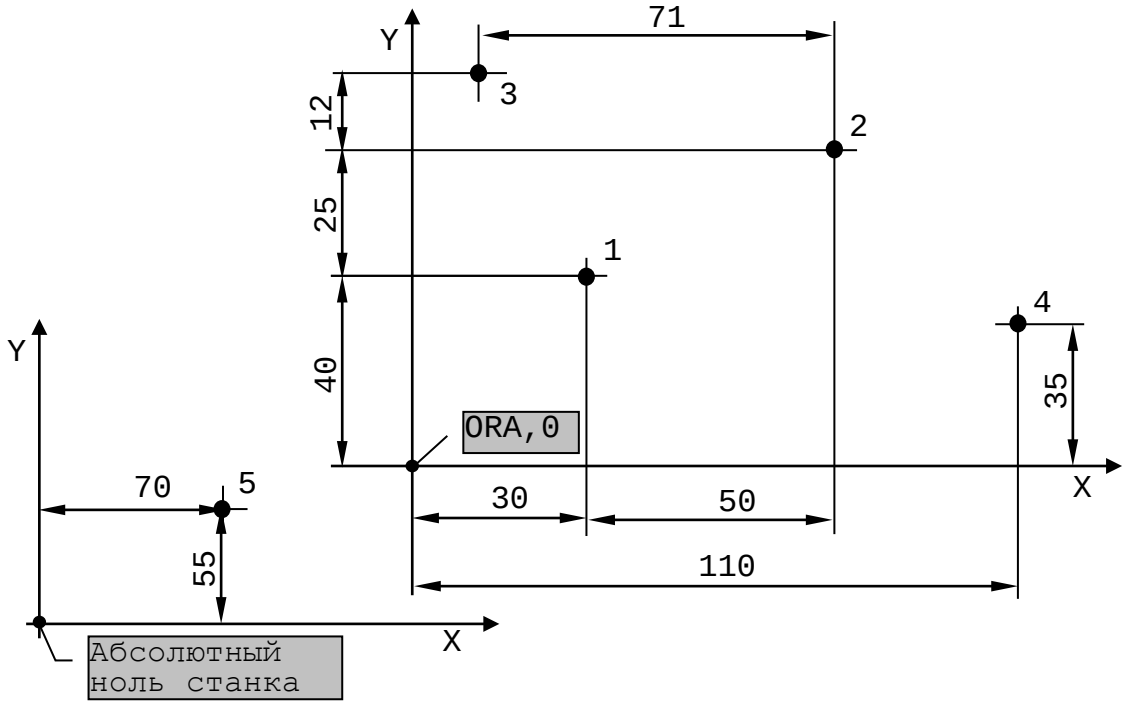


Рисунок 2.34

N1 GXY	перемещение осей X, Y к начальной точке
N2 X30 Y40	перемещение осей X, Y к точке 1
N3 G91 X50 Y25	перемещение по приращениям к точке 2 (X+50,Y+25 относительно точки 1)
N4 X-71 Y12	перемещение осей к точке 3 (X-71,Y+12 относительно точки 2)
N5 G90 X110 Y35	перемещение осей к точке 4 (X+110,Y+35 относительно начальной точки)
N6 G79 X70 Y55	перемещение осей относительно нуля станка к точке 5 (X+70,Y+55 по отношению к нулю станка)

2.8.11 Характеристики динамического режима G04, G09

К этому классу принадлежат функции:

G04 - выдержка времени в конце кадра;
G09 - замедление в конце кадра.

Формат:

{G04} ,
{G09} [ДРУГИЕ G] [ОПЕРАНДЫ] ,

где:

G04 - осуществляет выдержку времени в конце кадра. Время выдержки запрограммировано в кадре назначения **TMR = значение** (команда TMR будет рассмотрена далее); функция G04 действительна только в том кадре, в котором запрограммирована;

G09 - устанавливает скорость, равную 0, в конце кадра, где она была запрограммирована, но не изменяет ранее установленный режим динамики профиля, если он находится в процессе обработки; функция действительна только в том кадре, в котором запрограммирована.

2.8.12 Измерительные циклы (G72-G73-G74)

Устройство измерения, установленное на шпинделе, позволяет выполнить три цикла измерения:

G72 - измерение координат точки прямолинейным движением (с корректировкой радиуса;

G73 - измерение параметров отверстия;

G74 - измерение координат точки (без корректировки радиуса).

2.8.12.1 Измерение координат точки прямолинейным движением

Функция G72 измеряет при помощи щупа координаты точки в пространстве прямолинейным движением и заносит их в память системы как параметры **E**, определённые в цикле (запоминание начинается с запрограммированного параметра). Измерение выполняется с корректировкой радиуса щупа.

Формат:

G72 ось [ось] [ось] En ,

где:

ось - максимально 3 оси; осями являются запрограммированные оси, перемещения осуществляются в номинальных величинах;

En - определяет параметр, от которого необходимо начать запоминание размеров, вычисленных щупом.

Пример

G72 X100 Y50 E32 В E32 и E33 запоминаются соответственно вычисленные величины для X,Y.

В кадре с функцией G72 программируется перемещение заведомо большее, чем расстояние до измеряемой точки. По сигналу от датчика касания устройство фиксирует абсолютное положение измеряемой точки по всем координатам, заданным в кадре, и заносит эти значения в параметры **E**, начиная с номера параметра, указанного в кадре (**En**). В E32 и E33 запоминаются соответственно вычисленные величины для X и Y.

2.8.12.2 Измерение параметров отверстия

Функция G73 измеряет при помощи щупа параметры отверстия в данной плоскости интерполяции и заносит их в память системы как параметры **E**, определенные в цикле измерения (запоминание начинается с запрограммированного параметра). Оси металлорежущего станка должны быть размещены в центре отверстия. Полученными параметрами являются координаты центра и радиус отверстия. Измерение осуществляется с корректировкой радиуса щупа.

Формат:

G73 r En ,

где:

r - определяет теоретический радиус отверстия;

En - определяет параметр, от которого начинается запоминание параметров отверстия.

Пример

G73 r100 E55 - В E55, E56 и E57 заносятся соответственно абсцисса, ордината и радиус окружности.

2.8.12.3 Измерение координат точки

При задании функции G74 система определяет разницу между номинальными размерами (т.е. измеренными при помощи установленного щупа) и размерами, измеренными при помощи установленного инструмента. Этот цикл может быть использован для переквалификации инструмента или для определения его состояния. При вычислении полученных размеров корректировка радиуса не учитывается, т.е. проверяется фактический размер «инструмента».

Формат:

G74 ось [ось] [ось] En ,

где:

оси - максимально 3 одновременных оси;
En - определяет параметр, с которого начинается запоминание измеренных смещений.

Пример

```
G74 X100.2 E41
E41 =Pt - Pm           ,
```

где Pm - измеренная точка;
Pt - теоретическая точка.

Цикл состоит из тех же фаз, что и цикл G72. Разница между номинальной и действительной координатами хранится в E41 (положительная, если действительная координата измеряется раньше номинальной координаты, и отрицательная, если действительная координата измеряется после номинальной координаты).

2.8.13 Инверсная скорость подачи, задаваемая через параметр времени (G93)

Функция G93 определяет скорость подачи осей, выраженную как инверсия времени (в минутах), необходимого для выполнения элемента. Зная скорость и расстояние движения, можно вычислить значение **F** по следующим формулам:

1) линейная интерполяция:

$$F = \frac{\textbf{скорость подачи}}{\textbf{расстояние}} \qquad (2.5);$$

2) круговая интерполяция:

$$F = \frac{\textbf{скорость подачи}}{\textbf{дуга}} \qquad (2.6),$$

где:

скорость подачи - скорость линейная или круговая, выраженная в мм/мин (G71) или дюймах/мин (G70);
расстояние - векторное расстояние линейного движения, запрограммированное в мм или дюймах;
радиус - дуга, запрограммированная в мм или дюймах.

С активной G93 **F** действительна только в кадре, в котором она была запрограммирована.

Пример

```
G93 G1 X...Y...F...
       X...Y...F...
```

2.8.14 Остановка вращения шпинделя с угловой ориентацией (M19)

При использовании вспомогательной функции M19 представляется возможным осуществить остановку шпинделя с угловой ориентацией, причем значение угла указывается при характеризации системы или устанавливается из программы логики станка. Используется, когда необходимо осуществить обработку в натяжении. Для

этого следует сориентировать шпиндель, передвинуть ось Z (или ось X, в зависимости от расположения резца), войти в отверстие, заново сориентировать шпиндель по оси и начать обработку. Функция может быть также применена в операциях особо точного растачивания для избежания повреждений на расточенной поверхности при обратном ходе оси. M19 аннулируется функциями M03, M04, M13, M14. Когда считывается функция M19 в кадре, содержащем информацию движения, сначала выполняется функция, а затем - движение.

Пример приведён на рисунке 2.35.

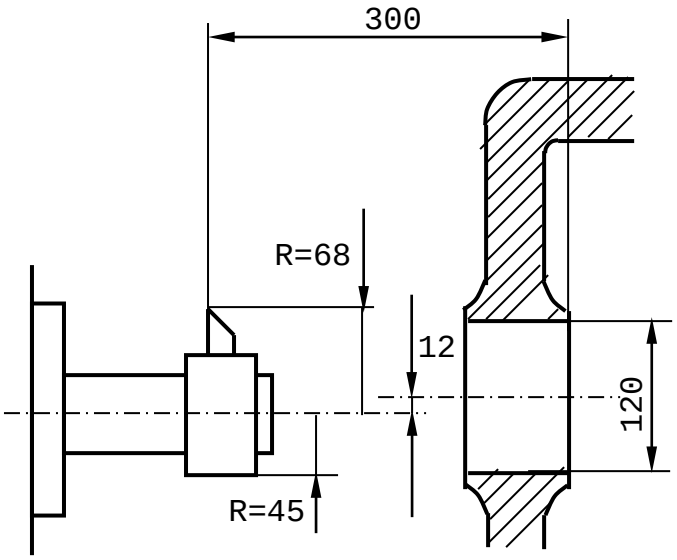


Рисунок 2.35

2.8.15 Блокирование осей (M10)

Если предусмотрено интерфейсом, вспомогательная функция M10 осуществляет блокирование осей, которые не должны двигаться во время обработки. Никакого особого внимания не требуется при программировании движений «от точки к точке» (G29). При непрерывном режиме (G27-G28) в кадре начала обработки, т.е. в кадре, содержащем M10, должны быть определены все оси, которые будут смещены в профиле, для избежания их блокирования. Если это условие не выполнено, блокированная ось будет пытаться двигаться, вследствие чего выдается сигнал «СБОЙ ПРИВОДА».

Примеры

```
1)    N8  G G29 X100 Y100
      N10 G1 X-100 M10 F250
      N11 Y-100
      N12 X100
      N13 Y100
      N14 G X.. Y.. M11
      ..............

2)    N9 G G27 X100 Y100
      N10 G1 X-100 Y100 M10 F250
      N11 Y-100
      N12 X100
      N13 Y100
      N14 G X.. Y.. M11
      ..............
```

В примере 1 только движущаяся ось Z остается деблокированной, в то время как в примере 2 остаются деблокированными оси X, Y, определённые в кадре N10. Функция M10 аннулируется функцией M11. Эти же условия действительны для функции M12, имеющей отношение к вращающимся осям.

2.8.16 Синхронизация начала движения со шпинделем (G35)

Синхронизация начала движения, заданного в кадре, с ноль-меткой шпинделя выполняется при программировании функции G35.

Формат:

G35 [другие коды G] [операнды] ,

Функция G35 действует только в том кадре, где она запрограммирована. Функцию G35 нельзя программировать, если в УЧПУ активна функция G00.

При необходимости синхронизировать начало выполнения группы кадров функцию G35 необходимо программировать в первом кадре этой группы.

Пример

```
N1 G97 G94 S100 M3
N2 G0 X Y
N3 G1 G35 X10 Y20 F100
N4 G2 I J
N5 G0 X Y
```

2.9 Осепараллельные коррекции (u,v,w)

Программируя факторы корректировки u,v,w в кадре обработки, инструмент перемещается в точку, имеющую координаты, равные запрограммированным координатам, плюс радиус инструмента, умноженный на фактор корректировки:

X = X запрограммированное + (R инструмента * u). (ось абсциссы);
Y = Y запрограммированное + (R инструмента * v). (ось ординаты);
Z = Z запрограммированное + (R инструмента * w). (ось, параллельная оси шпинделя).

Эти факторы корректировки могут быть использованы в случае простых профилей (контуры, параллельные оси) или в случае фрезерования трехмерных поверхностей. Факторы корректировки (u, v, w) не могут быть использованы в том случае, если имеется автоматическая корректировка радиуса инструмента (G41-G42) или язык GTL.

Что касается формата, то в том случае, если факторы отрицательны, после них ставится знак «-», знак «+» опускается. Для определения величины и знака «+» достаточно рассматривать вышеуказанные факторы как координаты X,Y,Z вершины профиля, данные в системе декартовых осей, которые параллельны осям станка и имеют начало в точке, для которой требуется корректировка.

Пример определения знака фактора корректировки приведён на рисунке 2.36.

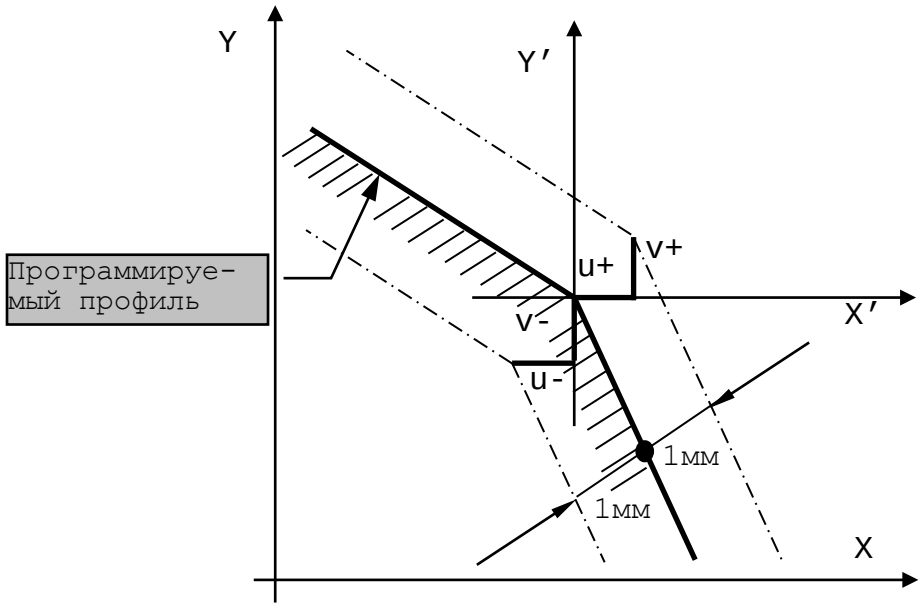


Рисунок 2.36

Система представляет возможным корректировать профили, которые могут быть смещены параллельно самим себе, увеличивая или уменьшая их координаты на величину, прямо пропорциональную величине фактора корректировки. Это происходит благодаря тому, что на станке эти изменения получены при умножении значений **u, v, w** на величину диаметра инструмента, находящуюся в памяти системы, разделенную на два.

Таким образом, можно выполнить корректировки профилей, состоящих из:

- отрезков прямой линии, параллельных осям или наклоненных к осям;
- отрезков прямой линии, касательных к дугам окружностей;
- дуг окружностей, касательных между собой, при условии, что смещенные параллельно самим себе, они будут являться касательными между собой.

Факторы **u, v, w** действительны только в том кадре, в котором они были запрограммированы и функционируют, если соответствуют следующим координатам:

u - к координате X (1 ось конфигурации или ее замена);
v - к координате Y (2 ось конфигурации или ее замена);
w - к координате Z (3 ось конфигурации или ее замена).

Примеры использования факторов корректировки u-v-w приведены на рисунках 2.37-2.42.

1)
```
N5  T1.01 M6 S.. F..
N6  G X Y30
N7  G1 Y10 v1
N8  X40 u-1
N9  Y30
N10 G X.. Y..
```

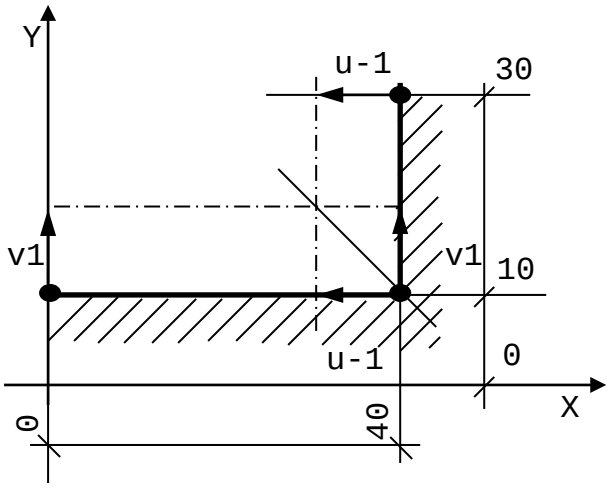
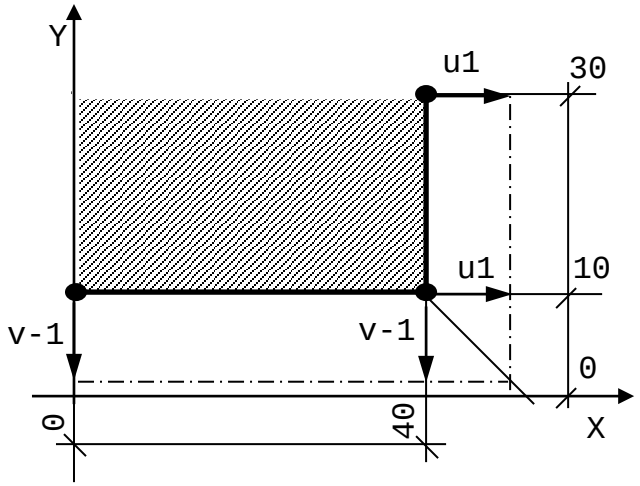


Рисунок 2.37



2)
```
N13 G X Y
N14 Y10 v-1
N15 X40 u1
N16 Y30
N17 G X.. Y..
```

Рисунок 2.38

3)
```
N13 G X Y
N14 G1 Z-10
N15 X-20 Y-20 u1 v1
N16 X20 u-1
N17 Y20 v-1
N18 X-20 u1
N19 Y-20 v1
N20 G X Y
```

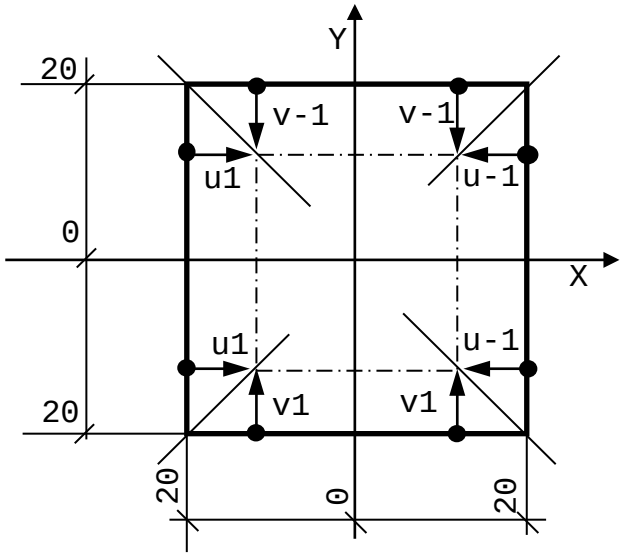


Рисунок 2.39

4)
```
N13 G X-35 Y-35
N14 Z-10
N15 G1 X-20 Y-20 u-1 v-1
N16 X20 u1
N17 Y20 v1
N18 X-20 u-1
N19 Y-20 v-1
N20 GZ
```

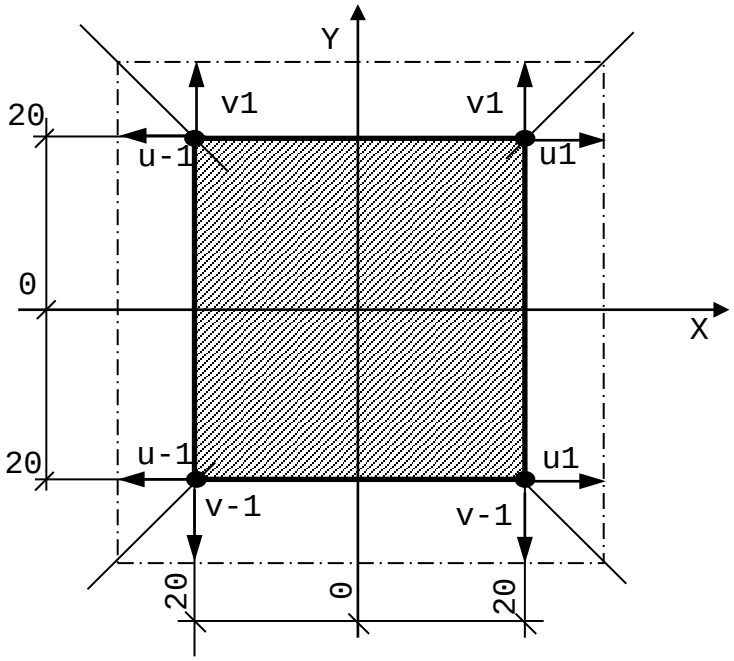


Рисунок 2.40

5)
```
N12.........
N13 G X-30 Y
N14 G1 Y20 v-1
N15 X30
N16 Y-20 v1
N17 X-30
N18.........
```

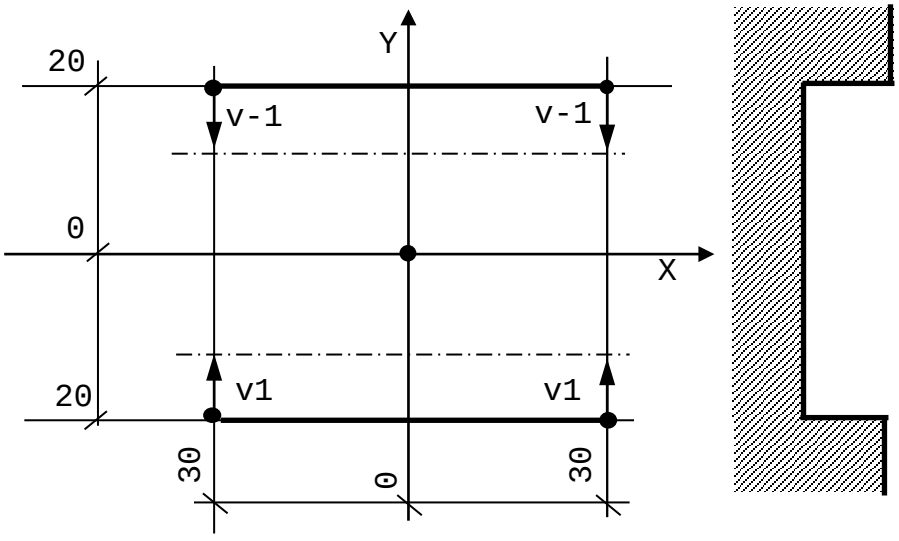


Рисунок 2.41

6) вращательные оси

```
N12 G X40 Y B0
N13 G1 X30 u1
N14 X25 B360 u1
N15 G X40
```

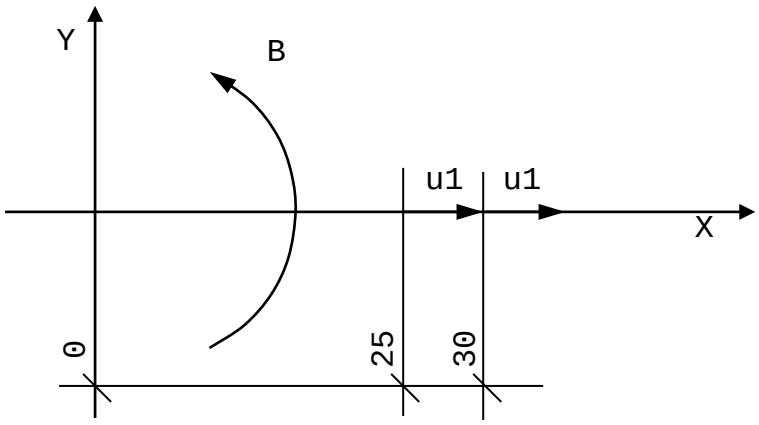


Рисунок 2.42

2.10 Кадры назначения глобальных переменных системы

Кадры назначения позволяют непосредственно из УП определить величины глобальных переменных.

Кадры назначения в зависимости от выходных переменных могут быть подразделены на 3 класса:

1. кадры назначения переменных через параметры (см. «Параметрическое программирование»);
2. кадры назначения геометрических переменных (см. «Геометрическое программирование высокого уровня»);
3. кадры назначения глобальных переменных системы.

В этом разделе рассматриваются кадры назначения глобальных переменных системы, которые обычно используются из программы (полный список кодов, назначаемых из программы или клавиатуры, приведён в таблице 17). Эти переменные, значения которых могут быть получены из программы, определяют параметры, используемые во время цикла обработки.

2.10.1 Определение выдержки времени - TMR

Глобальная переменная TMR позволяет назначить выдержку времени в конце кадра, а отработка этой паузы производится в кадрах с функциями G04 и/или в постоянных циклах и в токарных циклах нарезки пазов.

Формат:

TMR = ВЕЛИЧИНА ,

где:

ВЕЛИЧИНА - может быть запрограммировано явным и/или неявным (параметр **E** формата LR) образом.

При отработке G04 и/или в постоянных циклах при активной функции:

G94 - значение паузы, записанное в TMR, выражено в секундах (максимум 60 секунд);

G95 - значение паузы, записанное в TMR, выражено в количестве оборотов шпинделя (максимум 30000 оборотов).

Пример

```
TMR = 12.5          назначает выдержку времени, равную 12.5 с.

E32 = 13.4
TMR = E32           назначает выдержку времени, равную 13.4 с.
```

Примечание - TMR может присутствовать в любой части программы.

2.10.2 Определение припуска - UOV

Глобальная переменная UOV определяет величину припуска, который необходимо оставить вдоль профиля в операциях контурной обработки. Используется в циклах предварительной черновой обработки.

Формат:

UOV = ВЕЛИЧИНА ,

где:

ВЕЛИЧИНА - может быть запрограммирована явным или неявным (параметр **E** формата LR) образом и выражается в тех же единицах, что и размеры.

Пример

```
UOV = 0.5          назначает припуск, равный 0.5

E30 = 1.5
UOV = E30          назначает припуск, равный 1.5
```

Код UOV временно изменяет значение корректировок на величину, равную заданному значению. Отмена припуска программируется заданием кода UOV=0.

Примечание - Обычно переменная UOV программируется, но может быть использована в кадрах назначения, задаваемых с клавиатуры.

2.10.3 Определение переменной скорости возвращения при нарезании резьбы метчиком - RMS

Код RMS определяет процент изменения скорости вращения шпинделя при обратном движении инструмента в цикле нарезания резьбы в присутствии датчика на шпинделе по функции G84.

Формат:

RMS = ВЕЛИЧИНА ,

где:

ВЕЛИЧИНА - может быть в виде постоянной или параметра **E** формата BY.

Пример

```
RMS = 110 (+10% запрограммированного F)
RMS = 10 (-90% запрограммированного F)
```

Примечание - Обычно переменная RMS программируется, но может быть использована и в кадрах, задаваемых с клавиатуры.

2.10.4 Определение структуры пакета «А» и пакета «К» - SA, SK

Функции назначения переменных вх./вых. **SK**, **SA** позволяют установить значения непосредственно из УП в интерфейс PLC и предназначены для связи между УП и программой логики станка. Для этой цели в системе ЧПУ существует два типа структур данных, известных как пакет «А» и пакет «К».

Пакет «А» определяет все физические (электрические) сигналы типа ВКЛ./ВЫКЛ., соединяющие УЧПУ со станком. Эта структура находится в памяти и состоит из 1024 байтов.

Пакет «К» определяет все переменные связи между УП или действиями оператора с пульта оператора и программой логики станка. Для получения более обширной информации по структурам пакета «А» и пакета «К» необходимо обратиться к документу «Программирование интерфейса PLC».

Формат:

[индекс] = ВЕЛИЧИНА ,

где:

[индекс] - является величиной, определяющей переменную назначения, которой должно быть присвоено значение; ее значение строго зависит

от формата (формат, по умолчанию, бит - для SA и байт - для SK);

ВЕЛИЧИНА - может быть постоянной, символической переменной или последовательностью символов.

Примеры

1) SA12=SK800.BL - присваивает биту №12 структуры пакета «А» значение, содержащееся в восьмисотом бите структуры пакета «К».
2) SK5=SK7 - присваивает байту №5 структуры пакета «К» байт №7 этой же структуры.
3) SA128 = 1 - устанавливает (сигнал) бит №128 структуры пакета «А».
4) SK7.3CH="RIF" - пишет фразу RIF , начиная от байта №7 структуры пакета «К».
5) SA3.BY=255 - присваивает значение 255 байту №3 структуры пакета «А».

2.10.5 Определение группы переменных - SYVAR

Определяет группу переменных для программиста. Типами переменных, могут быть все те, которые предусмотрены для символических переменных языка УЧПУ.

Формат:

SYVAR [индекс] [формат] = ВЕЛИЧИНА ,

где:

[индекс] - значение, определяющее переменную, значение которой надо установить; ее значение зависит от формата (формат по умолчанию - байт);

ВЕЛИЧИНА - может быть постоянной, параметром **E**, переменной системы или последовательностью символов, если только совместимо с форматом переменной.

Пример

```
SYVAR = E4
SYVAR1 = E3+E4
SYVAR1.IN = 268
E4 = SYVAR
E35 = SYVAR2.LR
SYVAR16.3CH = "ABC"
```

По умолчанию можно определить 200 SYVAR формата байт.

Адресация глобальных переменных системы связана с двумя основными моментами:

1) индекс программирования переменной;
2) формат, в котором происходит обращение к переменной.

Форматы	Диапазон мин/макс
BL = 1/8 байта	0/1
BY = 1 байт	от 0 до 255
IN = 2 байта	от -32768 до +32768
LI = 4 байта	от -2.147.483.647 до +2.147.483.647
RE = 4 байта	+(-)7 значащих целых или десятичных чисел
LR = 8 байтов	+(-)16 - " - " - " -" - " -" - " -" - " -
CH = 1 байт для каждого	+(-)13 целых чисел символа

Адресации глобальных переменных для различных форматов приведены в таблице 2.7.

Примечания

1. При программировании индекса и формата переменной во избежание наложения памяти следует учитывать количество байтов, занятых предыдущей переменной.
2. Если формат переменной - CH, то число элементов, которое предшествует формату, указывает число символов, адресованных со стороны переменной. Значение по умолчанию - «1», максимальное значение - «32».

Примеры

1) SYVAR 1.4CH начиная с SYVAR 1, адресует четыре символа.

2)Адресация переменной SYVAR , начиная с SYVARO.BY:

```
SYVARO.BY
SYVAR1.CH
SIVAR1.IN
SIVAR1.LI
SIVAR2.RE
SYVAR2.LR
.........
```

Таблица 2.7 - Адресация глобальных переменных для различных форматов

<table>
<tr><td>S0.BL
S7.BL</td><td>S0.BY</td><td>S0.CH</td><td rowspan="2">S0.IN</td><td rowspan="4">S0.LI</td><td rowspan="4">S0.RE</td><td rowspan="8">S0.LR</td></tr>
<tr><td rowspan="15"></td><td>S1.BY</td><td>S1.CH</td></tr>
<tr><td>S2.BY</td><td>S2.CH</td><td rowspan="2">S1.IN</td></tr>
<tr><td>S3.BY</td><td>S3.CH</td></tr>
<tr><td>S4.BY</td><td>S4.CH</td><td rowspan="2">S2.IN</td><td rowspan="4">S1.LI</td><td rowspan="4">S1.RE</td></tr>
<tr><td>S5.BY</td><td>S5.CH</td></tr>
<tr><td>S6.BY</td><td>S6.CH</td><td rowspan="2">S3.IN</td></tr>
<tr><td>S7.BY</td><td>S7.CH</td></tr>
<tr><td>S8.BY</td><td>S8.CH</td><td rowspan="2">S4.IN</td><td rowspan="4">S2.LI</td><td rowspan="4">S2.RE</td><td rowspan="8">S1.LR</td></tr>
<tr><td>S9.BY</td><td>S9.CH</td></tr>
<tr><td>S10.BY</td><td>S19.CH</td><td rowspan="2">S5.IN</td></tr>
<tr><td>S11.BY</td><td>S11.CH</td></tr>
<tr><td>S12.BY</td><td>S12.CH</td><td rowspan="2">S6.IN</td><td rowspan="4">S3.LI</td><td rowspan="4">S3.RE</td></tr>
<tr><td>S13.BY</td><td>S13.CH</td></tr>
<tr><td>S14.BY</td><td>S14.CH</td><td rowspan="2">S7.IN</td></tr>
<tr><td>S120.BL
S127.BL</td><td>S15.BY</td><td>S15.CH</td></tr>
<tr><td>S128.BL</td><td></td><td></td><td></td><td></td><td></td><td>S2.LR</td></tr>
<tr><td colspan="7">Примечание - В таблице переменная S = SYVAR/SA/SK</td></tr>
</table>

2.10.6 Определение времени системы - TIM

Код TIM определяет группу переменных, используемых программистом для фиксации времени, в определенных моментах обработки на базе системного таймера. Эта группа имеет семь таймеров (от 0 до 6). Первый таймер (TIMO) зарезервирован и содержит таймер системы, который получает значение не за счет простого назначения, а при использовании трехбуквенного кода (TIM), введённого с клавиатуры. Единицей измерения этих таймеров является секунда.

Формат:

TIM [индекс] = ВЕЛИЧИНА ,

где

ВЕЛИЧИНА - может быть постоянной или параметром **E** (от E25 до E29).

Назначения переменным TIM могут быть запрограммированы только в кадре программы. Переменная TIM должна быть запрограммирована в кадре с GOO и нуждается в синхронизации (символ # перед кодом TIM).

Примеры

```
...........
N9 GO......
N10 # TIM1=TIMO      - придаёт TIM1 время системы (часы)
...........
...........
N89 GO.....
N90 # TIM2=TIMO-TIM1 - вычисляет время обработки от кадра N10 до кадра N90
...........
(DIS,TIM2)           - воспроизводит на видеокадре вычисленное время
```

2.10.7 Определение общего времени - TOT

Определяет группу переменных, используемых программистом для суммирования частичного времени циклов обработки, полученного в определенные моменты программы (разница между TIM0 и TIM, полученная в начале цикла обработки). Используя тот же номер в качестве индекса TIM и TOT, программист может использовать 6 переменных для получения времени и 6 переменных для суммирования частичных времен. Единственным допущенным форматом для размеров переменных является формат (RE) действительный.

Формат:

TOT [индекс] = ВЕЛИЧИНА ,

где:

ВЕЛИЧИНА - может быть постоянной или параметром **E** (от E25 до E29).

Пример

```
N10  E=75                    - количество деталей
N20 # TIM1=TIM0
N30  T2.2 M6
N40  XY S2000 F500 M13
.....
.....
N200 # TOT0=TIM0-TIM1
N210 TOT1=TOT0*E1
N220 TOT2=TOT1/3600
N230 (DIS,TOT2)              - указывает в часах время, необходимое для
                               выполнения партии из 75 деталей.
```

Примечания

1. Назначения переменной TOT могут быть запрограммированы только в кадре программы.
2. Переменная TOT нуждается в синхронизации (символ #).

2.10.8 Максимальная ошибка формы ERF

ERF определяет максимальную ошибку формы.

Формат:

ERF = ЗНАЧЕНИЕ ,

где:

ЗНАЧЕНИЕ - расстояние в дюймах или мм (используемая размерность может быть задана функциями G70 или G71).

Управление расчетом замедления на профиле выполняется при активной функции G27. По функции G27 движение осей на профиле замедляется до величины подачи, не позволяющей превысить предел отклонения от профиля, установленный в переменной ERF.

Эта переменная может быть определена в секции 4 файла PGCFIL. По умолчанию значение переменной равно 0 и может быть изменено из УП или с клавиатуры. Если ERF = 0, то оси будут замедляться до нуля в конце каждого кадра.

2.10.9 Максимальное отклонение направляющих косинусов MCD

Эта переменная определяет максимальный угол для осей, когда активна функция G27. Задаваемое значение находится в пределах от 0 до 2 и определяет максимальный угол на профиле от 0 до 180 градусов соответственно, после которого автоматическое управление замедлением по функции G27 не выполняется. Переменная MCD не влияет на режим динамики движения по функции G29.

Эта переменная может быть определена в секции 4 файла PGCFIL. По умолчанию значение переменной равно 0 и может быть изменено из УП или с клавиатуры.

Формат:

MCD = ЗНАЧЕНИЕ ,

где:

ЗНАЧЕНИЕ - значение изменяется от 0 до 2; значение определяет максимальный угол для осей, когда активна функция G27.

Если фунция **G27** активна, и максимальный разрешенный угол $\alpha \leq 90^0$, тогда необходимо записать в инструкцию, что MCD равно значению Sin α.

Если максимальный разрешённый угол $90^0 < \alpha \leq 180^0$, тогда необходимо записать в инструкцию, что MCD равно значению: 1 + [Sin(α-90^0)].

Пример

Если максимальный разрешенный угол равен 180^0, тогда инструкция должны быть определена следующим образом:

MCD = 1 + [Sin(α - 90^0)]
MCD = 1 + 1
MCD = 2

2.10.10 Остаток пути - RMN

При выполнении кадров УП по каждой интерполяционной оси ПрО формирует остаток пути, который индицируется в поле осей после выполнения команды UCV=3.

Эти остатки также записываются в переменную RMNpq, где «p» -цифра номера процесса, в котором определена ось (p = от 1 до 5); «q» -порядковый номер оси в интерполяторе скоординированных осей процесса «p».

Пример

E25=RMN12/20

Обычно переменную RMN используют для адаптивного контроля над основным процессом обработки. Для этого на уровне конфигурации системы должен быть сформирован дополнительный процесс, в котором загружается и циклически выполняется управляющая программа с алгоритмом воздействия на основной процесс обработки, адекватно результатам анализа остатка пути. Данная управляющая программа разрабатывается пользователем в соответствии с особенностью конкретного технологического процесса обработки. Разработчик адаптивной управляющей программы может устанавливать из неё через ПЛ процент подачи и другие сигналы для осей основного процесса обработки, а также с помощью кода DAW формировать напряжение максимум для шести свободных, не занятых управлением осями, каналов ЦАП.

2.10.11 Определение угла косоугольной системы реальных координат - UGF

Определяет угол косоугольной системы координат. Это является необходимым в случае, если на станке ось параллельная оси вращения шпинделя и ей поперечная образуют угол не равный 90 град. В этом случае при программировании используется код UAV=5, см. программирование косоугольной системы координат с помощью декартовых виртуальных координат.

Формат:

UGF = Значение ,

где:

Значение - определяет угол косоугольной системы координат (град.). Угол считается положительной величиной, если откладывается от положительного направления поперечной оси к положительному направлению продольной оси. Может быть константой или E-параметром такого же формата.

Пример

UGF = 20.5

E26=75
UGF=E26

2.11 Геометрическое программирование высокого уровня (GTL)

В системе УЧПУ представляется возможным в программе описать геометрический профиль в плоскости, используя не только стандартный язык программирования (G1-G2-G3), но и язык программирования высокого уровня GTL. Этот язык позволяет программировать профиль, состоящий из прямых и окружностей, используя только информацию, полученную с чертежа. Система сама вычисляет точки пересечения и точки касания геометрических элементов.

Язык программирования GTL и стандартный язык могут быть использованы одновременно в одной и той же программе, но не для одного и того же профиля. Геометрия GTL функционирует только при абсолютном программировании (G90).

2.11.1 Векторная геометрия

Определение профиля с использованием GTL основано на использовании четырёх типов геометрических элементов:

- точки начала отсчёта;
- точки;
- прямые;
- окружности.

Так как профиль определяется как геометрическими элементами, так и направлением, то для определения геометрических элементов в языке GTL используется особый тип геометрии - ВЕКТОРНАЯ ГЕОМЕТРИЯ. Для векторной геометрии определение элемента кроме параметров, необходимых для установления позиции в плоскости, требует также назначения направления движения.

Например, прямая линия проходит через точки **A** и **B**, как показано на рисунке 2.43, двигаясь от **A** к **B**, или прямая линия **l'**, лежащая на **l**, но проходящая от **B** к **A**.

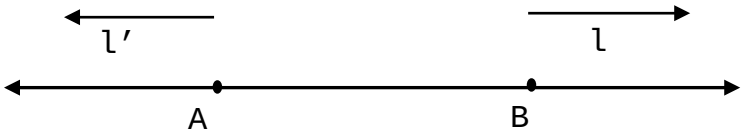


Рисунок 2.43

В векторной геометрии **l** и **l'** являются двумя различными линиями, имеющими противоположные направления. Программирование при помощи GTL, основанное на векторной геометрии, требует для каждой прямой линии назначения направления движения. Условимся, что направление движения прямой определяется углом, который она образует с положительной осью X. Правильный угол получается при вращении положительной оси X до наложения его на одну из прямых линий, которую надо определить. Угол будет иметь положительный знак, если ось X будет вращаться против часовой стрелки, и отрицательный - в обратном случае, как показано на рисунке 2.44.

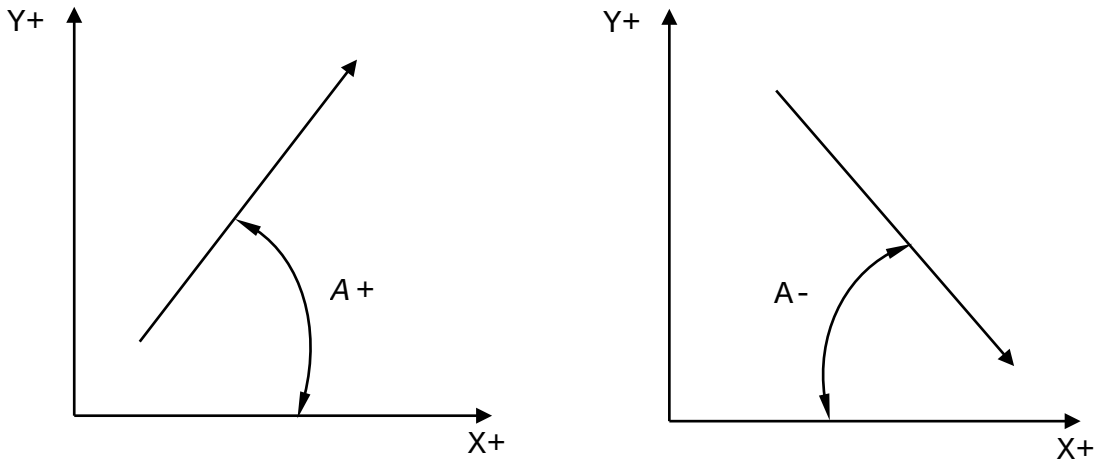


Рисунок 2.44

Направление должно быть придано также и окружностям. Условно принимается за положительное направление движение против часовой стрелки и за отрицательное - по часовой стрелке, как показано на рисунке 2.45.

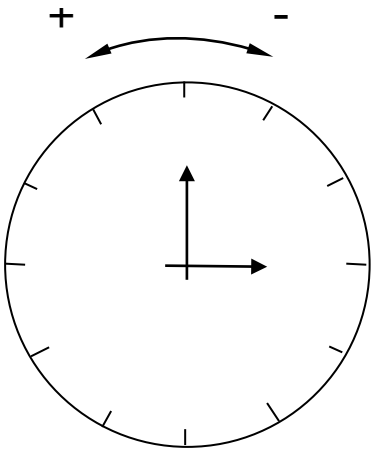


Рисунок 2.45

По договоренности придается положительное значение радиуса окружностям с направлением движения против часовой стрелки и отрицательное - в обратном случае, как показано на рисунке 2.46.

Направление, данное элементу, обычно соответствует направлению движения по профилю. Однако можно изменить направление элемента во время определения профиля, если это направление противоположно остальным элементам профиля.

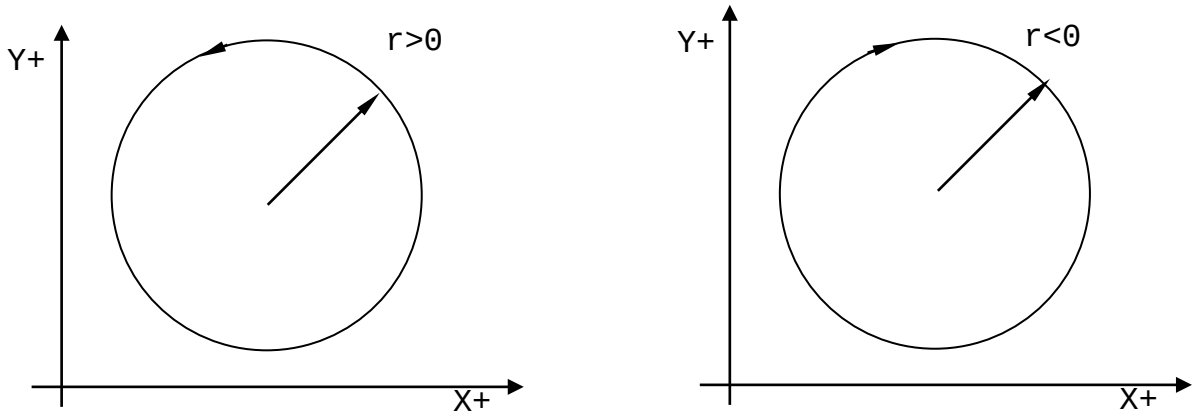


Рисунок 2.46

2.11.2 Хранение в памяти геометрических элементов

Хранение в памяти геометрических элементов предусматривает использование строчных символов **a-l-c-d-m-o-r-p-s-b** для определения соответственно:

- углов;
- прямых линий;
- окружностей;
- расстояний;
- модулей;
- точек начала отсчета;
- радиуса;
- номер пересечений (дискриминатор);
- скоса.

Необходимость использования для этой информации строчных символов вызвана тем, что эти же заглавные символы используются в языке УЧПУ для другой информации. Запоминание геометрических элементов в памяти осуществляется до определения профиля. Элементами, рассматриваемыми в GTL, являются прямые, окружности, точки, точки начала отсчета. Это - геометрические переменные, идентифицированные **НАЗВАНИЕМ** и **ИНДЕКСОМ**. Геометрическая переменная определяется в кадре назначения.

Формат:

НАЗВАНИЕ ИНДЕКС = <выражение> ,

где:

НАЗВАНИЕ - одно из четырёх символических названий, предусмотренных для геометрических элементов:

1) **o** - для определения точки начала отсчета;
2) **p** - для определения точки;
3) **l** - для определения прямой;
4) **c** - для определения окружности;

ИНДЕКС - определяет номер переменной геометрического элемента; этот номер заключён между 0 и 255, максимальный предел определяется при характеризации;

выражение - содержит всю информацию, необходимую для описания геометрического элемента; элементы могут быть определены:

- явным образом, программируя в кадре всю информацию, необходимую для определения геометрического элемента;
- неявным образом, вызывая другие геометрические элементы, определённые ранее.

Пример хранения в памяти элементов:

```
o1 = X30 Y30 a45
p1 = o1 X15 Y15
p2 = X60 Y30
l1 = p1, p2
l2 = X30 Y50, a45
c1 = l1, l2, r15
l3 = X0 Y0, X100 Y60
p3 = l3, c1
c2 = p3, r8
...........
```

Число геометрических элементов, хранимых в памяти, определяется на стадии характеризации системы. Формат геометрических определений предусматривает использование символа «,» (запятая) для отделения геометрического элемента (прямая - точка - окружность) от последующего геометрического элемента или информации (такой, как радиус «r» или угол «a»).

Примеры

```
p1 = X30 Y30
c1 = I10 J20 r30        разделитель не требуется
l1 = X20 Y20, X100 Y-10
       |         |
     точка     точка

l2 = I30 J20 r10, X80 Y80
         |           |
     окружность   точка

l3 = X100 Y100, a45
       |         |
     точка     угол

c3 = l1, l2, r18
      |   |   |
     |   |  радиус
     | прямая
    прямая
```

Дискриминатор s2 служит для выбора второго пересечения (s2). Иллюстрация приведена на рисунке 2.47.

p1=c1,l3

p2=c1,l3,s2

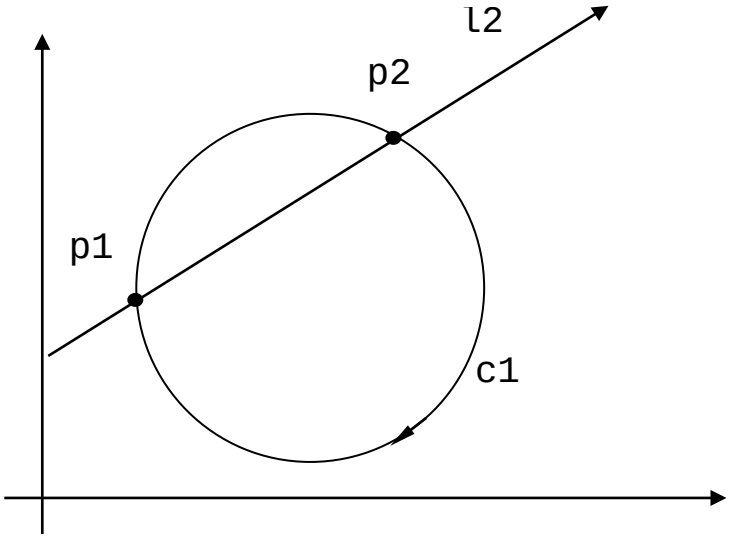


Рисунок 2.47

2.11.3 Список возможных геометрических определений

Точки начала отсчета:

pn= X.. Y.. a..

Точки:

pn = [om] X.. Y..
pn = [om] m.. a..
pn = lm, lp
pn = [-]lm, cp [,s2]
pn = cm, [-]lp [,s2]
pn = cm, cp [,s2]

Прямые линии:

ln = [om] X.. Y.., [op] X.. Y..
ln = [om] X.. Y.., a..
ln = [om] I.. J.. r.., [op] I.. J.. r..
ln = [om] I.. J.. r.., a..
ln = [om] I.. J.. r.., [op] X.. Y..
ln = [om] X.. Y.., [op] I.. J.. r..
ln = pm, pq
ln = pm, a..
ln = [-]cm, [-]cp
ln = [-]cm, a..
ln = [-]cp, pm
ln = pm, [-]cp
ln = [-]lm, d..

Окружности:

cn = [om] I.. J.. r..
cn = [om] m.. a.. r..
cn = [-]lm, [-]lp, r..
cn = [-]lm, [-]cp, r..
cn = [-]cp, [-]lm, r..
cn = pm, [-]lp, r..
cn = [-]lp, pm, r..
cn = [-]cm, [-]cp, r..
cn = pm, [-]cp, r..
cn = [-]cp, pm, r..
cn = pm, pq, r..
cn = pm, [-]lp
cn = pm, [-]cp [,s2]
cn = pm, pq, pr
cn = pm, r..
cn = [-]cm, [-]d..

Последовательность двух точек (..) указывает на необходимость объявления цифровых величин. Элементы в квадратных скобках - необязательные и могут быть опущены.

2.11.4 Определение точек начала отсчёта

Функция определения точек начала отсчёта дает возможность определить точки начала отсчёта в прямом формате (явным образом).

Обычно информация, находящаяся в программе, относится к системе осей, совпадающих с осями станка. Однако при проектировании деталь может быть выполнена на чертеже с использованием различных декартовых систем: абсолютной системы и других систем (начальных точек) отсчёта, которые могут быть приведены к абсолютной системе вращением и смещением осей. Геометрия GTL может быть определена при любой системе начала отсчёта.

Формат:

on = X.. Y.. a.. ,

где:

on - определяет название точки начала отсчёта;
X..Y.. - координаты новой начальной точки;
a.. - угол вращения (положительный против часовой стрелки).

Пример приведён на рисунке 2.48.

o4 = X20 Y30 a-45

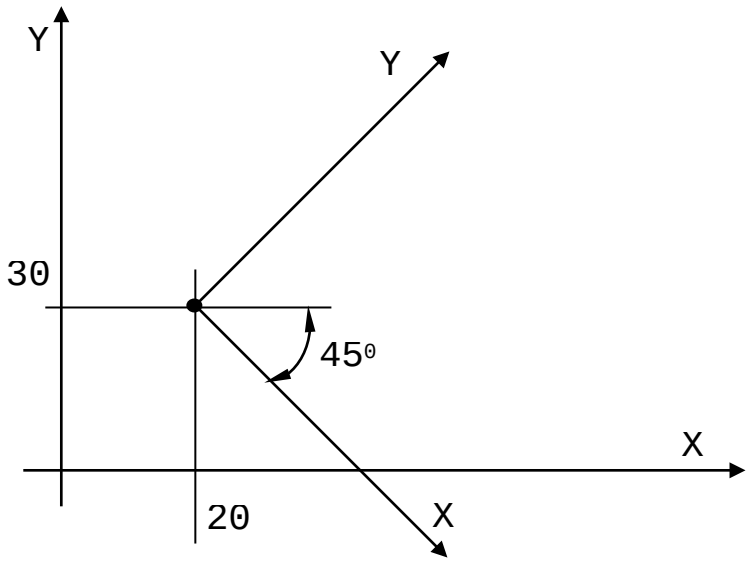


Рисунок 2.48

2.11.5 Определение точек

Функция определения точек позволяет определить точки в прямой (явной) форме или в косвенной (неявной) форме. Определение может быть дано как в декартовых координатах, так и в полярных.

Система полярного начала отсчета состоит из начальной точки, называемой полюсом, из которой начинается ось X, называемая полярной осью. Система полярного начала отсчёта приведена на рисунке 2.49.

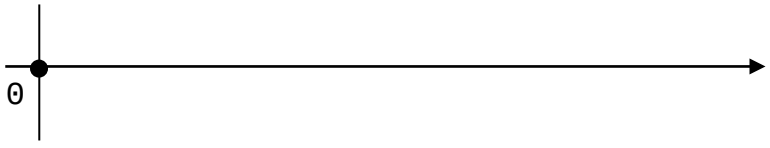


Рисунок 2.49 - Полярные оси

Любая точка плоскости может быть определена при помощи длины отрезка **p** (названной модулем), который соединяет ее с полярной точкой, и при помощи угла ϑ, который образуется отрезком прямой и полярной осью, как показано на рисунке 2.50.

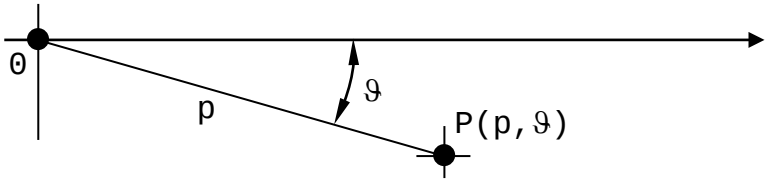


Рисунок 2.50 - Полярные координаты

Формат:

1) прямой:

- точка в декартовых координатах (примеры приведены на рисунках 2.51-2.52):

```
pn = [on] X.. Y.. ;
```

- точка в полярных координатах (пример приведён на рисунке 2.53):

```
pn = [on] m.. a.. ;
```

2) косвенный:

- точка пересечения двух прямых, определённых ранее (рисунок 2.54):

```
pn = lm,lp;
```

- точка пересечения прямой и окружности, определённая ранее (рисунок 2.55):

```
pn = [-] lm, cp [,s2]   ;
pn = cm, [-] lp [,s2]   ;
```

- точка пересечения двух окружностей (рисунок 2.56):

```
pn = cm, cp [,s2] ,
```

где:

- **Pn** - определяет название точки индекса n (n - число, заключённое между 1 и максимально конфигурируемым числом);
- **X.. Y..** - координаты точки;
- **[on]** - начальная точка отсчёта индекса n, определённая ранее, к которой относятся координаты X и Y;
- **m..** - модуль полярного вектора;
- **a..** - угол полярного вектора;
- **cm, cp** - ранее определённые элементы окружности индекса m и p;
- **[-] lm** - ранее определённые элементы прямой индекса m и p;
- **[-] lp** - можно изменить направление, вставляя знак «-»;
- **[,s2]** - индикатор второго пересечения.

В случае пересечения прямая-окружность или наоборот, существуют два возможных решения (рисунок 2.57): окружность c1 и прямая l2 пересекаются в точках **p1** и **p2**. Проходя прямую l2, следуя её направлению, сначала встречаем точку **p1** (1-е пересечение), а затем - точку **p2** (2-е пересечение). Для выбора второго пересечения (**p2**) следует использовать индикатор **s2**. Если он опущен, то выбирается первое пересечение (**p1**).

Пересечение прямая-окружность приведено на рисунке 2.55.

Также в случае пересечения окружность-окружность существуют два возможных решения: окружности **c1** и **c2** пересекаются в точках **p1** и **p2** (рисунок 2.58). Рассматривается сориентированная прямая, соединяющая центр 1-ой окружности с центром 2-ой окружности. Она делит плоскость на две полуплоскости. Для выбора точки в правой полуплоскости (**p2**) следует использовать индикатор **s2**. Если он опущен, то автоматически выбирается точка в левой полуплоскости (**p1**).

```
p1 = X30 Y160
```

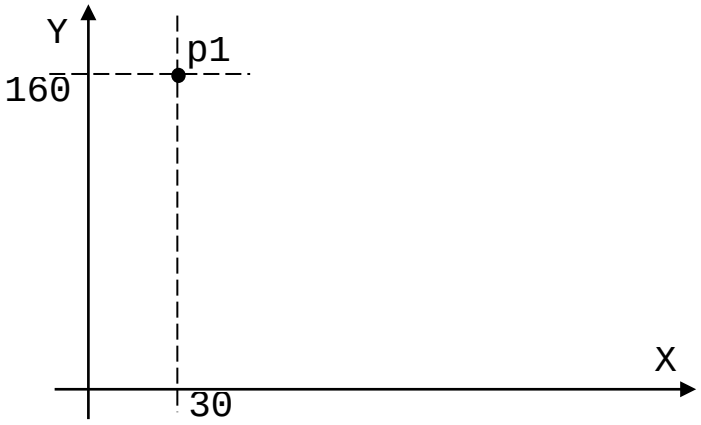


Рисунок 2.51

```
o1 = X30 Y30 a-20
p5 = o1 X20 Y10
```

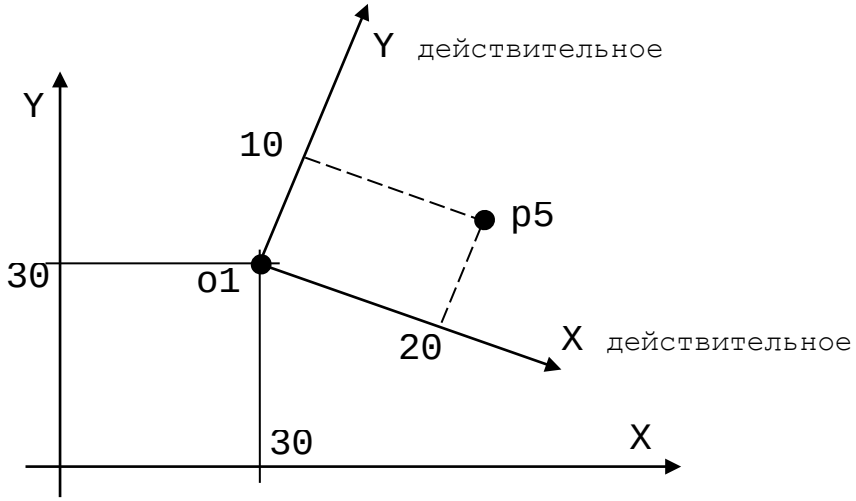


Рисунок 2.52

```
p2 = m55 a60
```

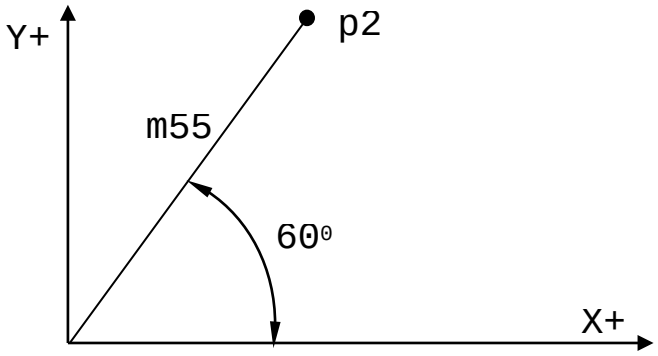


Рисунок 2.53

```
p1 = l1, l2
```

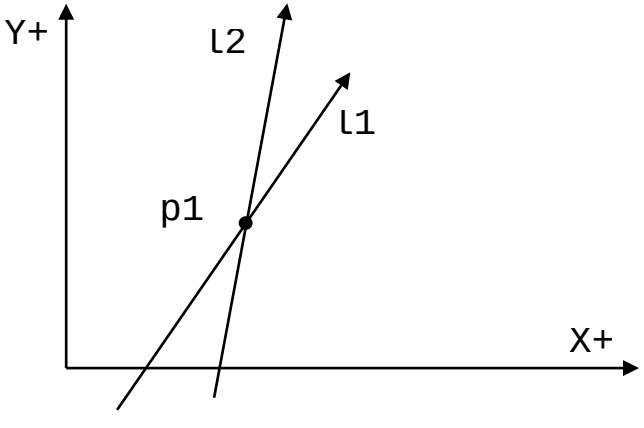


Рисунок 2.54

```
p1 = l4, c3
p2 = l4, c3, s2
p1 = -l4, c3, s2
```

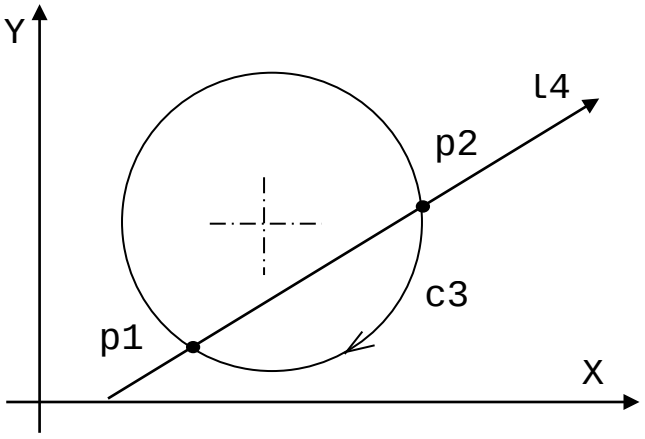


Рисунок 2.55

```
p1 = c1, c2
p2 = c1, c2, s2
p1 = c2, c1, s2
```

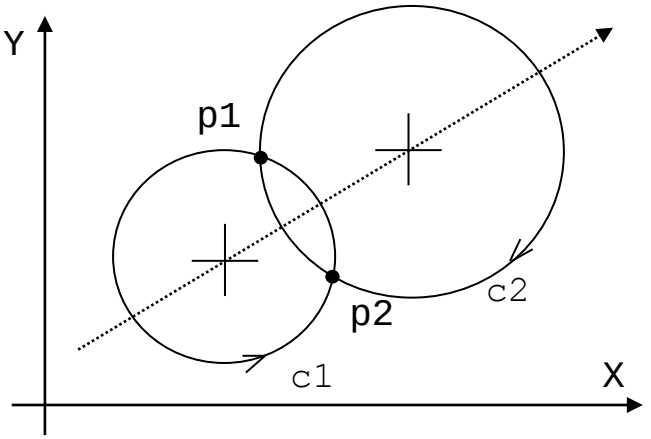


Рисунок 2.56

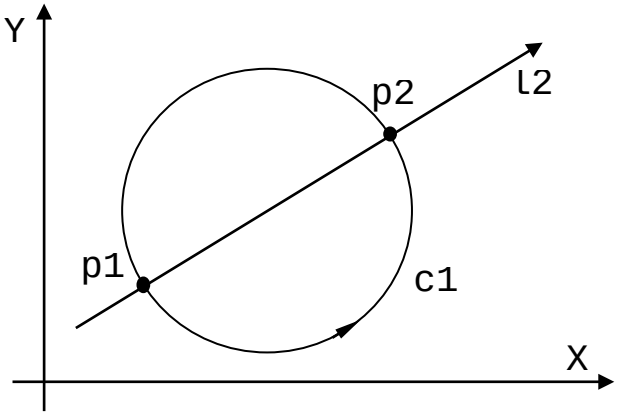


Рисунок 2.57

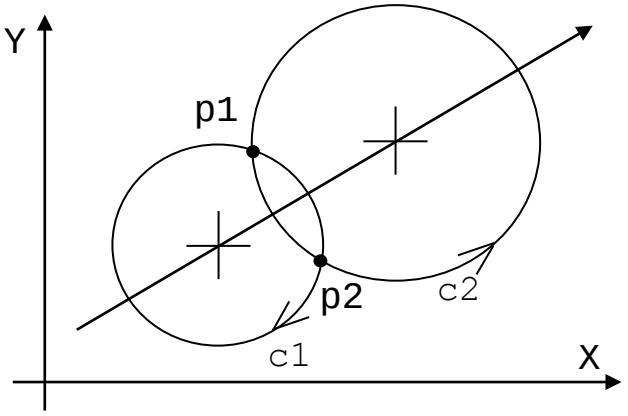


Рисунок 2.58

2.11.6 Определение прямой линии

Функция определения прямой линии позволяет определить прямую линию в прямой (явной) или косвенной (неявной) форме.

Направление прямой линии всегда от первого ко второму среди определяемых элементов. В случае если прямая касается с окружностью, возможны два решения, т.к. прямая может быть касательной с одной или с другой стороной окружности. Для выбора требуемого решения следует убедиться в том, что в точке касания окружность и прямая имеют одно и то же направление.

Несовместимость направлений движения геометрических элементов приведена на рисунке 2.59. Совместимость направления движения приведена на рисунке 2.60.

Формат:

1) прямая форма программирования прямых линий:

- прямая, проходящая через две точки (рисунок 2.61):

```
ln = [om] X..Y.., [op] X..Y.. ;
```

- прямая, проходящая через одну точку и образующая угол с осью абсциссы (рисунки 2.63, 2.64):

```
ln = [on] X..Y..,a..    ;
```

- прямая, касательная к одной окружности и образующая угол с осью абсциссы (рисунки 2.65, 2.66):

```
ln = [om] I..J.. r..,a..      ;
```

- прямая, касательная к двум окружностям (рисунки 2.67, 2.68):

```
ln = om I..J.. r..,  op I.. J.. r.. ;
```

- прямая, касательная к окружности и проходящая через точку (рисунок 2.62):

```
ln = [om] I..J..r.., [op] X..Y..    ;
ln = [om] X..Y.., [op] I.. J..r..   ;
```

2) косвенная форма программирования прямых линий:

- прямая, проходящая через две точки (рисунок 2.69):

```
ln = pm, pq ;
```

- прямая, проходящая через точку и образующая угол с осью абсциссы (рисунок 2.73):

```
ln = pm, a..      ;
```

- прямая, касательная к двум окружностям (рисунки 2.71, 2.72):

```
ln = [-]cm, [-] cp      ;
```

- прямая, касательная к одной окружности и образующая угол с осью абсциссы (рисунок 2.74):

```
ln = [-] cm, a..  ;
```

- прямая, касательная к окружности и проходящая через точку (рисунок 2.70):

```
ln = [-] cp,pm    ;
ln = pm, [-] cp   ;
```

- прямая, параллельная прямой на расстоянии **d** (рисунки 2.75, 2.76):

```
ln = [-] lm, d..  ,
```

где:

Ln	- определяет название прямой с индексом n (n - число, заключённое между первым и максимально конфигурируемым номером);
X..Y..	- координаты точки;
a..	- угол, образованный осью абсциссы и прямой (положительный - против часовой стрелки);
r..	- радиус окружности (положительный - против часовой стрелки);
pm pg	- предопределённые элементы точки с индексми **m** и **q**;
[-] cm [-] cp	- предопределённые элементы окружности с индексами **m** и **q**.

Направление движения окружности может быть изменено при помощи отрицательного знака для гарантирования совместимости направлений прямой и окружности в точке касания.

[-] lm	- предопределенный элемент прямой с индексом m;
d..	- расстояние между двумя прямыми; положительное, если прямая - слева, отрицательное - в обратном случае, глядя в направлении, предопределенном прямой.

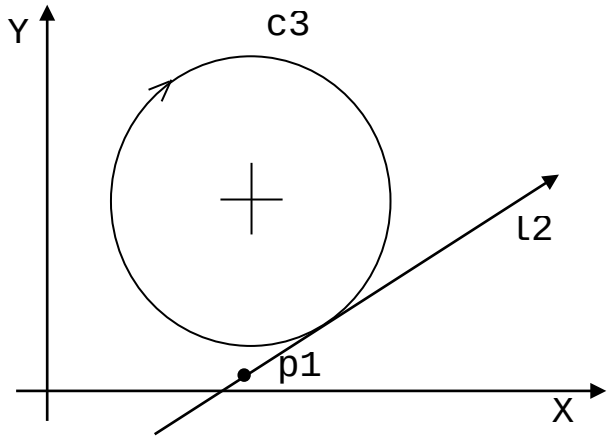


Рисунок 2.59

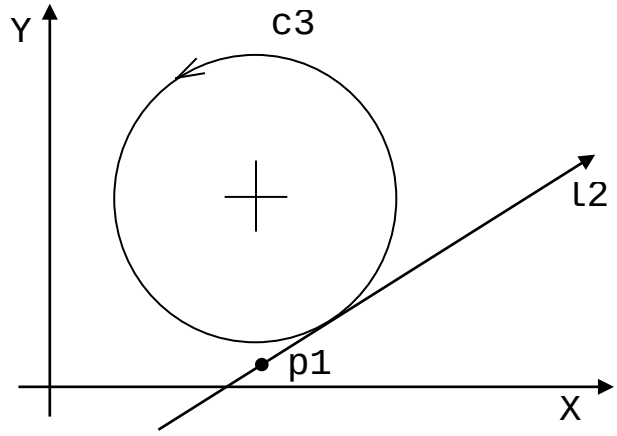


Рисунок 2.60

```
l1 = X40 Y20, X60 Y70
```

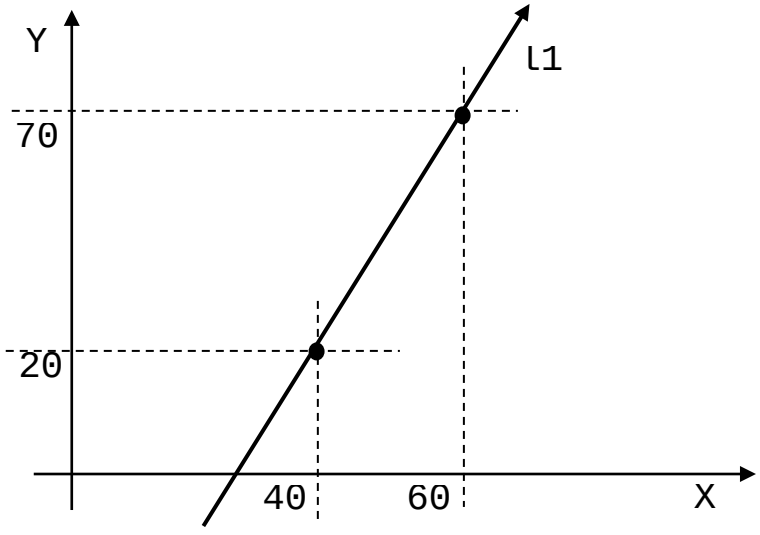


Рисунок 2.61

```
l1 = X10 Y15, I45 J30 r-15
l2 = X10 Y15, I45 J30 r15
```

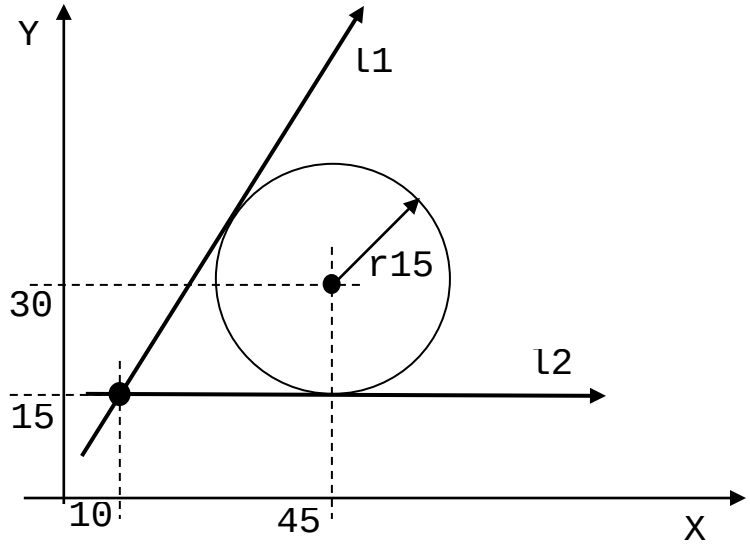


Рисунок 2.62

```
l2 = X20 Y20, a-20
```

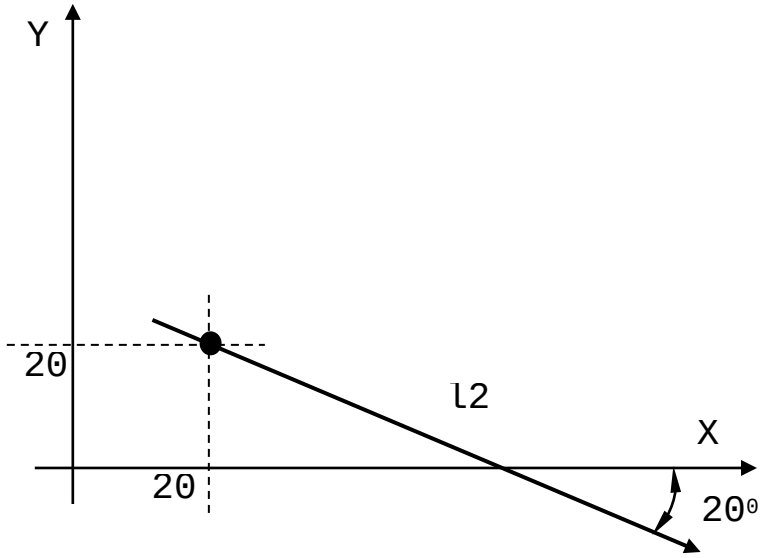


Рисунок 2.63

```
o1 = X30 Y30 a-40
l5 = o1 X25 Y30, a60
```

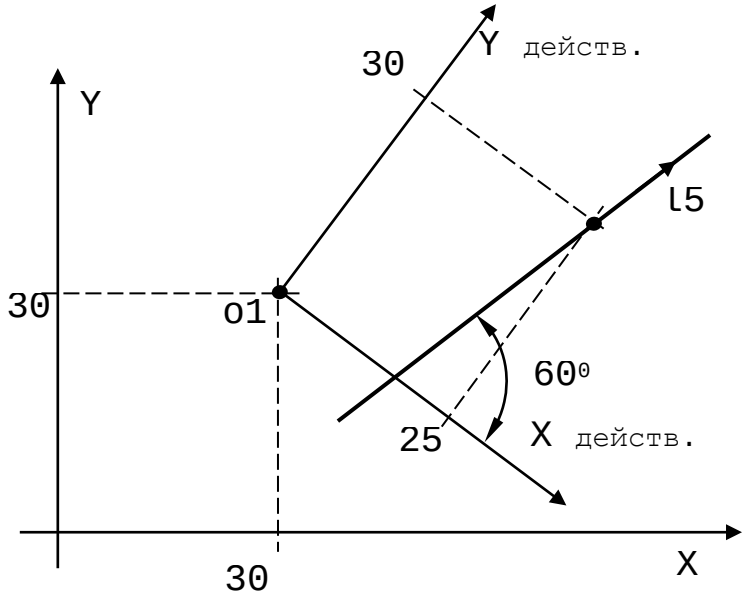


Рисунок 2.64

```
l1 = I60 J80 r40, a45
```

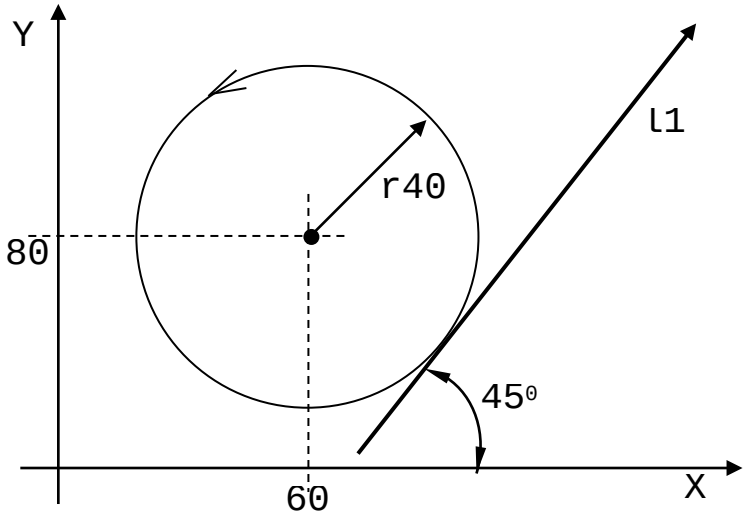


Рисунок 2.65

```
o3 = X25 Y10 a20
l4 = o3 I25 J15 r10, a115
```

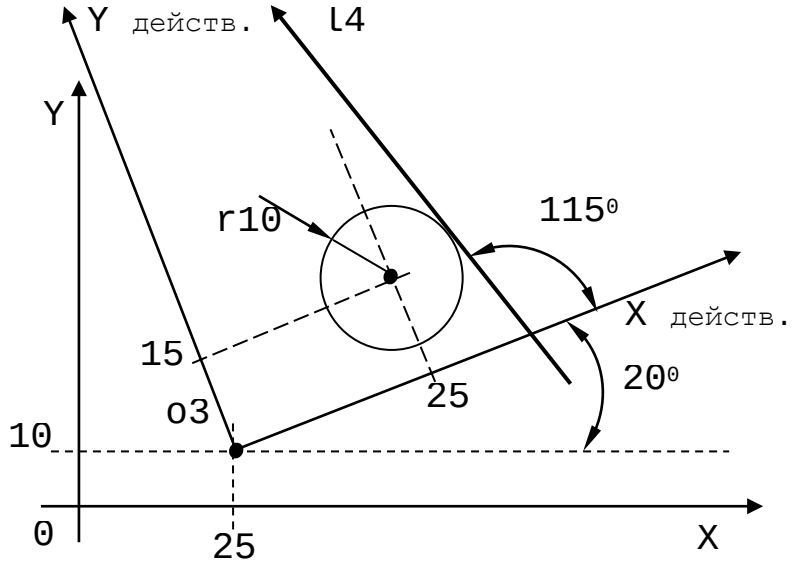


Рисунок 2.66

```
l3 = I25 J35 r-17, I70 J20 r13
```

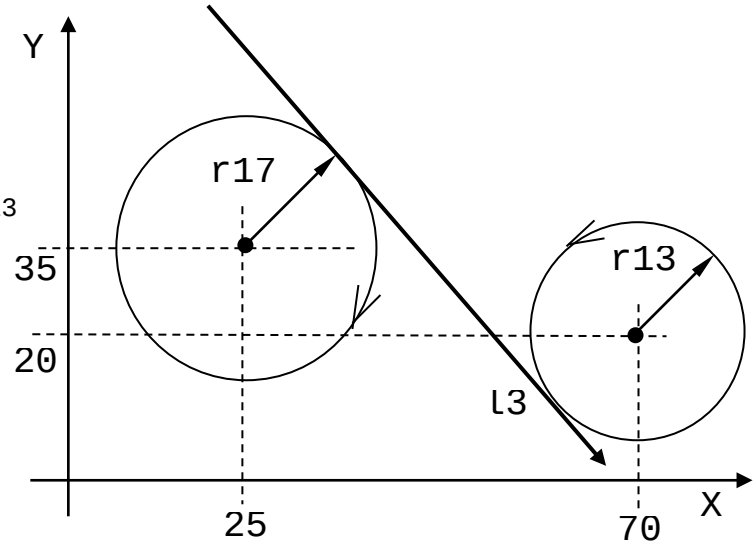


Рисунок 2.67

```
l4 = I25 J35 r17, I70 J20 r13
```

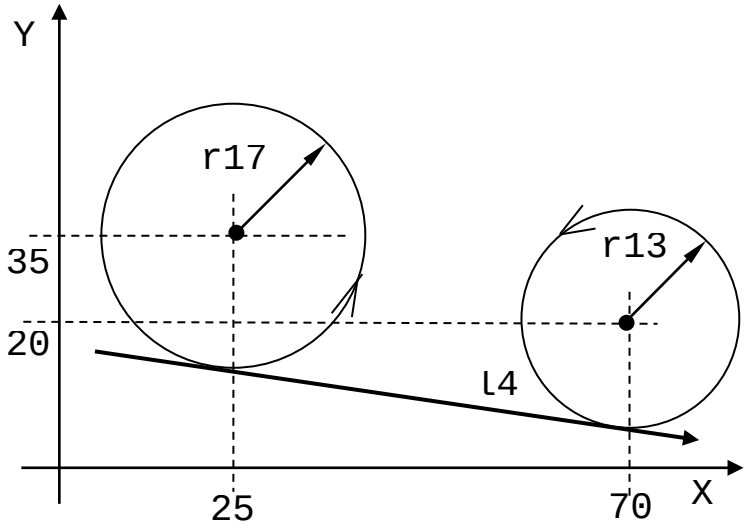


Рисунок 2.68

```
l9 = p7, p8
```

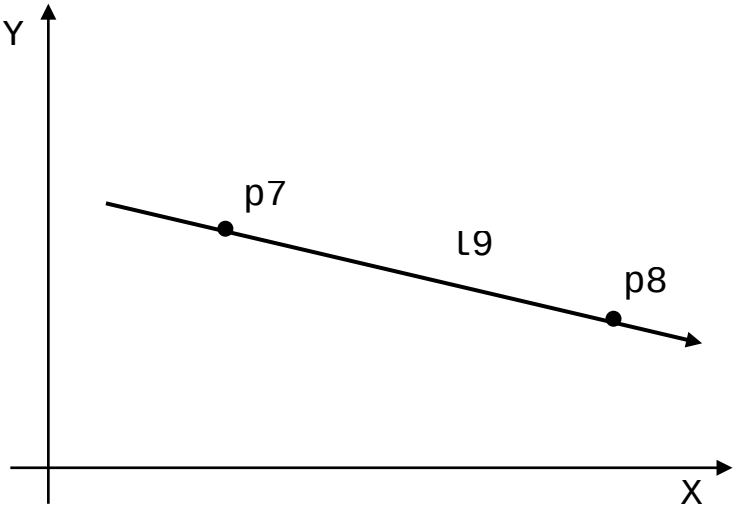


Рисунок 2.69

```
l1 = p1, c1
l2 = p1, -c1
```

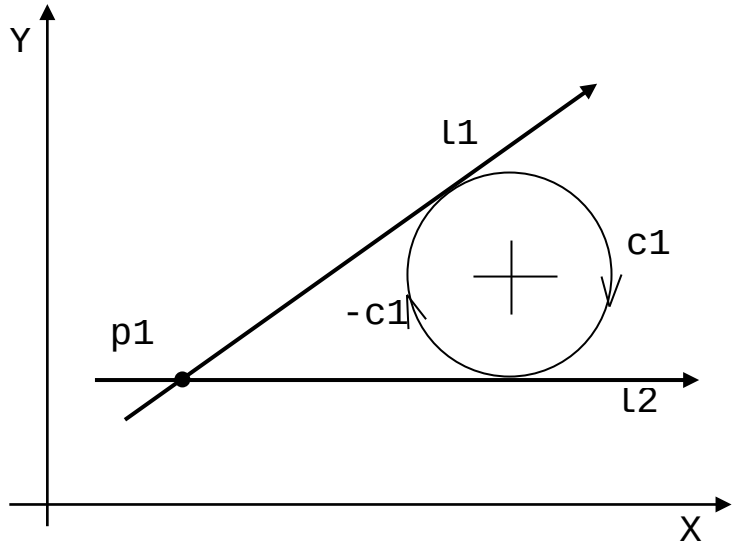


Рисунок 2.70

```
l3 = c1, c2
```

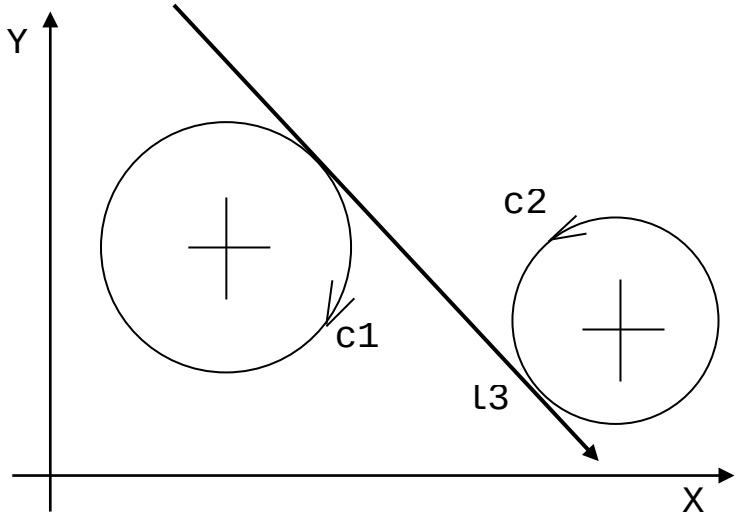


Рисунок 2.71

```
l4 = -c1, c2
```

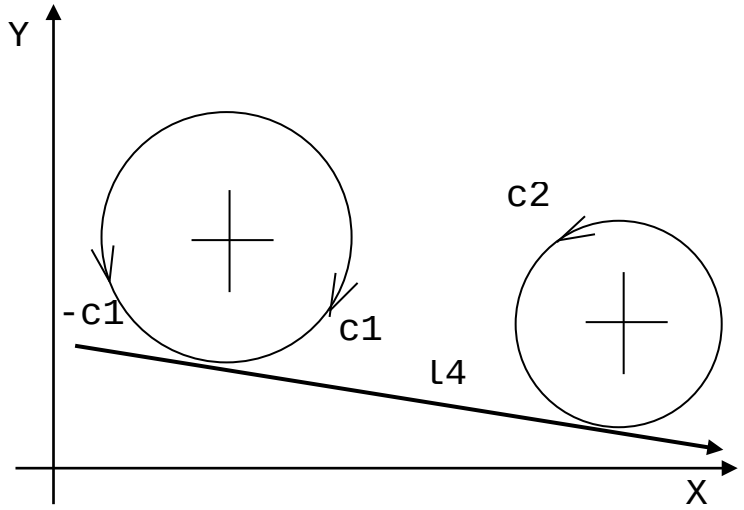


Рисунок 2.72

```
l3 = p1, a45
```

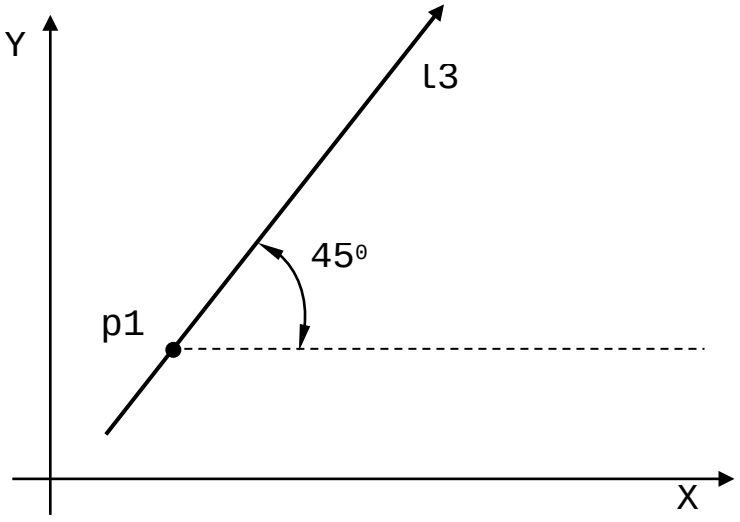


Рисунок 2.73

```
l1 = c1, a50
l2 = -c1, a50
```

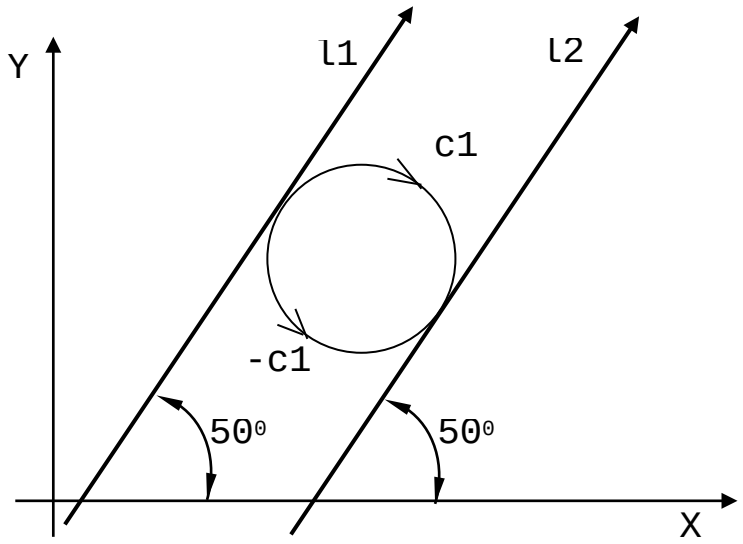


Рисунок 2.74

```
l2 = l1, d20
l3 = l1, d-15
```

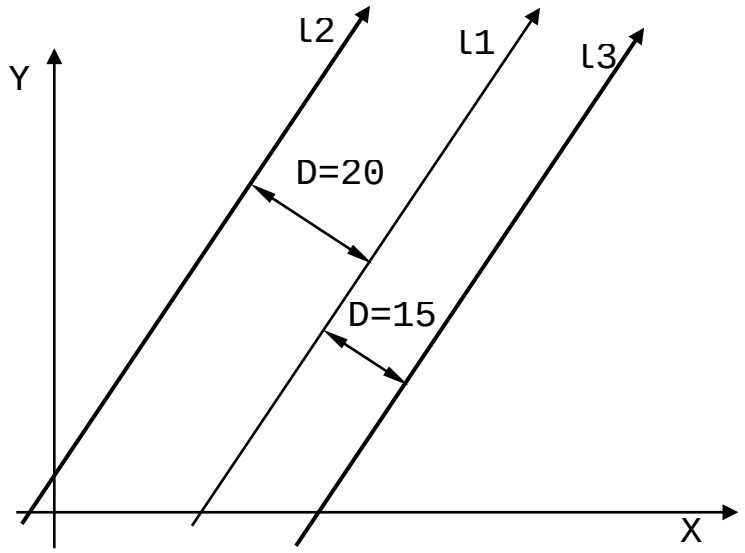


Рисунок 2.75

```
l2 = -l1, d-50
```

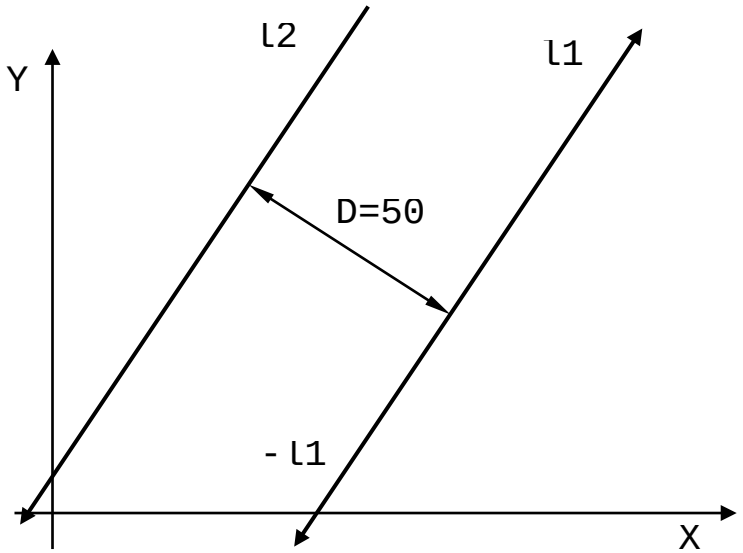


Рисунок 2.76

2.11.7 Определение окружностей

Язык GTL позволяет определить окружности в прямой форме (явной) или в косвенной (неявной) форме. Определяя окружности в косвенной форме, программист должен учитывать совместимость направлений элементов (знак «-» может изменить направление предопределенных элементов). Если не учитывается направление элементов, то задавая окружность известного радиуса и прямую линию, можно получить от 1 до 8 вариантов окружности, касательной к двум элементам. Возможные варианты окружности, касательной к прямой и окружности приведены на рисунке 2.77.

Если учитывать совместимость направлений движения предопределённых элементов и окружности, которую следует определить, количество возможных решений можно уменьшить до двух.

Чтобы различить две возможные окружности, имеющие одно и то же направление и один и тот же радиус, необходимо учитывать последовательность элементов:

- прямая – окружность (окружность – прямая);
- значение центрального угла двух возможных тангенциальных окружностей.

GTL всегда создает окружность с направлением от первого ко второму элементу и дугой, имеющей меньший центральный угол. Окружности, касательные к меньшей дуге, показаны на рисунке 2.78. Окружность c3 получается при определении прямой l1 в первой позиции и окружности c2 - во второй, т.к. элемент c3 позволяет движение от прямой l1 к окружности c2 с дугой, имеющей меньший центральный угол. Окружность c4 получается при определении окружности c2 в первой позиции и прямой l1 во второй, т.к. элемент c4 позволяет движение от окружности c2 к прямой l1 с дугой, имеющей меньший центральный угол. То же самое можно сказать для определения окружности, касательной к двум предопределенным окружностям. В этом случае можно получить от 1 до 8 решений в соответствии с рисунком 2.79. Окружности, касательные к двум предопределенным окружностям, показаны на рисунке 2.80.

Совместимость направлений движения предопределенных элементов и окружности, которую надо определить, сводит количество возможных решений до двух. Для различения двух окружностей, имеющих одинаковое направление и одинаковый радиус, рассматриваются две дуги, в которых новый элемент разделен точками, касательными к элементам начала. GTL создает окружность, двигаясь от первой окружности ко второй, с дугой, имеющей меньший центральный угол (рисунок 2.80). Для получения окружности c3 при определении следует установить в правой позиции элемент c1, а затем c2. Для получения окружности c4 элемент c2 должен предшествовать элементу c1 в определении.

2.11.7.1 Формат прямого программирования

Формат прямого программирования:

- окружность с декартовыми координатами центра и радиуса (рисунки 2.81, 2.82):

  cn = [om] I.. J.. r.. ,

- окружность с полярными координатами центра и радиуса (рисунок 2.83):

  cn = [om] m.. a.. r. ,

где:

cn	- устанавливает название окружности индекса n (n - номер, заключённый между первым и максимально конфигурируемым номером);
I..J..	- координаты центра окружности;
r..	- радиус окружности (положительный - для направления против часовой стрелки, отрицательный - для направления по часовой стрелке);
[-] lm	- предопределённые элементы прямой с индексом **m** и **p**;
[-] lp	- может принять противоположное направление при использовании знака «-»;
pm pq pr	- предопределённые элементы точки индекса **m**, **q**, **r**;
[-] cm [-] cp	- предопределённые элементы окружности с индексом **mp**; могут принять противоположное направление при использовании знака «-»;
[s2]	- атрибут для наибольшей из двух возможных окружностей;

d.. - расстояние между двумя окружностями; положительное, если, глядя из [-] **cm**, **cn** находится слева от неё, и отрицательное, если находится справа.

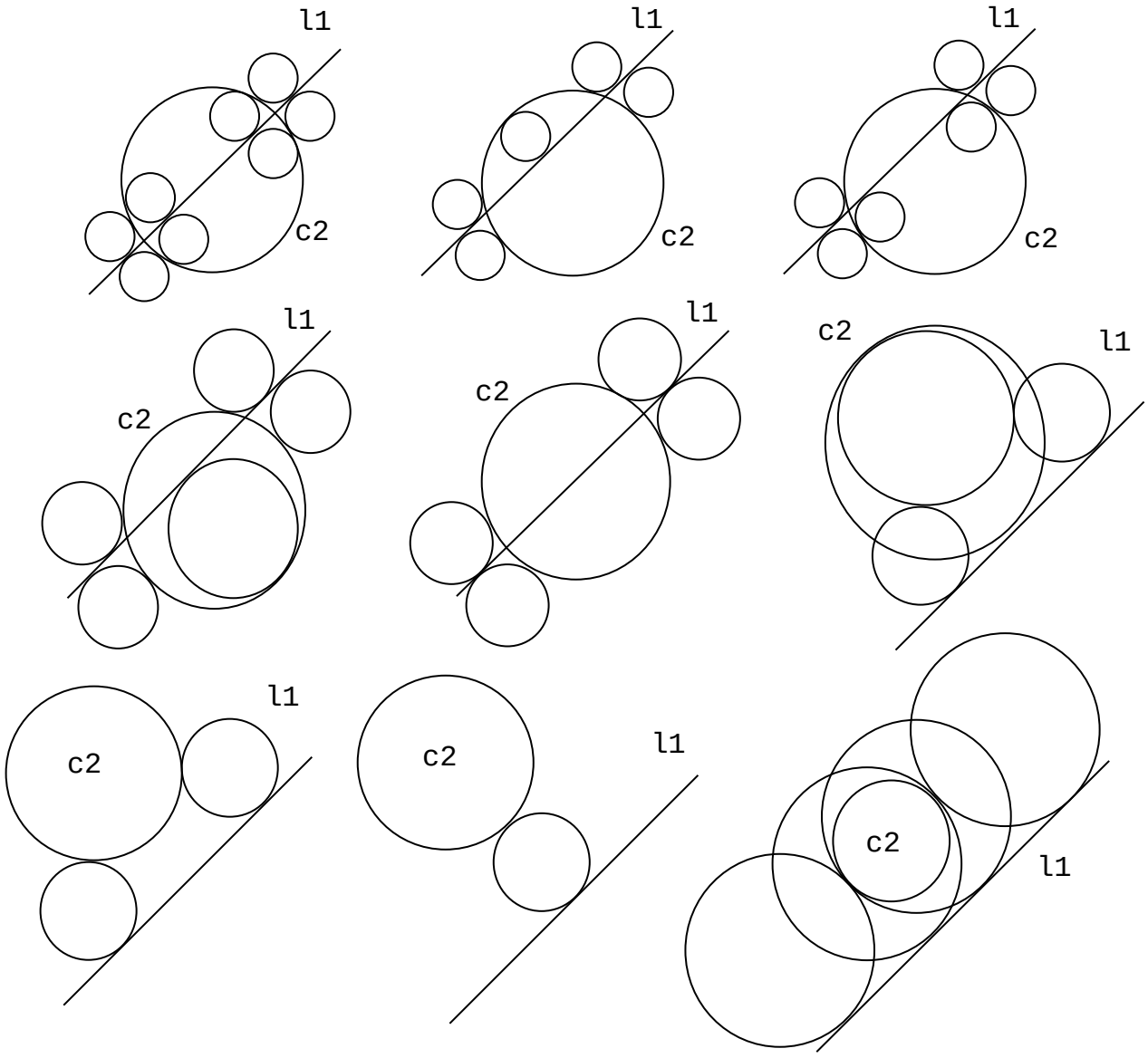


Рисунок 2.77

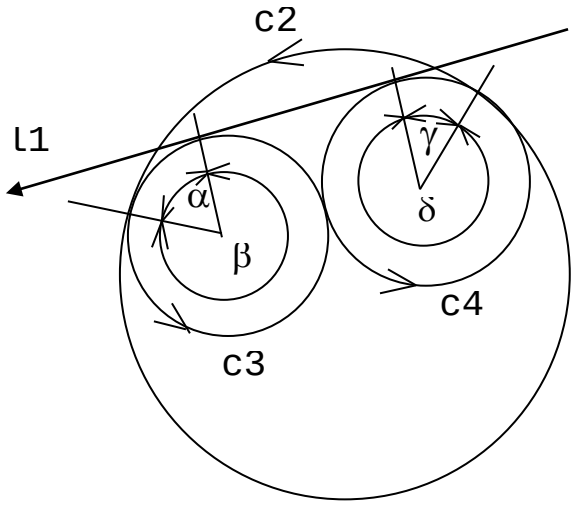


Рисунок 2.78

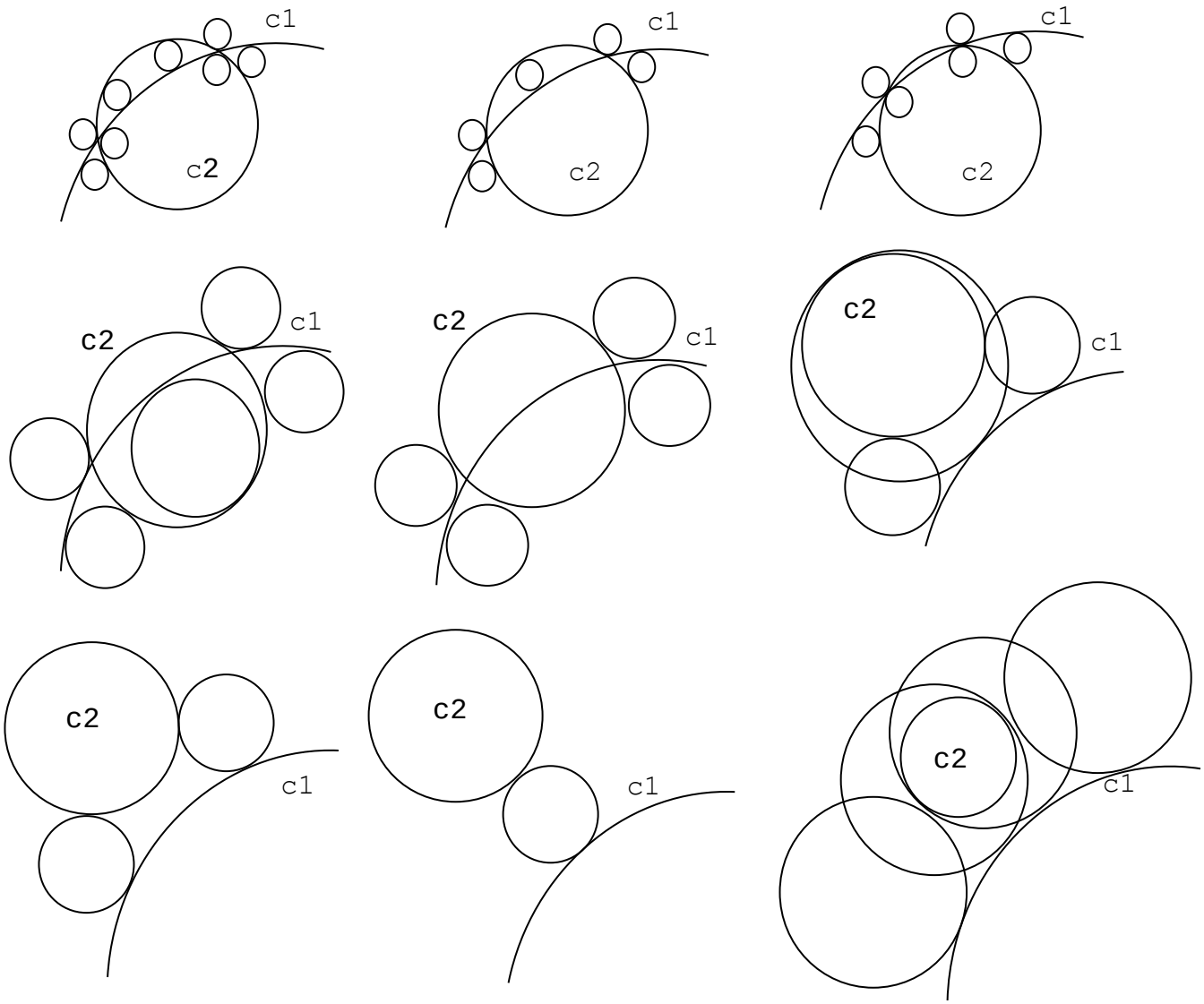


Рисунок 2.79

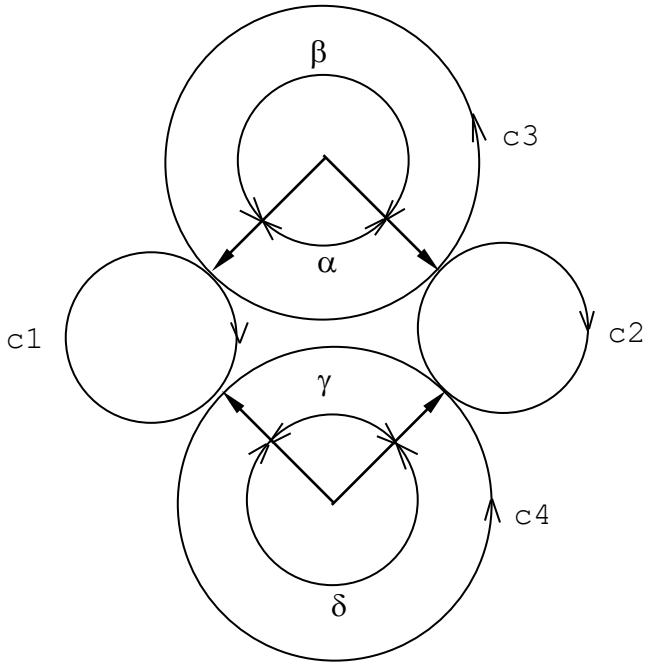


Рисунок 2.80

```
c2 = I50 J100 r-40
```

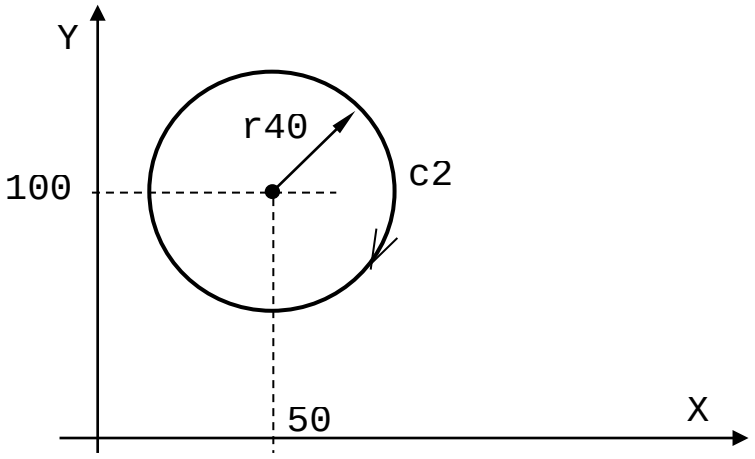


Рисунок 2.81

```
c1 = o1 I20 J20 r-15
```

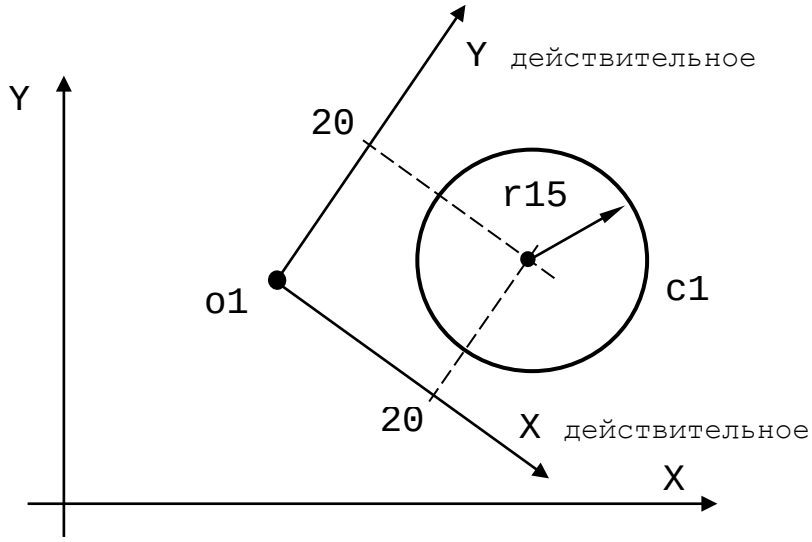


Рисунок 2.82

```
c2 = m70 a30 r15
```

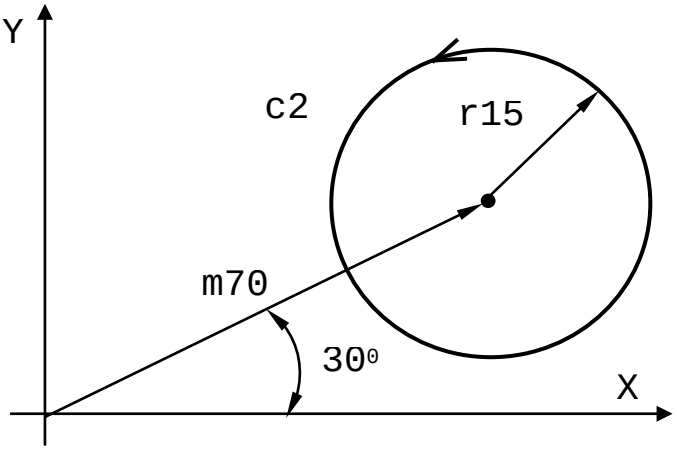


Рисунок 2.83

2.11.7.2 Формат косвенного программирования

Формат косвенного программирования:

- окружность с данным радиусом и касательная к двум предопределённым прямым (рисунок 2.84):

  cn = [-] lm, lp, r.. ;

- окружность с данным радиусом и касательная к прямой и окружности (предопределённым) (рисунки 2.85-2.87):

  cn = [-] lm, [-] cp, r.. ,
  cn = [-] cp, [-] lm, r.. ;

- окружность с данным радиусом, проходящая через предопределённую точку, и касательная к предопределённой линии (рисунок 2.88);

  cn = pm, [-] lp, r.. ,
  cn = [-] lp,pm, r.. ;

- окружность с данным радиусом и касательная к двум предопределённым окружностям (рисунки 2.89-2.90):

  cn = [-] cm, [-] cp, r.. ;

- окружность с данным радиусом, проходящая через одну предопределённую точку, и касательная к предопределённой окружности (рисунок 2.91):

  cn = pm, [-] cp, r.. ,
  cn = [-] cp,pm, r.. ;

- окружность с данным радиусом, проходящая через две предопределённые точки (рисунок 2.92):

  cn = pm, pq, r.. ;

- окружность с центром в предопределенной точке и касательная к предопределённой прямой (рисунок 2.93):

  cn = pm, [-] lp ;

- окружность с центром в предопределенной точке и касательная к предопределённой окружности (рисунок 2.94):

  cn = pm, [-] cp [,s2] ;

- окружность, проходящая через три точки (рисунок 2.95):

  cn = pm,pq,pr ;

- окружность с данным радиусом и с центром в точке (рисунок 2.96):

  cn = pm,r.. ;

- окружность концентрическая к предопределённой окружности и отдалённая от неё на данную величину (рисунок 2.97):

  cn = [-] cm,d.. ,

где:

cn - устанавливает название окружности индекса **n** (n - номер, заключённый между первым и максимально конфигурируемым номером);

I..J.. - координаты центра окружности;

r.. - радиус окружности (положительный для направления против часовой стрелки, отрицательный для направления по часовой стрелке);

[-] lm - предопределённые элементы прямой с индексом **mp;**
[-] lp - может принять противоположное направление при использовании знака «-»;
pm pq pr - предопределённые элементы точки индекса **mqr;**
[-] cm [-] cp - предопределенные элементы окружности с индексом **mp;** могут принять противоположное направление при использовании знака «-»;
[s2] - атрибут для наибольшей из двух возможных окружностей;
d.. - расстояние между двумя окружностями; положительное, если, глядя из [-] cm, cn находится слева от нее, отрицательное, если находится справа.

c3 = l1, l2, r-15

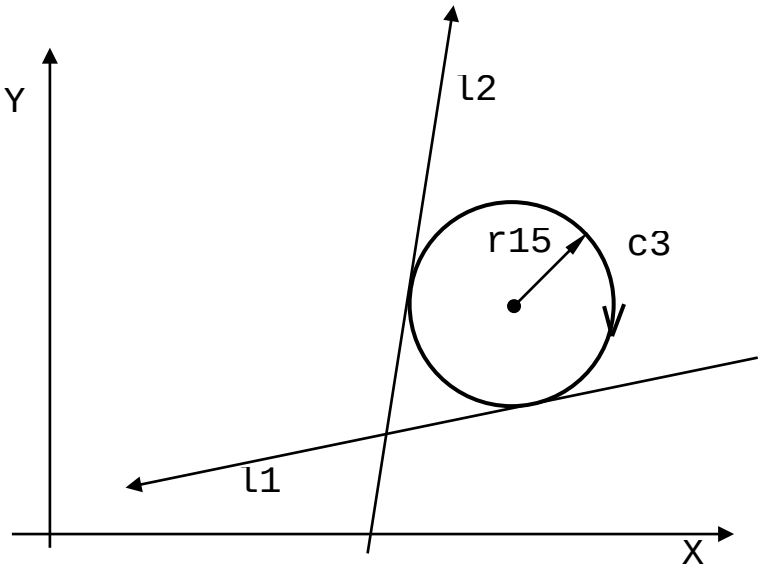


Рисунок 2.84

c3 = l1, -c2, r8

c4 = -c2, l1, r8

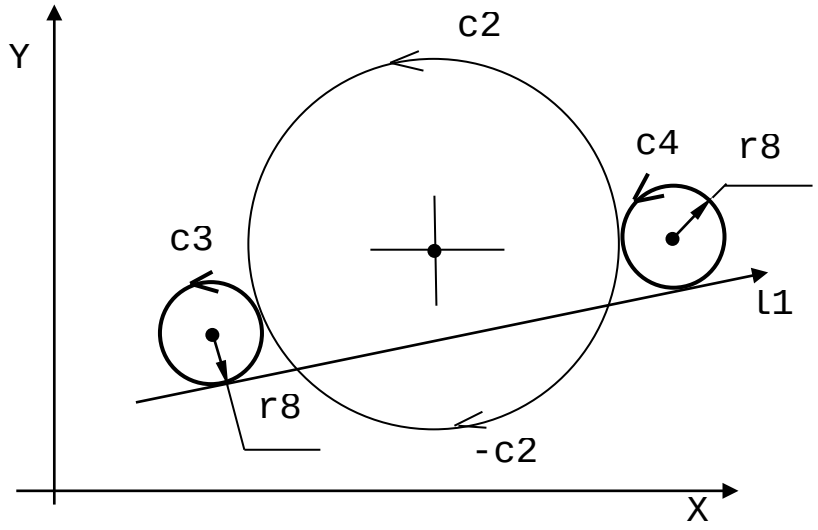


Рисунок 2.85

c9 = -c2, l1, r-8

c10 = l1, -c2, r-8

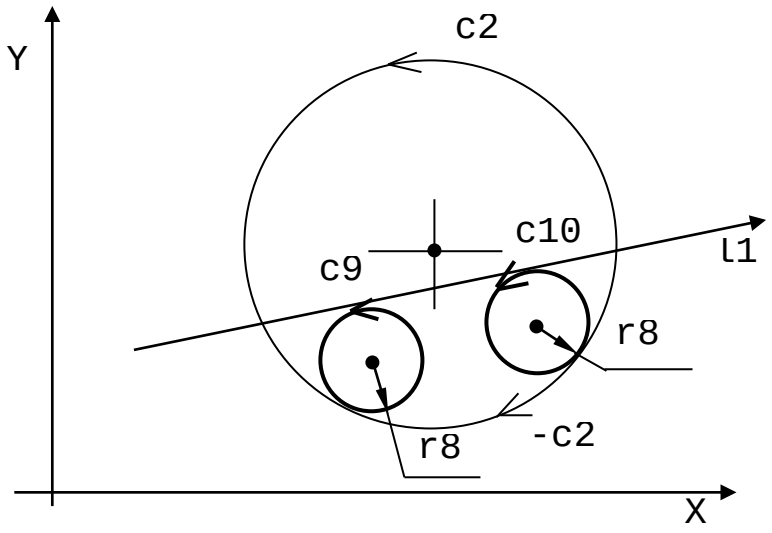


Рисунок 2.86

c4 = -l2, c1, r-40

c5 = c1, -l2, r-40

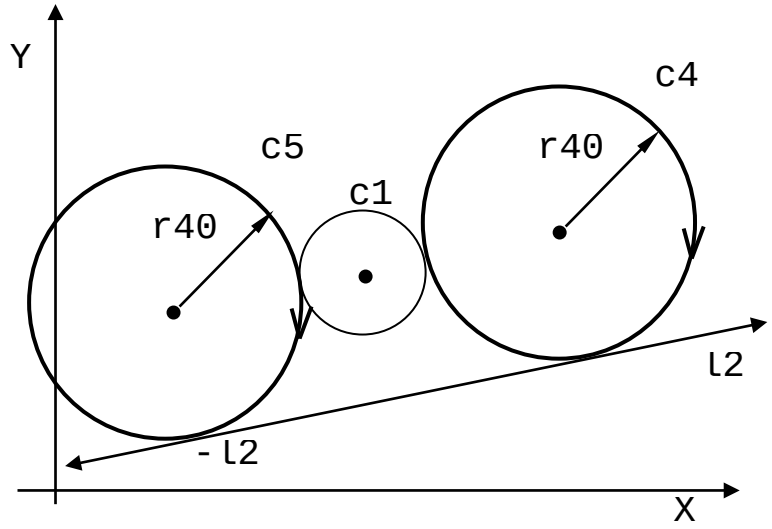


Рисунок 2.87

c3 = p1, -l1, r25

c4 = -l1, p1, r25

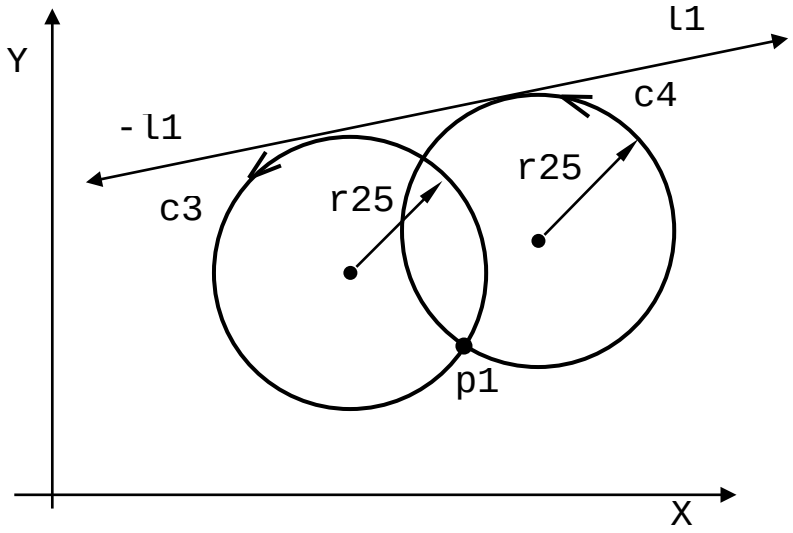


Рисунок 2.88

```
c5 = c1, c2, r-8

c6 = c2, c1, r-8
```

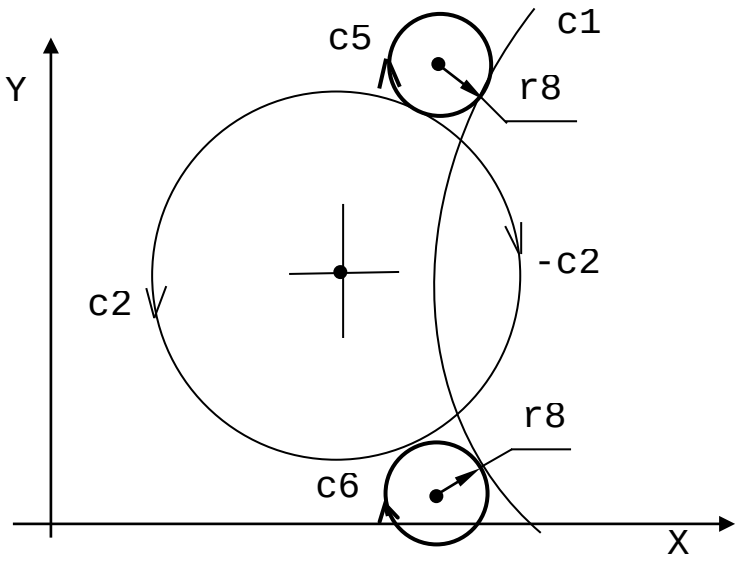


Рисунок 2.89

```
c9 = -c2, c1, r-8

c10 = c1, -c2, r-8
```

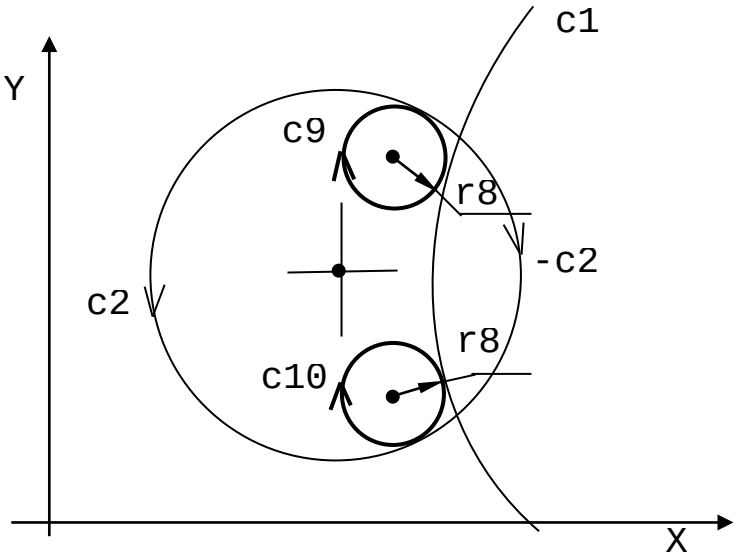


Рисунок 2.90

```
c2 = c1, p1, r60

c3 = p1, c1, r60
```

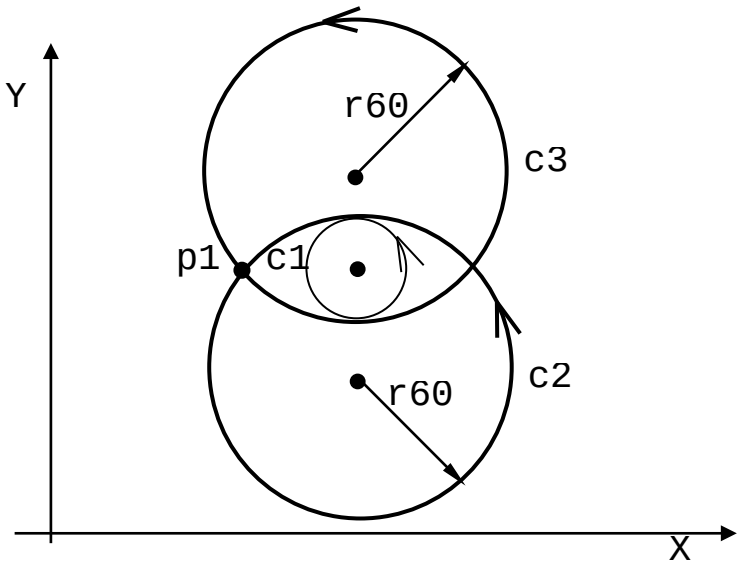


Рисунок 2.91

```
c1 = p1, p2, r20
c2 = p2, p1, r20
```

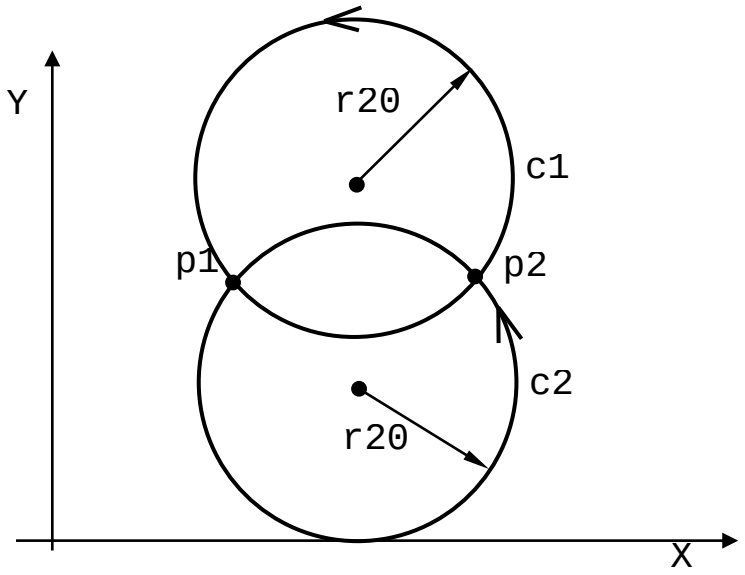


Рисунок 2.92

```
c3 = p1, l2
```

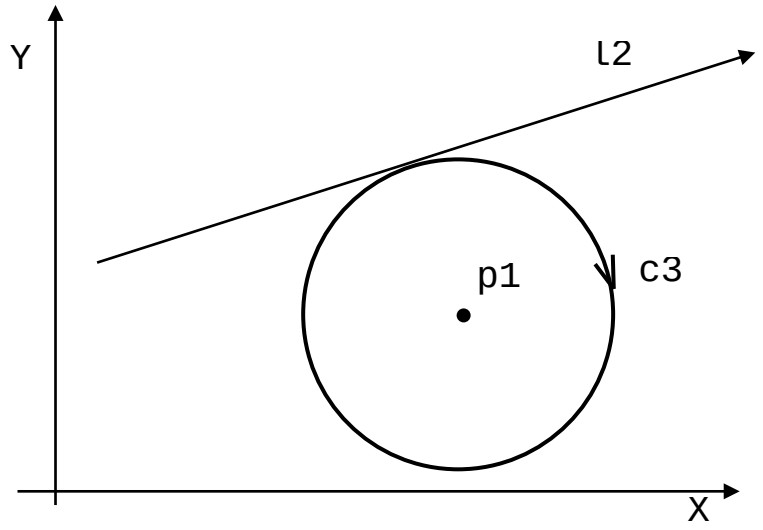


Рисунок 2.93

```
c2 = 1, c1
c3 = p1, c1, s2
```

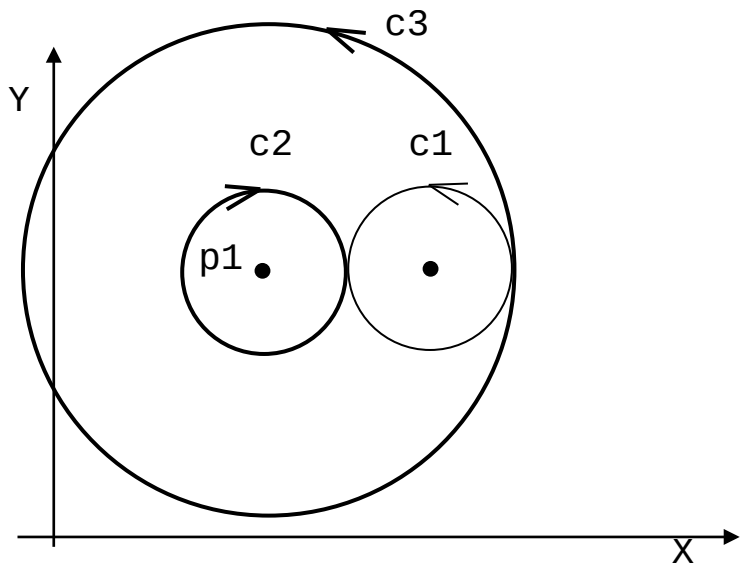


Рисунок 2.94

```
c1 = p1, p2, p3
c2 = p3, p2, p1
```

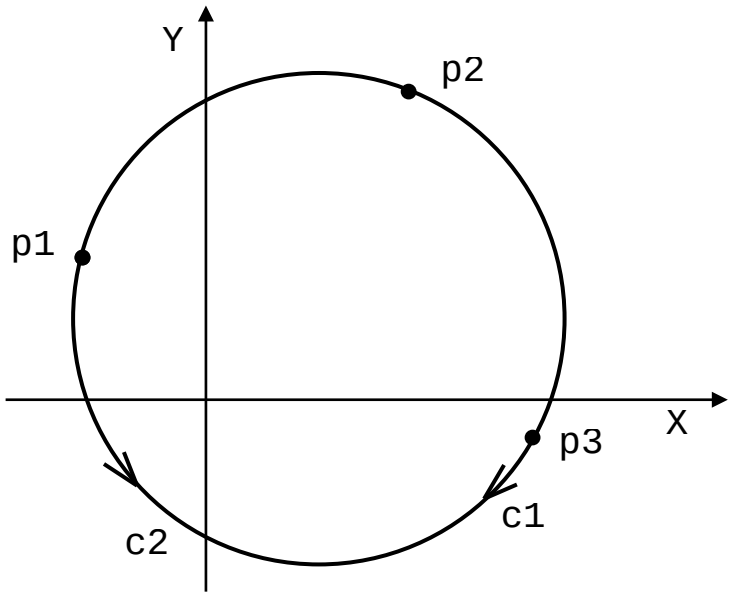


Рисунок 2.95

```
c1 = p1, r-40
```

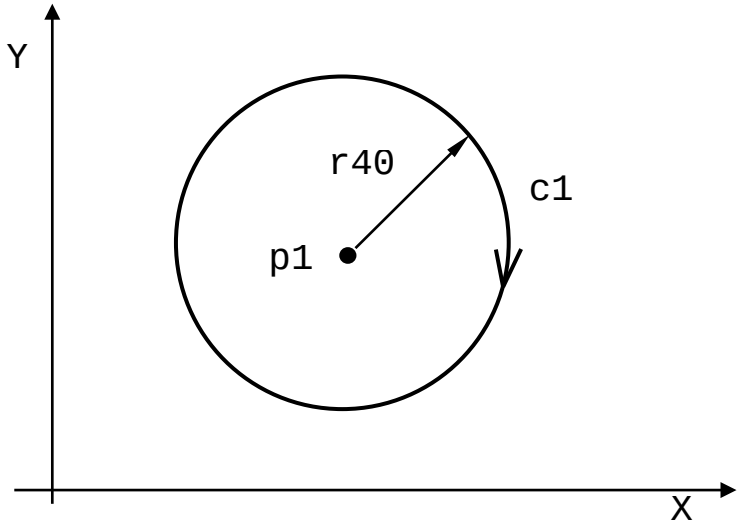


Рисунок 2.96

```
c6= c5, d-10
```

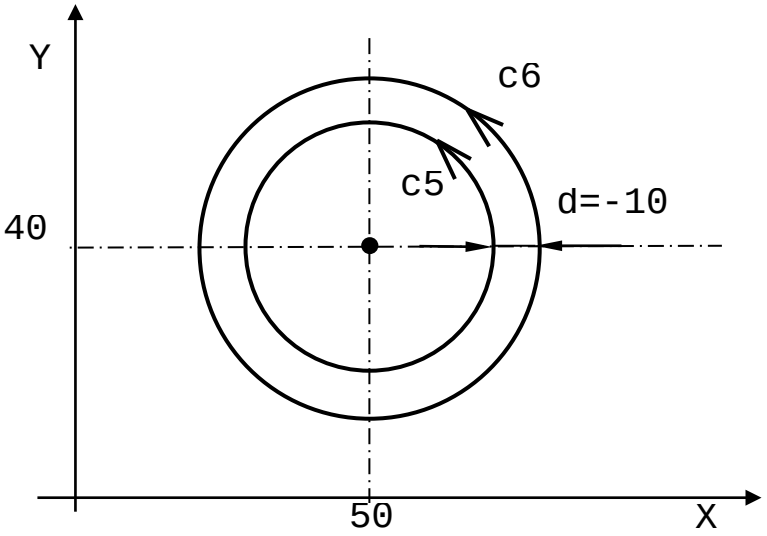


Рисунок 2.97

2.11.8 Определение профиля

Под профилем подразумевается последовательность геометрических элементов, накопленных в памяти системы до начала обработки. Профиль может быть открытым и закрытым.

2.11.8.1 Начало и конец профиля

Профиль, запрограммированный в геометрии GTL, определяется через функции **G21** и **G20**:

G21 - устанавливает начало профиля;
G20 - устанавливает конец профиля.

2.11.8.2 Открытый профиль

Если профиль открытый, он должен начинаться с точки (**pn**) и заканчиваться точкой, отличной от первой. Компенсация радиуса инструмента действует перпендикулярно к первому элементу на точке начала профиля и перпендикулярно к последнему элементу на точке конца профиля. Компенсация радиуса должна быть открыта на первой точке профиля программированием в кадре функций G21 G41/G42, и закрыта на последней точке с функциями G20 G40, как показано на рисунке 2.98.

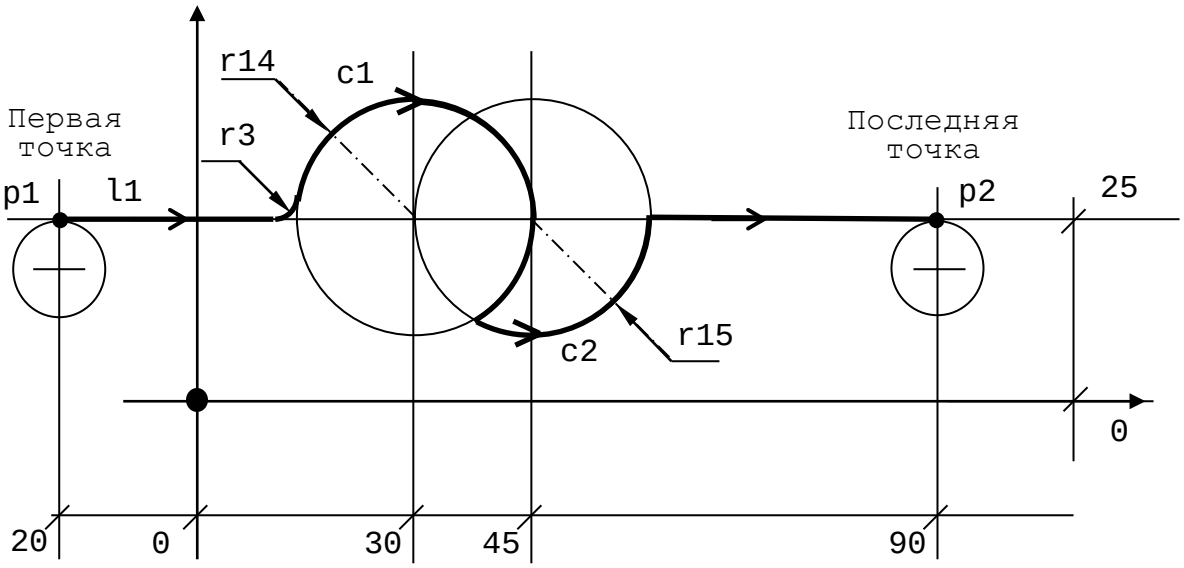


Рисунок 2.98

```
.............
l1 = XY 25,a
p1 = X-20 Y25
p2 = X90 Y25
c1 = I30 J25 r-14
c2 = I45 J25 r15
.............
G21 G42 p1          - первая точка
l1
r3
c1 s2
c2 s2
l1
G20 G40 p2
.............       - последняя точка
```

Компенсация радиуса должна быть открыта на первой точке профиля и закрыта на последней. Компенсация радиуса отменяется в первом кадре движения осей в плоскости профиля, следующего за функцией G40.

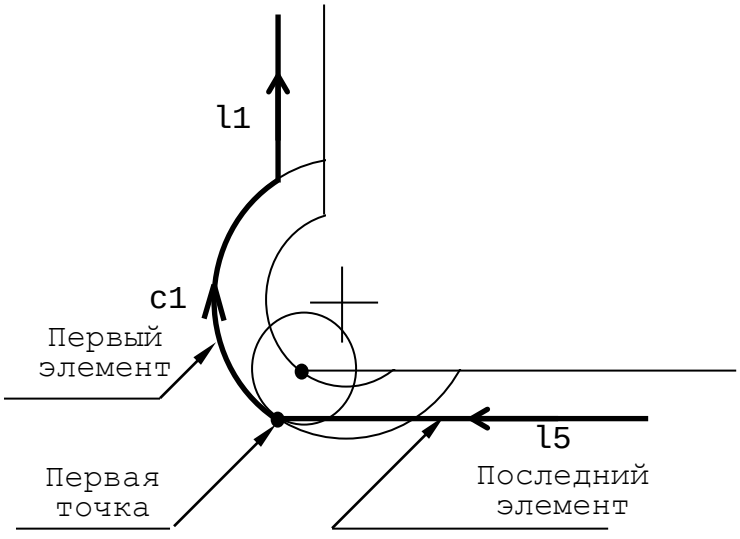
2.11.8.3 Закрытый профиль

Если профиль закрытый, сначала следует запрограммировать последний элемент (см. рисунки 2.99-2.100), а затем после последнего элемента вызвать первый элемент профиля. Первая точка скорректированного профиля - пересечение первого и последнего смещённых элементов (первая точка равна последней точке). Компенсация радиуса должна быть открыта в начале профиля в кадре вызова последнего элемента программированием функций G21 G41/G42 и закрыта в конце профиля в кадре вызова первого элемента с функциями G20 G40. Если первый и/или второй элементы являются окружностями, возможны два пересечения. Если не даётся никакой дополнительной информации, система выбирает первое. В случае если необходимо второе пересечение, следует программировать дискриминатор s2. Дискриминатор s2 программируется в кадре вызова последнего элемента в начале профиля и на последнем элементе в конце профиля (см. рисунок 2.100).

Пример
Закрытый профиль (см. рисунок 2.100):

```
c1 = I.. J.. r..
................
l1 = X.. Y..,a90
................
l5 = X.. Y..,a180
................
G21 G42 15 s2       - последний элемент
c1 s2               - первый элемент
l1
................    - последний элемент
l5 s2               - первый элемент
G40 c1
................
```

Компенсация радиуса должна быть открыта в начале профиля в кадре вызова последнего элемента и закрыта в конце профиля в кадре вызова первого элемента. Компенсация радиуса отменяется в первом кадре движения осей в плоскости профиля, следующего за функцией G40.




```
..............
l5=XY-15,a180
..............
l1=X-30Y-15,a135
..............
G21G42l5
l1
..............
l5
G20G40l1
```

Рисунок 2.99

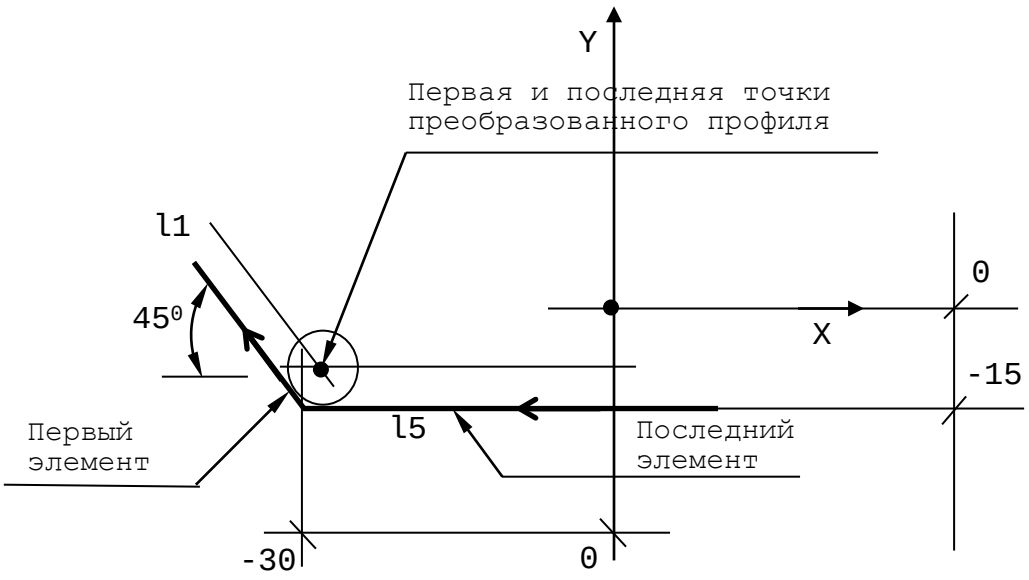


Рисунок 2.100

2.11.8.4 Движение осей шпинделя

В любой точке профиля представляется возможным двигать оси, не участвующие в контурной обработке, даже на первой точке, например, для входа в деталь. В случае открытых профилей движение на первой точке программируется после программирования точки. В случае закрытых профилей оно должно быть запрограммировано между определением последнего элемента профиля и первым элементом.

```
.........                G21 G42 l5    - последний элемент
G21 G42 p1               Z-10
Z-10                     l1            - первый элемент
l1                       .........
```

2.11.9 Соединение геометрических элементов

Геометрические элементы профиля могут быть связаны между собой за счёт тангенциального сопряжения, пересечения или присутствия автоматического соединения или фаски.

2.11.9.1 Пересечение между элементами

В случае пересечения двух прямых, возможно только одно решение. В случае пересечения прямая-окружность или окружность-окружность всегда возможны два решения. Система автоматически выбирает первое. Если необходимо получить второе, следует запрограммировать дискриминатор s2 после определения первого элемента.

Примеры пересечения прямая-окружность приведены на рисунке 2.101.

В случае пересечения прямой с окружностью первое и второе пересечения определяются направлением движения прямой линии. В случае пересечения окружности с окружностью (рисунок 2.102) первым пересечением является то, что слева от прямой, соединяющей центр первой окружности с центром второй, а второе пересечение то, что справа от той же прямой линии.

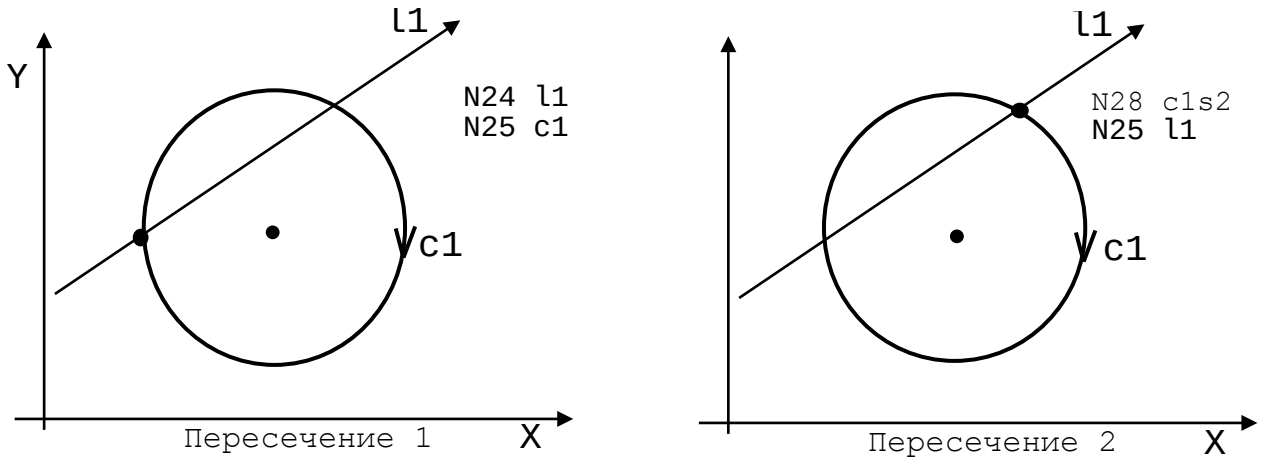


Рисунок 2.101

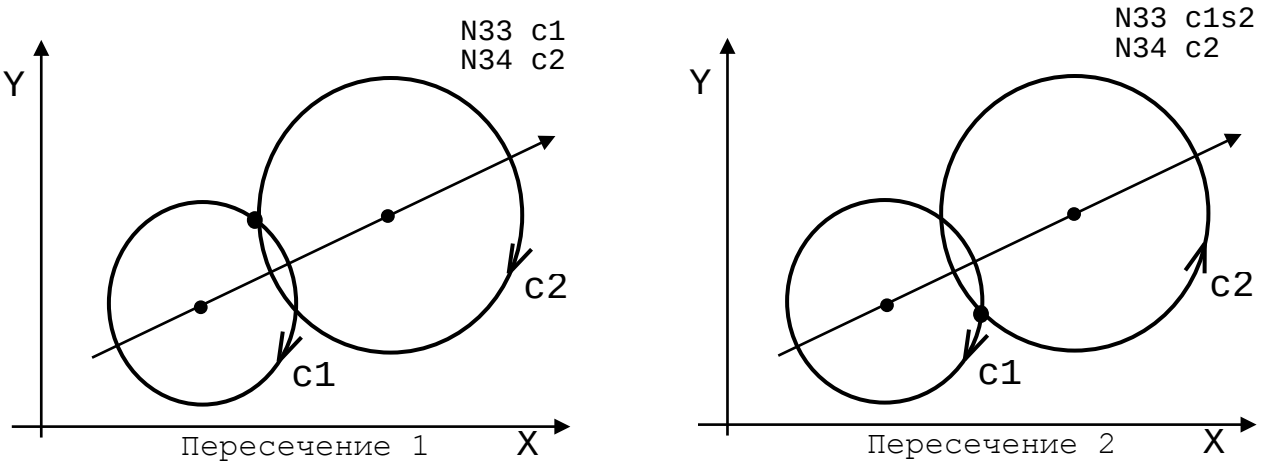


Рисунок 2.102

2.11.9.2 Соединения между элементами при помощи автоматического радиуса

Если элементы пересекаются, можно определить соединение между ними (прямые линии или окружности), программируя значение радиуса: положительное в направлении против часовой стрелки, отрицательное в направлении по часовой стрелке. Пример приведён на рисунке 2.103.

Соединение r не может быть запрограммировано в кадре, следующем сразу же за кадром с G21, или же в кадре, предшествующем кадру с G20 (т.е. профиль не может начинаться и заканчиваться с соединения). В случае активизации компенсации радиуса инструмент размещается на пересечении двух геометрических элементов, смещённых радиусом инструмента. Если необходимо ввести радиус между двумя элементами, следует программировать нулевой радиус r0 в соответствии с рисунком 2.104.

2.11.9.3 Скосы

Скосы между прямолинейными элементами можно определить, программируя значение скоса без знака, рассматриваемое как расстояние от точки пересечения. Пример приведён на рисунке 2.105.

Скос не может быть запрограммирован в кадре, который непосредственно следует за кадром G21 или предшествует G20 (т.е. профиль не может начинаться или закончиться со скосом).

В геометрическом программировании GTL перемещения всегда осуществляются с рабочей подачей, для программирования быстрого перемещения необходимо программировать скорость рабочей подачи F со значением быстрого хода.

Если плоскость интерполяции не является той, которая образована осями X и Y, следует вначале определить плоскость, а затем определять элементы профиля.

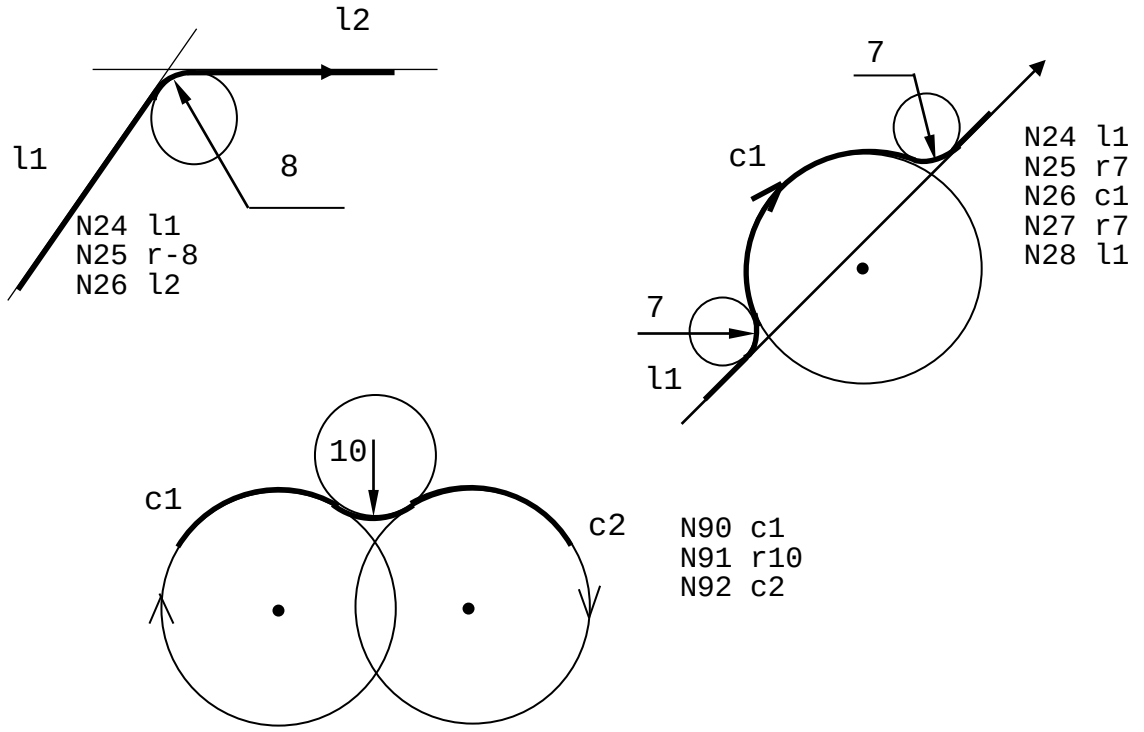


Рисунок 2.103

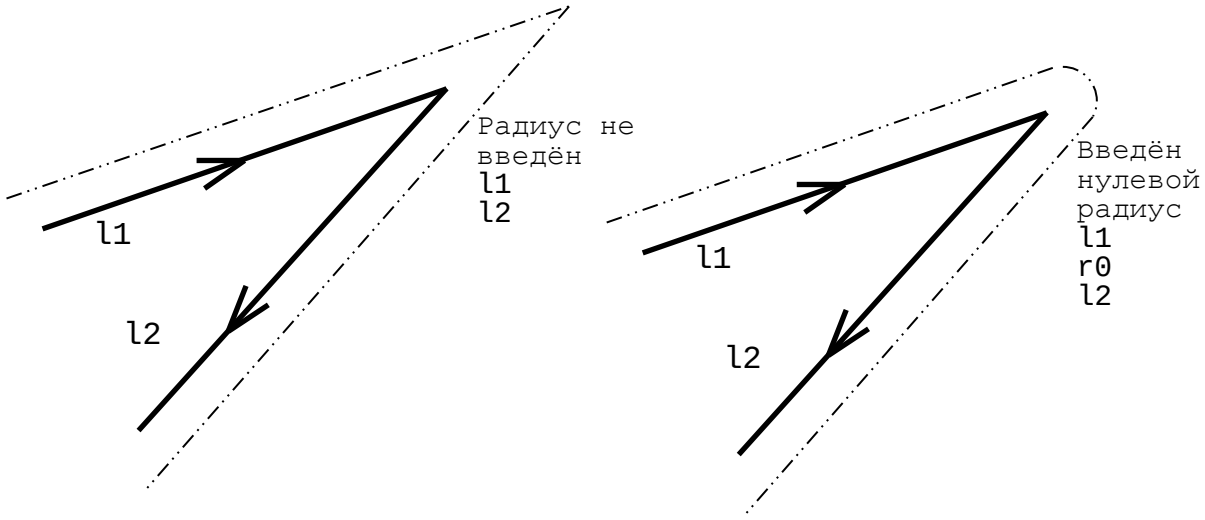


Рисунок 2.104

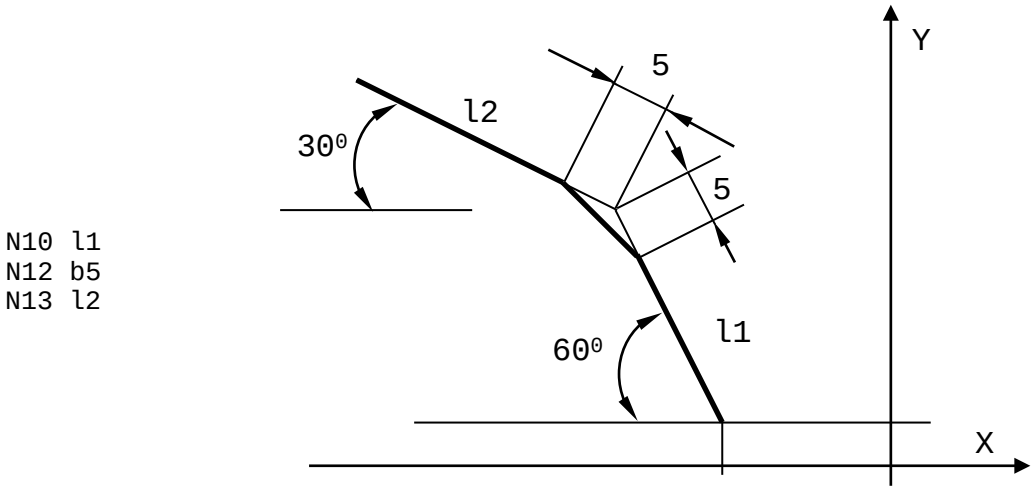


Рисунок 2.105

Пример

```
N1 (DPI, B, X)
N2 l1= B70 X40, a150
N3 l2= B8 X8, a-95
N4 p1= l1, l2
N5 l3= B8 X8, B70 X15
N6 c1= I70 J40, r-25
..........
N12 G21 G42 l2
```

2.11.10 Примеры программирования при помощи GTL

Примеры программирования при помощи GTL приведены на рисунках 2.106-2.109.

Пример 1 (см. рисунок 2.106):

```
N1  (DIS,"EXAMPLE GTL")
N2  l1 = X70 Y40, a150
N3  l2 = X8 Y8, a-95
N4  p1 = l1, l2
N5  l3 = X8 Y8, X70 Y15
N6  l4 = X50 Y, a90
N7  c1 = I70 J40 r-25
N8  c2 = p1, r-20
N9  T1.1 S800 F250 M6 M3 M7
N10 G XY
N11 Z-10
N12 G21 G42 l2
N13 l3
N14 r3
N15 l4
N16 r3
N17 c1
N18 r5
N19 l1
N20 r5
      N21 c2 s2
N22 l2
N23 G20 G40 l3
N24 G Z2
      N25 G X Y M30
```

Пример 2 (см. рисунок 2.107):

```
N1 (DIS,”EXAMPLE 2”)
N2 l1=X-50Y10,X30Y50
N3 l2=X30Y50,X70Y10
N4 l3=X70Y0,a-90
N5 l4=X0Y-20,A180
N6 l5=X10Y-20,X0Y0
N7 l6=X0Y0,X-10Y-20
N8 l7=X-50Y0a90
N9 c1=I-10J40r18
N10 c2=I50J30r-14
N11 c3=I40J-20r10
N12 S..M..T3.3M6M..
N13 G0X-30Y0
N14 Z-10
N15 G21 G42 l7
N16 l1
N17 r-8
N18 c1
N19 r-8
      N20 l1
      N21 l2
N22 c2s2
N23 l2
      N24 l3
      N25 r-10
N26 l4
N27 c3s2
N28 l4
N29 r-8
      N30 l5
      N31 l6
      N32 r-8
      N33 l4
      N34 r-10
      N35 l7
      N36 G20G40l1
      N37 G0Z0
```

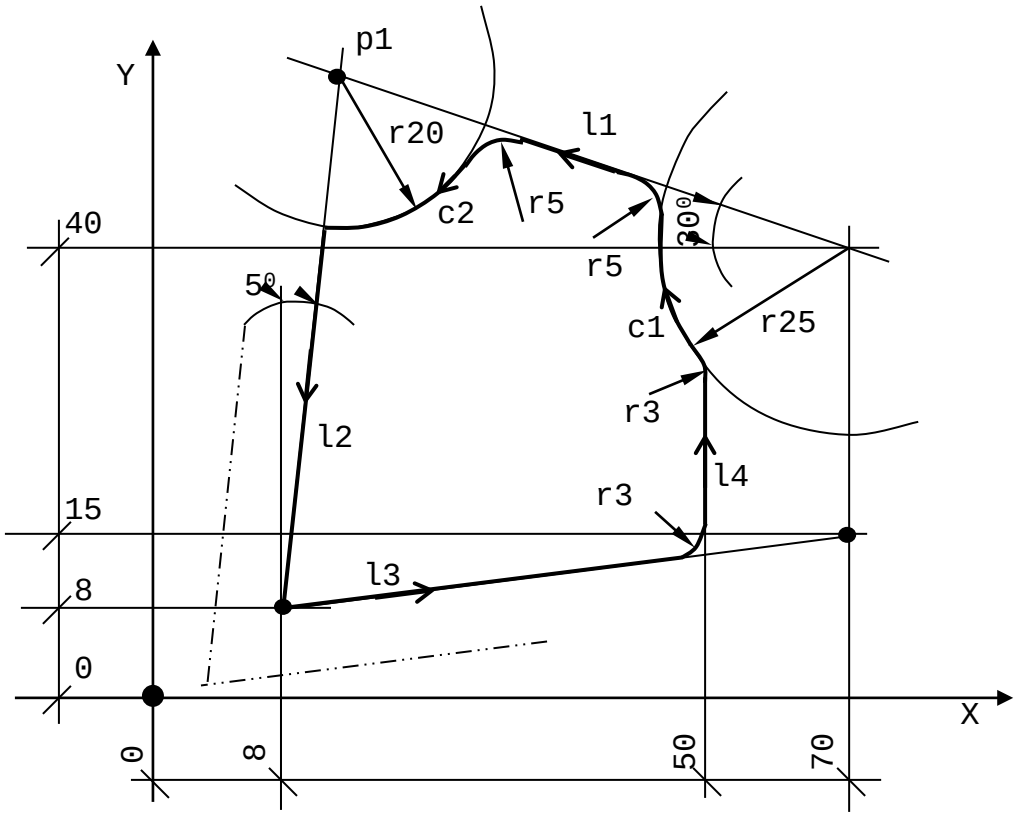


Рисунок 2.106

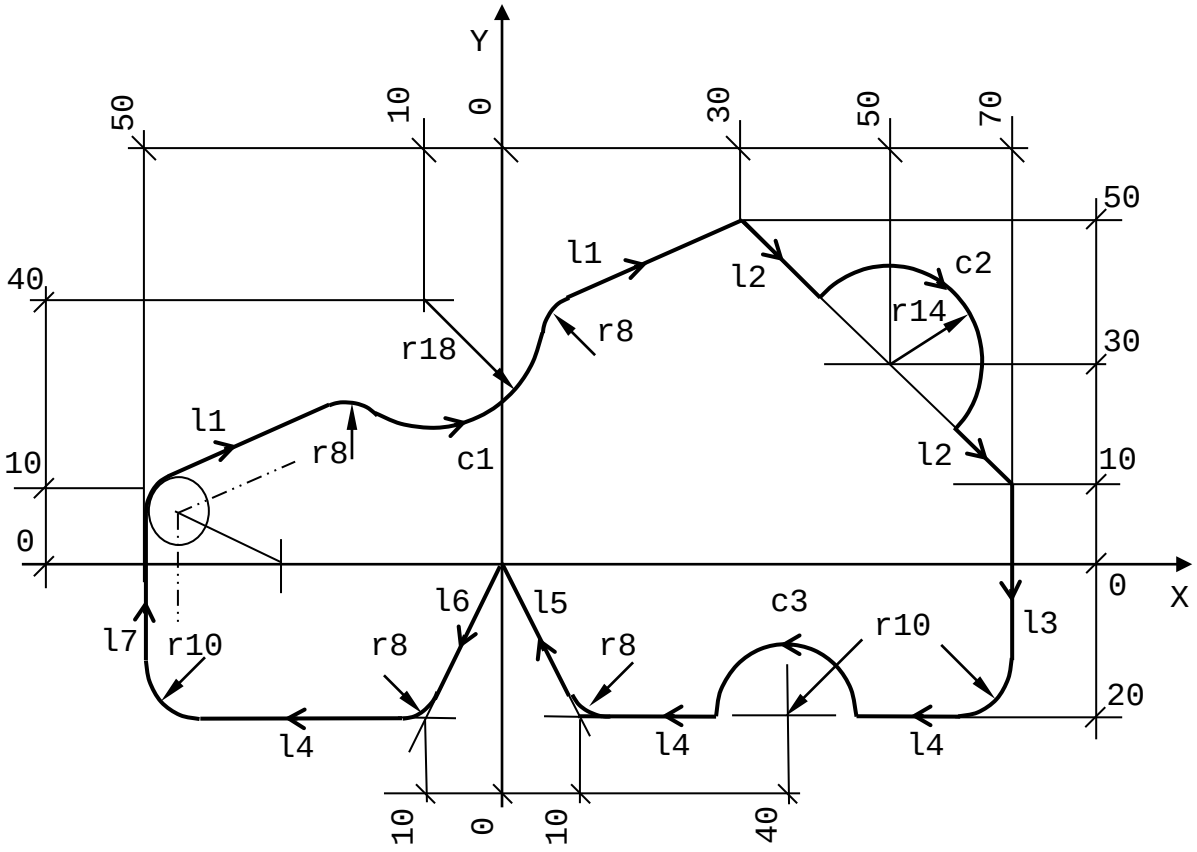


Рисунок 2.107

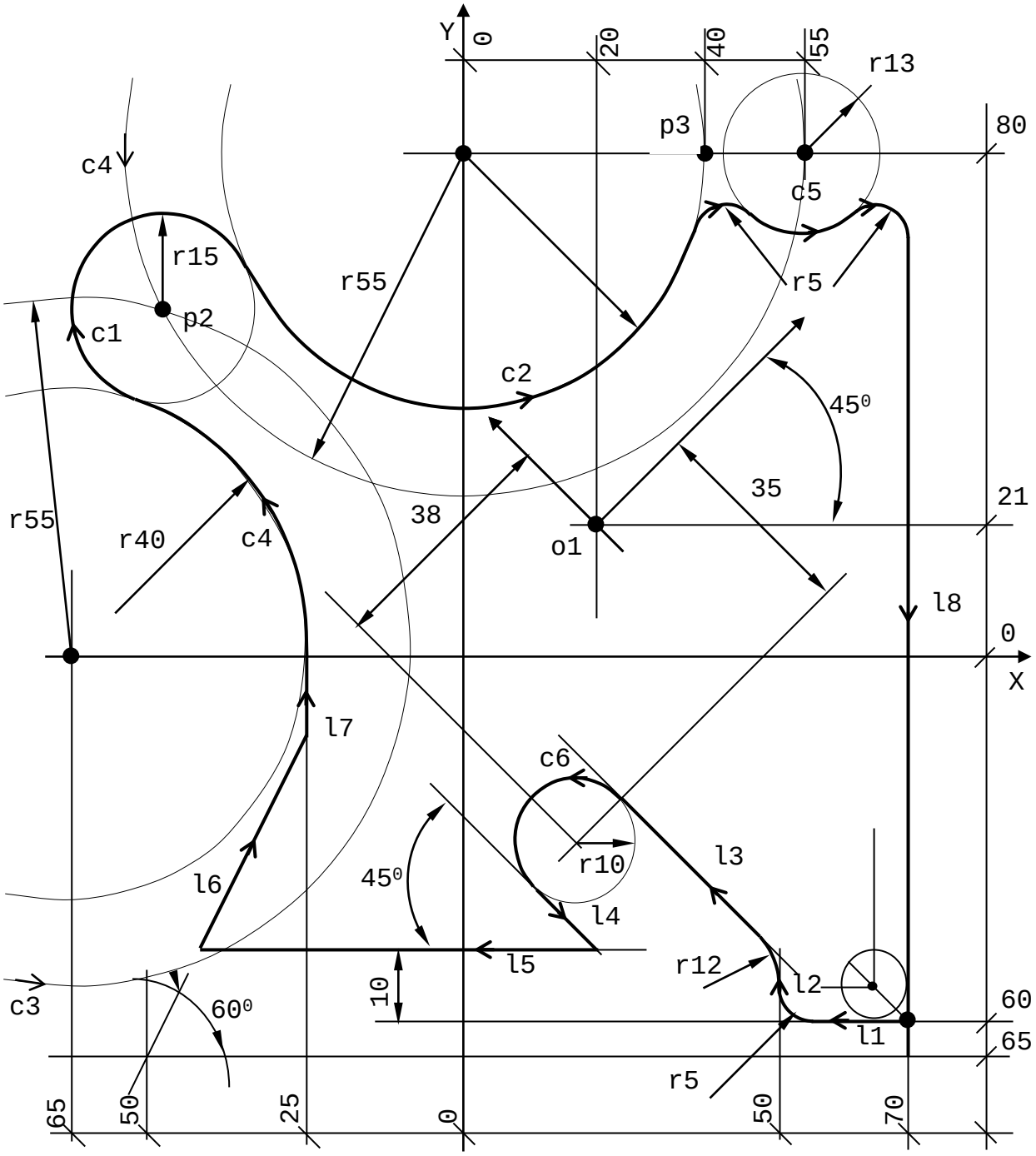


Рисунок 2.108

Пример 3 (см. рисунок 2.108):

```
N1 (DIS,EXAMPLE 3)
N2 S..F..T1.1M6M..
N3 o1=X20Y21a45
N4 l1=X0Y-60a180
N5 l2=X50Y0,a90
N6 c6=o1I-38J-35r10
N7 l3=c6,a135
N8 l4=c6,a-45
N9 l5=X0Y-50,a180
N10 l6=X-50Y-65,a60
N11 l7=X-25Y0,a90
N12 c3=I-65J0r55
      N20 G21 G42 l8
      N21 Z-10
N22 l1
N23 r-5
N24 l2
      N25 r12
      N26 l3
      N27 c6
N28 l4
      N29 l5
N30 l6
N31 l7
```

```
N13 c4=I0J80r55
N14 p2=c3,c4
N15 c1=p2,r-15
N16 p3=X40Y80
N17 c2=c1,p3,r40
N18 c5=I55J80r13
N19 l8=X70Y0,a-90
```

```
N32 r40
      N33 c1
      N34 c2
      N35 r-5
N36 c5
N37 r-5
N38 l8
      N39 G20 G40 l1
      N40 G Z
      N41 X..Y..M30
```

Пример 4 (см. рисунок 2.109):

```
N1  (DIS,"GTL WITH ROTATION")
N2  F.. S.. T2.2 M6
N3  UOV=2
N4  p1=X50 Y0
N5  c1=I0 J0 r50
N6  c2=I0 J0 r10
N7  l1=c2,a180
N8  l3=X0 Y0,a45
N9  l2=c2,a45
N10 p2=l3,c1,s2
N11 X60 Y0
N12 Z-20
"START" N13 E25=0
N14 (RPT,8)
N15 (URT,E25)
N16 G21 G42 p1
```

```
N17 c1
      N18 r3
      N19 l1
      N20 r-3
N21 l2
N22 r3
      N23 c1
N24 G20 G40 p2
      N25 E25=E25+45
N26 (ERP)
N27 (URT,0)
      "END" N28
N29 UOV=0
N30 (EPP,START,END)
      N31 G0 Z20
      N32 X Y M30
```

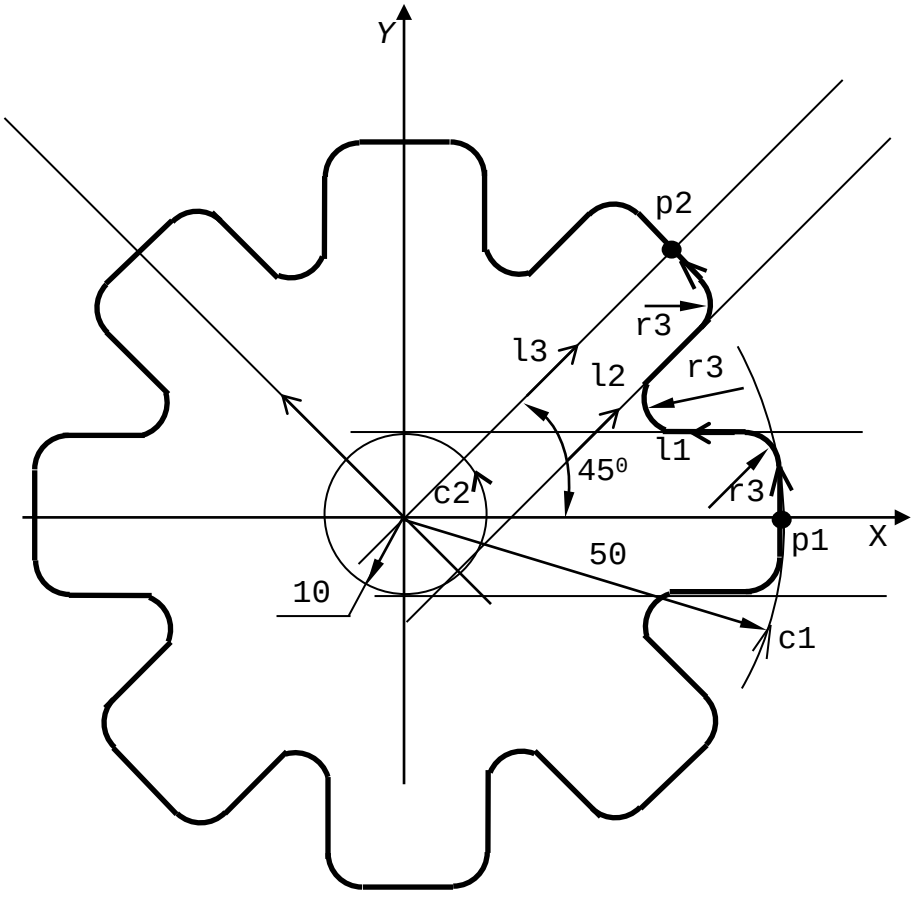


Рисунок 2.109

2.12 Параметрическое программирование

Используя коды **E**, можно через параметры программировать геометрические и технологические данные цикла обработки. С параметрами допускаются математические и тригонометрические действия, а также вычисление выражений. Максимальное число параметров **E** не ограничено и определяется во время конфигурации системы. Параметры **E** имеют различные индексы для переменных различного формата. Описание параметров **E** для различных форматов представлено в таблице 2.8.

Параметры **E** получают значения в кадрах назначения.

Таблица 2.8 – E-параметры и их форматы

Формат	Параметры	Мин/макс величина
BY (байт)	E0...E9	от 0 до 255
IN (целое)	E10..E19	от -32768 до +32767
LI (целое с двойной точностью)	E20..E24	от -2.147.483.647 до +2.147.483.647
RE (действительное)	E25..E29	+(-)7 целые или десятичные цифры
LR (действительное с двойной точностью)	E30..(*)	+(-)16 целые и десятичные цифры +(-)13 целые числа
(*) - максимальное число E-параметров, определённое во время конфигурации.		

Формат кадра назначения:

En = <выражение> ,

где:

<выражение> - может быть цифровой величиной или математическим выражением, результат которого будет запомнен под параметром **E** индекса **n**. **«Выражение»** - это математическое выражение, составленное из арифметических операторов, функций и операндов (параметры **E**, числовые константы).

Допустимы следующие арифметические операции:

+ сложение;
- вычитание;
***** умножение;
/ деление.

Возможные функции:

- **SIN (A)** - вычисляет синус A;
- **COS (A)** - вычисляет косинус A;
- **TAN (A)** - вычисляет тангенс A;
- **ARS (A)** - вычисляет арксинус A;
- **ARC (A)** - вычисляет арккосинус A;
- **ART (A)** - вычисляет арктангенс A;
- **SQR (A)** - вычисляет квадратный корень A;
- **ABS (A)** - вычисляет абсолютное значение A;
- **INT (A)** - вычисляет целое число A;
- **NEG (A)** - инвертирует знак A;
- **MOD (A,B)** - вычисляет остаток отношения между A и B;
- **FEL (A,B)** - извлекает элемент с индексом **B** (1, 2, 3) для геометрического элемента «линия» (прямая), которая определена в GTL с индексом **A** (ln = p1,p12, где n=A); для индекса **B**: 1 - синус угла; 2 – косинус; 3 - расстояние от прямой линии до начальной точки;
- **FEP (A,B)** - извлекает элемент **C** индексом **B** (1,2) для геометрического элемента «точка», которая определена в GTL с индексом **A** (pn = X10Y15, где n=A); для индекса B: 1 = абсцисса точки; 2 = ордината;
- **FEC (A,B)** - извлекает элемент с индексом **B** (1,2,3) для геометрического элемента «окружность», которая определена в GTL с индексом **A** (cn = X10Y15r24, где n=A); для индекса **B**: 1 - абсцисса центра; 2 – ордината; 3 - радиус окружности.

Значения для (A) и (A,B) могут быть параметрами **E** или числовыми константами. Выражение вычисляется с учётом приоритета скобок и знаков. Результат, если совместим, преобразуется в формат параметра **E**, указанный слева от знака «=».

Примеры

```
E30 = FEL(5,1)     придаёт E30 значение синуса угла, который образует прямая l5 с абсциссой.
E34 = FEP(4,2)     придаёт E34 значение ординаты точки p4.
E42 = FEC(8,3)     придаёт E42 значение радиуса окружности c8.
```

Значения вычисления параметров:

```
N1 E37=(E31*SIN(E30)+123.4567)/SQR(16)   - выполняет математическое решение выражения и придаёт результат параметру E37.
"LAB1"E51=-0.00000124+5 - выполняет вычисление выражения и придаёт результат параметру E51.
E40=TAN(35)        - извлекает тангенс 35 градусов и придаёт результат параметру E40.
/E35=FEP(37,1)     - извлекает абсциссу точки p37, ранее занесённой в память, и придаёт значение параметру E35.
E31=NEG(E31)       - меняет знак параметра E31.
E7=81              - придаёт значение параметру E7.
E25=E25+30         - новым значением параметра E25 будет сумма константы 30 и текущего значения параметра E25.
E2=SK396           - придаёт E2 содержимое байта 396 пакета «K».
E8=SYVAR1          - придаёт E8 значение переменной SYVAR1.
```

Операнды тригонометрических функций должны быть выражены в градусах. Результат функций ARS, ARC, ART также выражается в градусах.

Параметры **E** могут быть использованы как внутри программы, так и внутри подпрограммы, и могут быть воспроизведены.

Пример

```
(DIS,E54)   - воспроизводит на экране величину E54=...
```

Параметры стираются при выключении УЧПУ и могут быть приведены к нулю, если это предусмотрено в фазе характеризации. Использование параметров **E** сведено в таблицу 2.9.

Таблица 2.9 - Использование параметров E

Параметры (Формат)	Данные (геометрические, технологические)	Примеры программирования
E0..E9 (BY)	функции G функции M коды RPT	GE1 ME3 (RPT,E9)
E10..E19 (IN)	Номер абсолютной начальной точки функции T функции S	(UAO,E10) (UOT,E11,X..,Y..) TE14.E15 SE15
E20..E24 (LI)		
E25..E29 (RE)	функции F коды URT коды SCF индексные оси код UGF	FE27 (URT,E25) (SCF,E26) PE29 UGF=E28
E30...(*) (LR)	координаты осей A B C X Y Z U … координаты R составные операции IJK факторы корректировок u v w глобальные переменные системы: TMR UOV	 XE32 RE33 KE34 vE35 TMR=E38 UOV=E40

2.13 Кадры с трёхбуквенными операторами

Этот раздел описывает функциональность и синтаксис кадров, имеющих в качестве операторов трёхбуквенные коды. Можно установить семь классов трёхбуквенных операторов:

1) операторы, изменяющие систему начала отсчёта осей;
2) операторы, изменяющие последовательность выполнения программы;
3) смешанные операторы;
4) операторы ввода/вывода;
5) операторы контроля инструмента;
6) операторы видеографического управления;
7) операторы управления коррекциями.

2.13.1 Трёхбуквенные операторы, модифицирующие систему отсчёта осей

Операторы этого класса позволяют изменять декартовую систему отсчёта, по отношению к которой был запрограммирован профиль. К этому классу принадлежат следующие операторы: **UAO, UOT, UIO, MIR, URT, SCF, RQO**.

2.13.1.1 Использование абсолютных начальных точек - UAO

Оператор **UAO** выбирает одну из абсолютных начальных точек, ранее определённых командой **ORA**.

Формат:

(UAO,n[,VAR-1,VAR-2...VAR-n]) ,

где:

n - определяет номер начальной точки, которую надо выбрать; может быть цифровой постоянной или параметром **E** типа целый (от E10-E19);

VAR-1 - символ, представляющий название оси, для которой определяется начальная точка «n»; для необъявленных осей остаётся в силе текущая начальная точка. Если «название оси» не присутствует, начальная точка «n» приводится в действие для всех осей, для которых была объявлена эта начальная точка.

Пример

(UAO,1)	- абсолютная начальная точка 1 активна для всех осей;
.....	- часть УП, отнесенная к начальной точке 1 для всех осей;
(UAO,2,X,Y)	- абсолютная начальная точка 2; активизируется для осей X и Y;
(UAO,3,B)	- абсолютная начальная точка 3; активизируется только для оси B;
.....	- часть УП, отнесенная к начальной точке 2 для осей XY, к точке 3 - для оси B и к точке 1 - для всех остальных осей;
(UAO,0)	- активизирование нулевой начальной точки для всех осей.

При включении УЧПУ и после команды «СБРОС» автоматически активизируется нулевая начальная точка для всех осей. Максимально могут присутствовать шесть «названий осей». Не могут быть определены одинаковые «названия осей». Если требуется привести в действие различные начальные точки для различных осей, необходимо программировать столько кадров с этими операторами, сколько имеется начальных точек. Если выбранная начальная точка (-n) загружена в файл альтернативной системы измерения, она автоматически переводится в текущую систему измерения.

2.13.1.2 Определение и использование временных начальных точек - UOT

Оператор **UOT** выбирает абсолютную начальную точку, объявленную в кадре, изменяя её временно на величину, равную запрограммированной.

Формат:

(UOT,n,VAR-1 [,VAR-2...VAR-n]) ,

где:

n - имеет то же значение, что и для оператора UAO;

VAR-1 - операнд типа «ось-размер»; значение, приданное ему, рассматривается как корректировка, к которой надо прибавить значение, содержащееся в абсолютной начальной точке для той оси. Для необъявленных осей остаётся в силе текущая начальная точка.

Пример

```
(UAO,0)                  - активизируется абсолютная начальная точка 0. Программа, отнесенная к абсолютной начальной точке 0 для всех осей.
1) (UOT,0,X100,Y100)
   ........              - применяется временная начальная точка к начальной точке 0 с корректировками X100 и Y100 (временная начальная точка).
2) (UOT,1,X-250,Y-50)
...                      - применяет временная начальная точка к абсолютной начальной точке 1 с корректировками X-250 и Y-50.
   (UAO,0)               - активизируется абсолютная начальная точка 0 для всех осей.
```

По крайней мере, должен присутствовать один операнд оси. Максимально могут присутствовать шесть осей. Не могут быть определены операнды осей с одним и тем же названием. Временная начальная точка остаётся активной до того, как определяется новая временная начальная точка, или до вызова абсолютной начальной точки, или до команды «СБРОС». Размер в операторе UOT необходимо программировать в текущей размерности (G70/G71).

Пример изображен на рисунке 2.110.

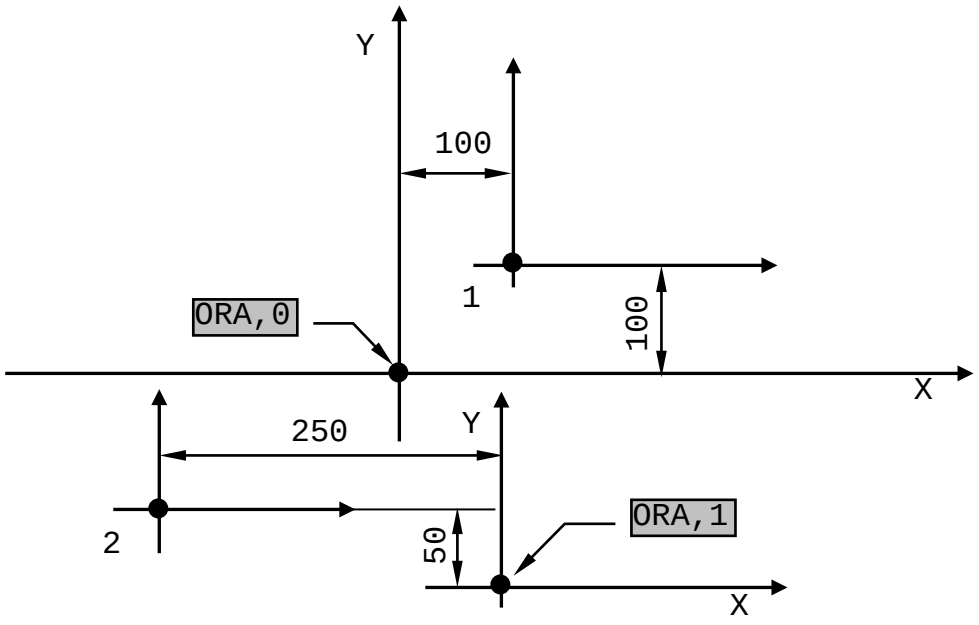


Рисунок 2.110

2.13.1.3 Определение и использование начальных точек по приращениям - UIO

Эта команда позволяет приращением переместить текущую начальную точку для всех осей, заданных в ней.

Формат:

(UIO,VAR-1 [,VAR-2,...VAR-n]) ,

где:

VAR-n - представляет ось и размер. Система берёт размер как абсолютное смещение и прибавляет его к абсолютной начальной точке для данной оси. Для необъявленных осей текущая начальная точка остается

в силе. Количество значений **VAR-n** должно быть не больше шести (одно значение на одну ось).

Пример приведён на рисунке 2.111.

```
.............
N65 (UIO,X20,Y20) -точка 1
.............
N121 (UIO,Y-40)   -точка 2
.............
N180 (UIO,X-45)   -точка 3
.............
N230 (UIO,Y-35)   -точка 4
.............
N300 (UAO,0)
```

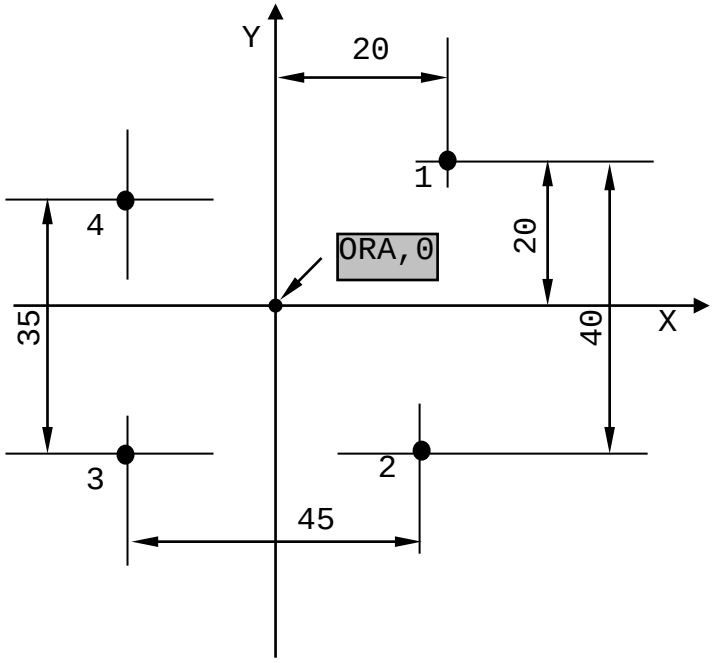


Рисунок 2.111

Начальная точка, заданная по приращениям, остаётся в силе до её переопределения с новой командой UIO, или восстанавливается абсолютная начальная точка при помощи (UAO,O) или операции [«СБРОС»].

Значение приращений в команде UIO необходимо программировать в текущей размерности (G70, G71).

2.13.1.4 Зеркальная обработка - MIR

Оператор MIR инвертирует запрограммированные направления перемещений, объявленных в операторе. Для необъявленных осей предыдущая функция MIR остаётся в силе. Если не запрограммирован никакой операнд, функция MIR выводится из действия для всех конфигурируемых осей.

Формат:

(MIR [,VAR-1,...,VAR-n]) ,

где:

VAR-n - должен быть буквой, соответствующей одному из возможных названий конфигурируемых осей системы.

Пример приведён на рисунке 2.112:

```
.....
N24 (MIR,X)
.....
N42 (MIR,X,Y)
.....
N84 (MIR,X)
.....
N99 (MIR)
```

Зеркальная обработка активизируется для запрограммированной оси, начиная с первого движения данной оси после команды MIR. Зеркальное отображение осуществляется относительно текущей начальной точки. Максимально может быть запрограммировано шесть осей. Нельзя программировать 2 раза одну и ту же ось. Если присутствуют команды вращения (URT) и зеркальной обработки (MIR), они устанавливаются в следующем порядке: MIR и URT.

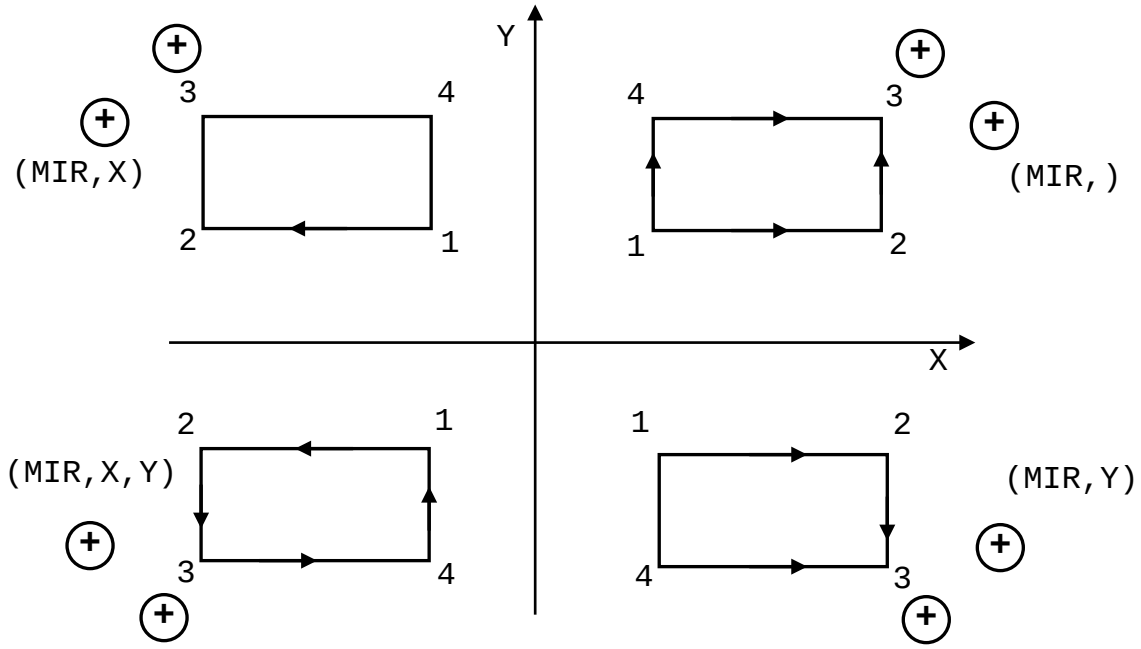


Рисунок 2.112

2.13.1.5 Поворот плоскости - URT

Оператор URT вращает плоскость интерполяции на угол, значение которого дано операндом. Центром вращения является текущая начальная точка.

Формат:

(URT , ЗНАЧЕНИЕ) ,

где:

ЗНАЧЕНИЕ - представляет величину угла, выраженную в градусах и десятых градуса; может быть выражен явно или неявно (параметр **E** типа от E25 до E29). Если операндом является «0», функция отменяется.

Операнд должен присутствовать обязательно. После кадра с URT вращение применяется к запрограммированным координатам. Координаты, относящиеся к нулю станка (G79), не вращаются. Если присутствуют команды вращения (URT) и зеркальной обработки (MIR), они устанавливаются в следующем порядке: MIR и URT.

Угол поворота, заданный оператором URT, автоматически обнуляется при выполнении команды SPG или операции [«СБРОС»].

Пример поворота плоскости изображен на рисунке 2.113.

```
(UOT,0,X100,Y50)
(URT,30)
..........
..........
..........
(UAO,0)
(URT,0)
```

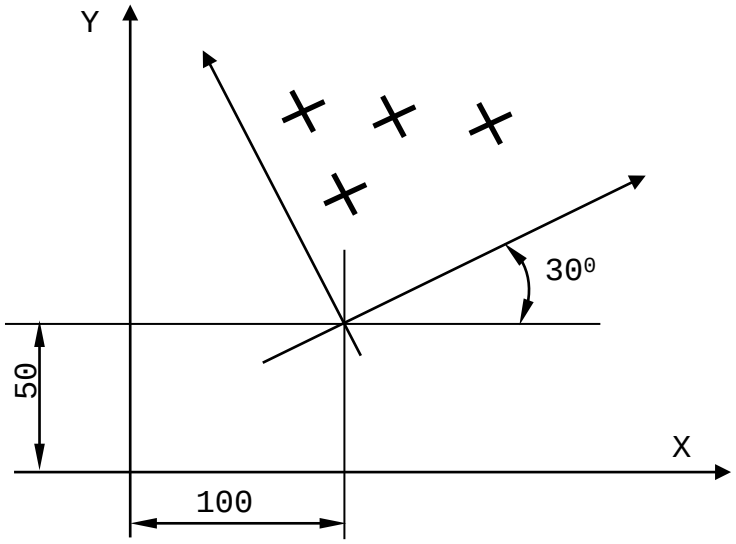


Рисунок 2.113

2.13.1.6 Масштабирование - SCF

Масштабирование применяется для объявленных в операторе SCF осей.
Формат:

(SCF[,n[,VAR-1,...,VAR-m]]) ,

где:

N - определяет коэффициент масштабирования, который должен быть применён; может быть запрограммирован как явно, так и неявно при помощи параметра **E** типа RE (от E25 до E29);

VAR-1 - символ, представляющий одну из осей, для которой приведён в действие коэффициент масштабирования; для необъявленных осей масштабирование отменяется. Если к SCF не присоединён какой-либо операнд, масштабирование отменяется для всех осей.

Может быть запрограммировано максимально шесть названий осей.

Пример

```
.........
  (SCF,3)        - применяет коэффициент 3 ко всем конфигурируемым осям
  .........
  (SCF,2,X)      - применяет коэффициент 2 для оси X и отменяет коэффици-
                   ент 3 для других осей
  .........
  (SCF)          - отменяется коэффициент масштабирования для всех осей
```

2.13.1.7 Модификация начальной точки - RQO

Оператор RQO изменяет начальную точку для осей, объявленных в операторе, на запрограммированную величину.
Формат:

(RQO,n,VAR-1 [,VAR-2,...,VAR-N]) ,

где:

n - определяет номер модифицируемой начальной точки, его величина заключена между 0 и 99 и тесно связана с числом записей, определённых во время создания файла начальных точек; **n** - может быть выражено явно или неявно при помощи параметра **E** типа целый (от E10 до E19);

VAR-i - операнд типа «ось-размер», приданное ему значение является коррекцией запрограммированной начальной точки данной оси.

Пример

(RQO,3,XE31) - изменяет начальную точку 3 оси X на величину, объявленную в E31.

Должен присутствовать, по крайней мере, один операнд (ось). Может быть запрограммировано максимально шесть операндов (осей). Не может быть определено больше одного операнда с одним и тем же названием оси. Начальная точка изменяется как в файле начальных точек (следовательно, результат модификации постоянный), так и в памяти пользователя, если эта начальная точка активна для оси в момент модификации.

В файле начальных точек модификации выполняются в той системе измерения, в которой выражена выбранная начальная точка. Однако необходимо определить программируемую величину модификации в текущих единицах измерения (G70/G71), и к ним не применяется масштабирование.

2.13.1.8 Примеры с использованием операторов MIR и URT

Примеры с использованием операторов MIR и URT иллюстрируются рисунками 2.114-2.117.

Пример 1 приведён на рисунке 2.114.

Примечание - Использование операторов RPT, ERP рассматривается в п.2.13.2.1.

```
N200 S1500 T8.8 M6 M3
N201 (RPT,2)
N202 G X90 Y20
N203 Z2
N204 G1 Z-8 F150
N205 X40 F200
N206 G2 Y60 I40 J40
N207 G1 X90
N208 G Z-100
N209 (MIR,X)
N210 (ERP)
N211 (MIR)
N212 GZ50
.............
```

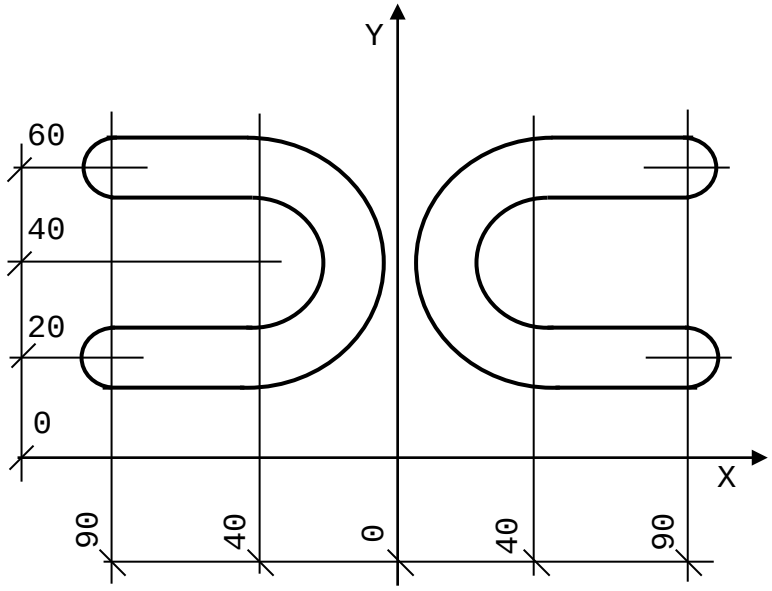


Рисунок 2.114

Пример 2 приведён на рисунке 2.115.

```
N100 S2000 F200 T3.3 M6
N101 (UOT,0,X30,Y22)
N102 (URT,20)
N103 G81 R3 Z-25 M3
N104 X25 Y25
N105 X40 Y10
N106 X55
N107 X70 Y25
N108 G80 Z20
N109 (UAO,0)
N110 (URT,0)
N111 S1000 T4.4 M6
.............
```

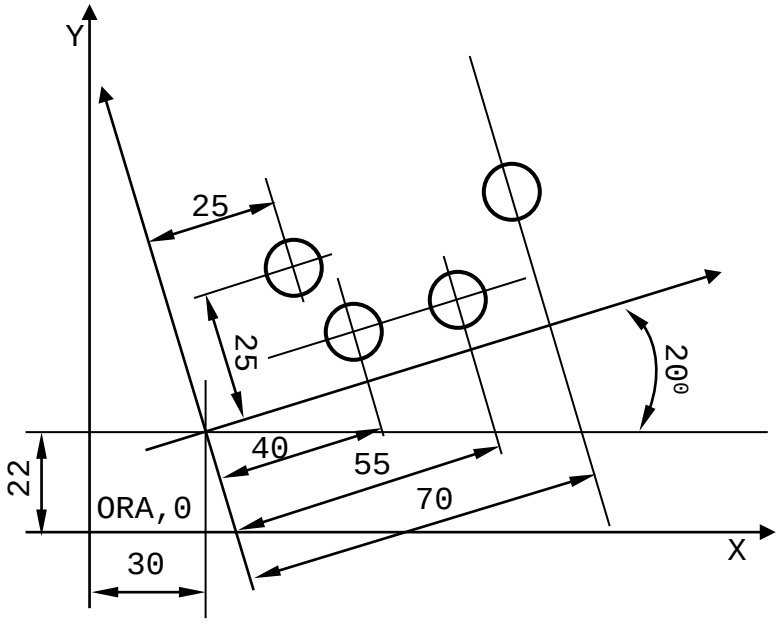


Рисунок 2.115

Пример 3. Использование URT совместно с RPT при параметрическом программировании показано на рисунке 2.116.

```
N1  (DIS"...")
N2  S1500 T5.5 M6 M3
N3  E25=0
N4  (RPT,8)
N5  (URT,E25)
N6  G X40 Y
N7  Z2
N8  G29 G1 Z-6 F150
N9  X80 F200
N10 Z-12 F150
N11 X40 F200
N12 G Z20
N13 E25=E25+45
N14 (ERP)
N15 (URT,0)
.........
```

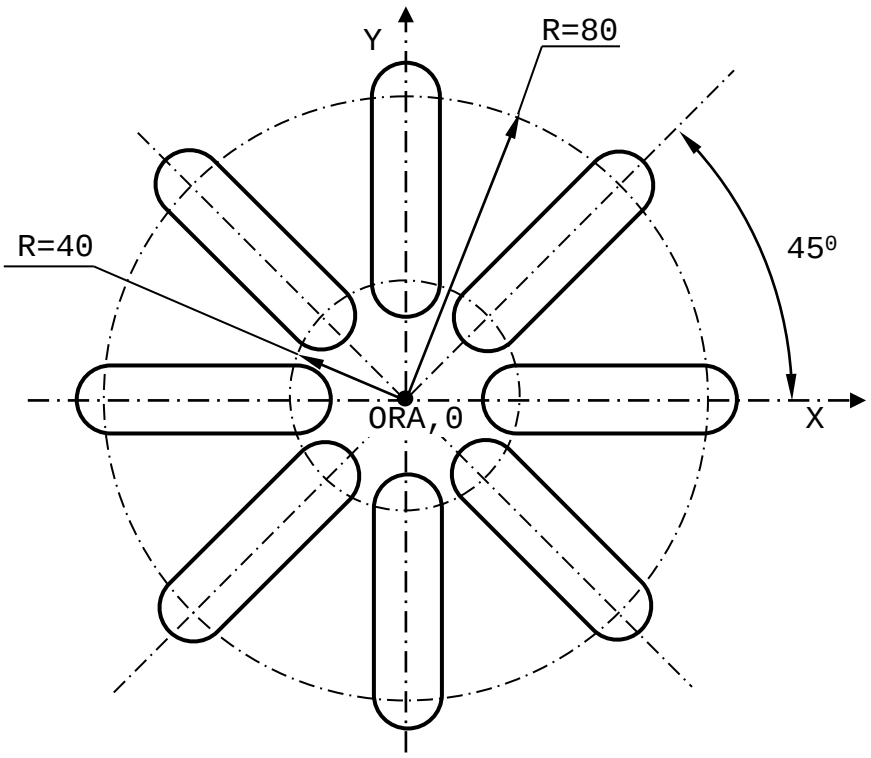


Рисунок 2.116

Пример 4. Поворот профиля, определенного в GTL, приведён на рисунке 2.117

```
N75 (DIS,"MILL D=8")
N81  c1=I15 J15 r5
N82  c2=I50 J30 r5
N83  c3=I30 J50 r5
N84  l1=c1,c2
N85  l2=c2,c3
N86  l3=c3,c1
N87  F170S800T1.1M6 M3
N88  E26=0
N89  (RPT,6)
N90  (URT,E26)
N92  G21 G41 c1
N93  Z-10
N94  l1
N95  c2
N96  l2
N97  c3
N98  l3
N99  c1
N100 Z
N101 G20 G40 l1
N102 E26=E26+60
N103 (ERP)
N104 (URT,O)
N109 G X Y Z10 M30
```

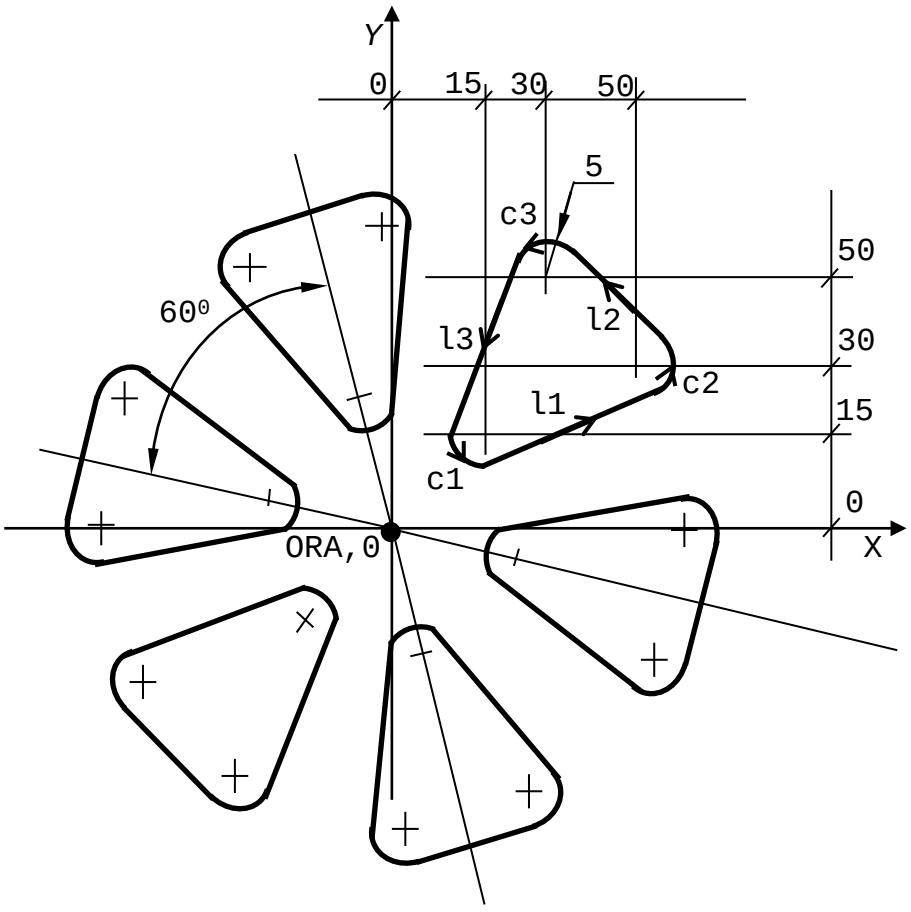


Рисунок 2.117

2.13.2 Трёхбуквенные операторы, изменяющие последовательность выполнения программы

К этому классу принадлежат следующие операторы:

RPT - повторить последовательность кадров;
ERP - определить конец повторяющейся последовательности кадров;
CLS - вызвать подпрограмму для выполнения;
EPP - выполнить подпрограмму;
BNC
BGT
BLT
BEQ - операторы переходов.
BNE
BGE
BLE

2.13.2.1 Повторение частей программы - RPT

При помощи операторов **RPT** и **ERP** часть программы, которая следует за кадром **RPT** и которая заканчивается кадром (**ERP**), повторяется определённое количество раз.

Формат:

(RPT,n) ,

где:

n - число повторений; должно быть целым числом от 0 до 99, может быть представлен параметром **E** типа байт (от E0 до E9). Схема на рисунке 2.118 иллюстрирует, как может быть определён цикл обработки.

Пример

```
(RPT,8)
.......
.......
(RPT,10)
.......
.......
(RPT,5)
.......

(ERP)
......
......
(ERP)
......
......
(ERP)
```

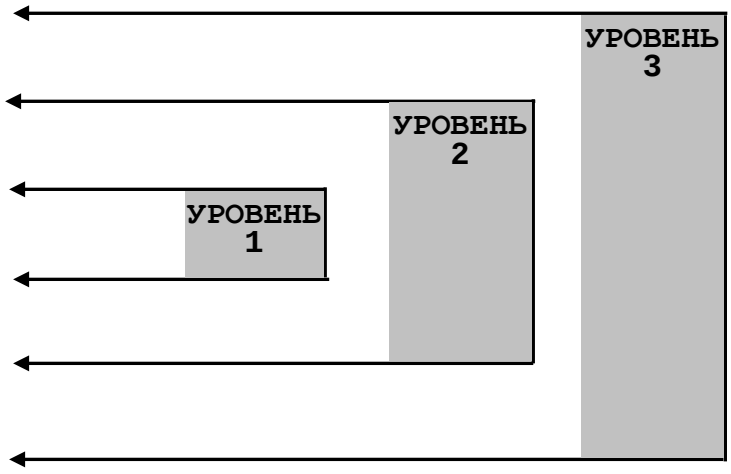


Рисунок 2.118

Допущено три уровня повторений, т.е. можно программировать до двух операторов RPT внутри другого оператора RPT.

2.13.2.2 Использование подпрограммы - CLS

Оператор CLS позволяет вызвать и выполнить программу (подпрограмму), находящуюся в памяти. Под подпрограммой понимаем последовательность кадров, которые определяют цикл обработки. Подпрограмма может быть вызвана из основной программы.

Формат:

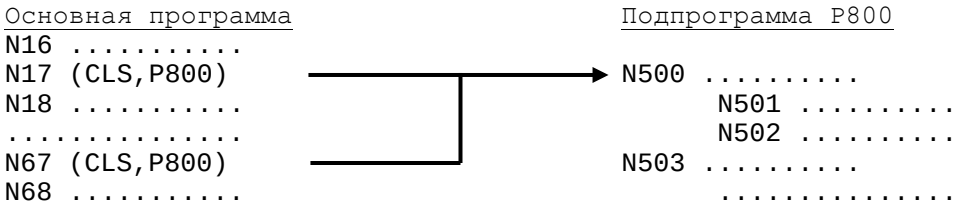
(CLS, ФАЙЛ [/ДИРЕКТОРИЙ]) ,

где:

ФАЙЛ - название программы для вызова;
ДИРЕКТОРИЙ - название устройства памяти, которое содержит программу. Если определение устройства памяти в команде отсутствует, применяется то, которое объявлено для данного процесса по умолчанию в секции 4 файла PGCFIL инструкции NDD.

Синтаксические обязательства - название программы состоит из алфавитно-цифровой последовательности и может иметь максимально шесть символов. Название программы отделяется от названия устройства символом «/». Устройство заменяется последовательностью двух или трёх алфавитно-цифровых символов. Алфавитные символы, которые используются для параметров **ФАЙЛ** и **УСТРОЙСТВО**, могут быть только заглавными; первым символом должна быть буква.

Пример
N1 (CLS,P800/MP2) - Передаёт управление подпрограмме P800, расположенной в директории MP2 (если MP2 характеризована по умолчанию, название не записывается):



Допустимы два уровня вызова. Программа, вызванная с CLS, может, в свою очередь, вызывать другие программы, в то время как эти программы уже не могут вызвать другие программы.
Подпрограммы могут быть параметрическими. Цифровые значения параметров определяются в основной программе в момент вызова.

Пример использования подпрограмм приведён на рисунке 2.119.

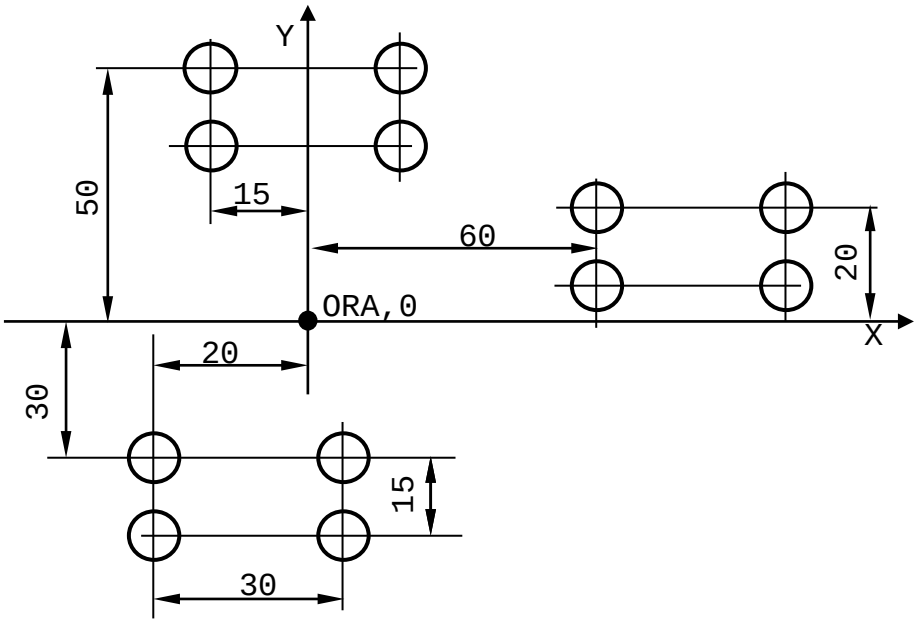


Рисунок 2.119

Основная программа
```
N20 S2000 F180 T2.2 M6 M3
N21 (UOT,0,X-20,Y-25)
N22 (CLS,S600)
N23 (UOT,0,X-15,Y-50)
N24 (CLS,S600)
N25 (UOT,0,X60,Y20)
N26 (CLS,S600)
N27 Z..
```

Подпрограмма S600
```
N501 G81R..Z..
N502 XY
N503 Y-15
N504 X30
N505 Y
N506 G80
```

2.13.2.3 Выполнение части программы - EPP

Оператор EPP выполняет часть программы, заключённую между двумя метками, определёнными в операторе.

Формат:

(EPP, МЕТКА1,МЕТКА2) **,**

где:

МЕТКА1,МЕТКА2 - являются метками, которые ограничивают часть программы, требуемую для вызова и выполнения;

МЕТКА - алфавитно-цифровая последовательность, состоящая максимально из шести символов и заключённая в знак « » (кавычки); должна быть запрограммирована перед номером кадра и после символа «/» в случае его программирования.

Пример

```
.........
"START"N25              - первый кадр с меткой
.........
.........
"END"N100               - последний кадр с меткой
.........
N150(EPP,START,END)     - система выполняет кадры с N25 до N100
.........
```

После выполнения оператора программа продолжается от кадра, следующего за оператором EPP. Невозможно программировать код EPP, который при выполнении встречает другой код EPP.

Этот оператор может быть применён, например, при контурной обработке, когда для чистовой и черновой обработки используется одна и та же последовательность кадров. При черновой обработке необходимо программировать оператор припуска UOV.

В операциях позиционирования «от точки к точке» можно программировать все точки, на которых, например, осуществляется сверление, и тогда использовать EPP для вызова разных инструментов для выполнения различных операций.

2.13.2.4 Переходы внутри программы

Можно программировать внутри программы переходы к кадру, содержащему поле **МЕТКИ**. Переходы могут быть независимыми или зависимыми от параметров **E**, сигналов логики станка или цифровых величин. Операторы переходов приведены в таблице 2.10.

Таблица 2.10 - Операторы переходов

Формат	Функция
(BNC,METKA)	Переход к кадру с меткой (METKA) безусловно
(BGT,VAR1,VAR2,METKA)	Переход если VAR1 больше VAR2
(BLT,VAR1,VAR2,METKA)	Переход если VAR1 меньше VAR2
(BEQ,VAR1,VAR2,METKA)	Переход если VAR1 равен VAR2
(BNE,VAR1,VAR2,METKA)	Переход если VAR1 отличен от VAR2
(BGE,VAR1,VAR2,METKA)	Переход если VAR1 больше или равен VAR2
(BLE,VAR1,VAR2,METKA)	Переход если VAR1 меньше или равен VAR2

Обозначения в таблице 2.14:

VAR1 и VAR2 - являются переменными, на базе которых проверяется отношение; могут быть параметрами, сигналами логики станка, глобальными переменными системы, цифровыми значениями или последовательностью символов;

МЕТКА - для него действительны те же правила, которые были описаны для кода EPP.

Пример

```
N10 (BGT,E1,123,END)      - переход к END, если значение параметра
                            E1 больше 123.
N20 (BEQ,SA3,1,LAB1)      - переход к LAB1, если булевская перемен-
                            ная SA3 включена.
```

```
N30 (BNE,E1,E5,START)          - переход к START, если значение параметра
                                 E1 отлично от значения E5.
N40 (BEQ,SYVAR1.2CH,"OK",LAB1) - переход к LAB1 если символы SYVAR1 - O K.
```

В случае, если переменная имеет формат символа (CH), объектом проверки будет последовательность символов, длина которой определяется индексом, который предшествует CH. Если индекс не определён, его значение по умолчанию принимается равным «1».

Пример

```
(BEQ,SYVAR2.3CH,"ABC",END)
```
- переход к метке END, если три символа из SYVAR2 являются символами ABC.

Если переменные VAR1 и VAR2 имеют формат LR или RE, необходимо определить десятичный порог, ниже которого обе переменных можно считать равными, и проверить, является ли разница между переменными меньше порога. Следует иметь в виду, что при таком формате любая математическая операция между переменными имеет ошибку округления процессора, которая накапливается после каждой операции.

Пример

Переходить к метке LAB1, если E42=E41, при помощи сравнения разницы между ними относительно порога (при помощи BLT):

```
E42=0.021
E41=0.015
E47=0.0015
E43=0.0001              - порог = 0.0001
"LAB2" E41=E41+E47
E44= ABS (E42-E41)
(BLT, E44, E43, LAB)    - переходить к метке LAB1, если E44=E42-E41<E43
(BBS, LAB2)
"LAB1" (DIS, E41)
```

2.13.3 Примеры программирования

2.13.3.1 Примеры использования кода RPT

1) Пример обработки одинаковых циклов, повторяющихся на равном расстоянии, приведён на рисунке 2.120.

```
(DIS," N 3 паза")
(DIS,".........")
N1  S600 T6.6 M6 M3
N2  (RPT,3)
N3  X40 Y35
N4  Z2
N5  (RPT,2)
N6  G91 Z-8
N7  G90 G1 G41 X40 Y20 F300
N8  X60
N9  Y50
N10 X20
N11 Y20
N12 G40 X40
N13 Y35 F1000
N14 (ERP)
N15 G Z2
N16 (UIO,X80,Y20)
N17 (ERP)
N18 (UAO,0)
N19 Z20
N20 XYM 30
```

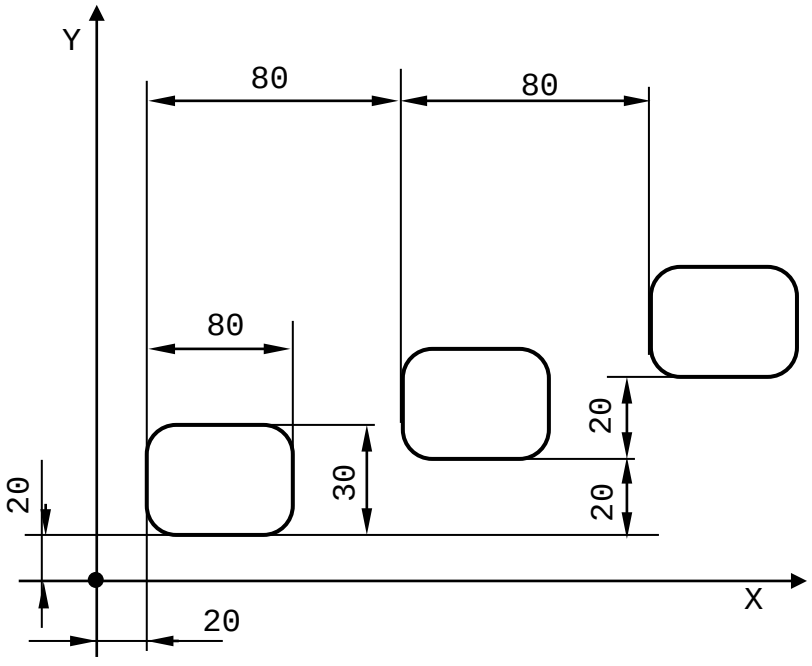


Рисунок 2.120

2) Пример обработки отверстий, расположенных на одинаковом расстоянии друг от друга, приведён на рисунке 2.121.

```
(DIS," отверстия ")
N1  F200 S900 T1.1 M6
N2  G81 R5 Z-10 M3
N3  X10 Y10
N4  (RPT,7)
N5  G91 X10
N6  (ERP)
N7  Y40
N8  (RPT,7)
N9  X-10
N10 (ERP)
N11 G80 G90 M5
```

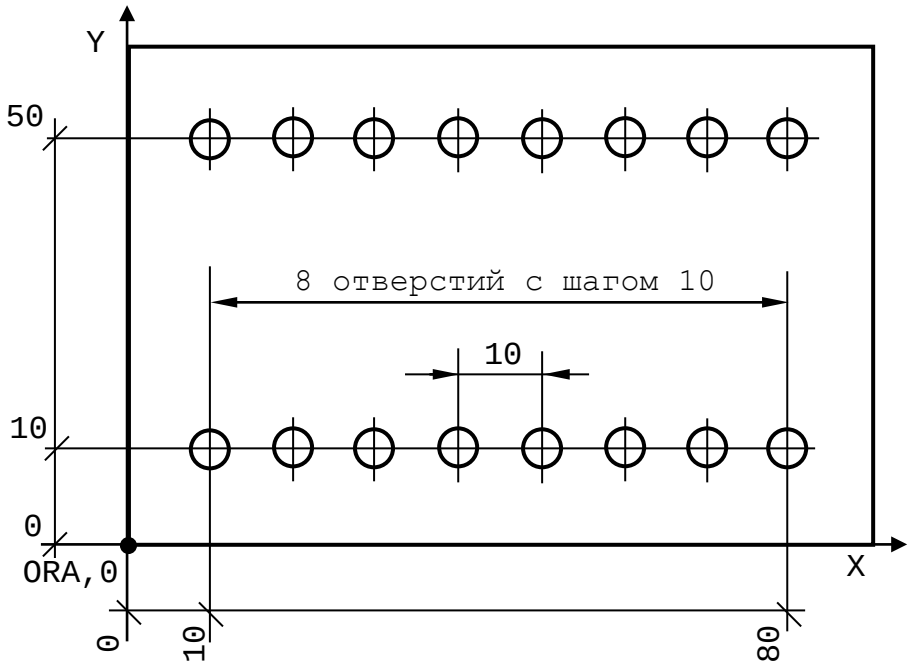


Рисунок 2.121

3) Пример использования кода RPT для обработки с черновым и чистовым проходом приведён на рисунке 2.122.

```
(DIS,".........")
N1  S350 T6.6 M6
N2  X60 Y M3
N3  Z-50
N4  UOV=0.5
N5  (RPT,2)
N6  G1 G41 X60 Y60 F500
N7  G3 Y-60 I60 J
N8  G1 X100
N9  G3 Y60 I100 J
N10 G1 X60
N11 UOV=0
N12 (ERP)
N13 GZ 20 M5
N14 X Y M30
N14 X Y M M30
```

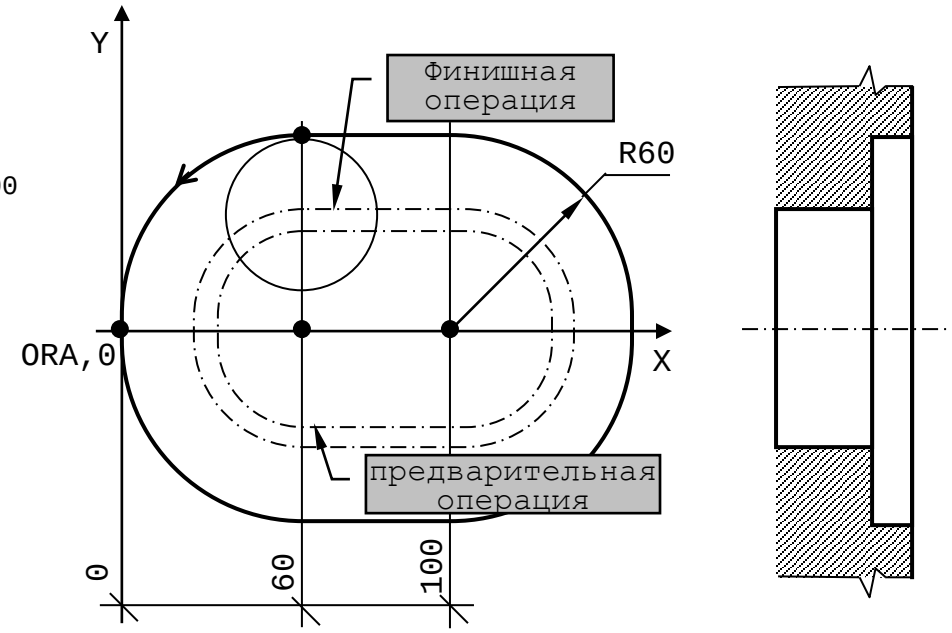


Рисунок 2.122

2.13.3.2 Примеры использования оператора EPP

1) Пример использования оператора EPP в операциях фрезерования приведён на рисунке 2.123.

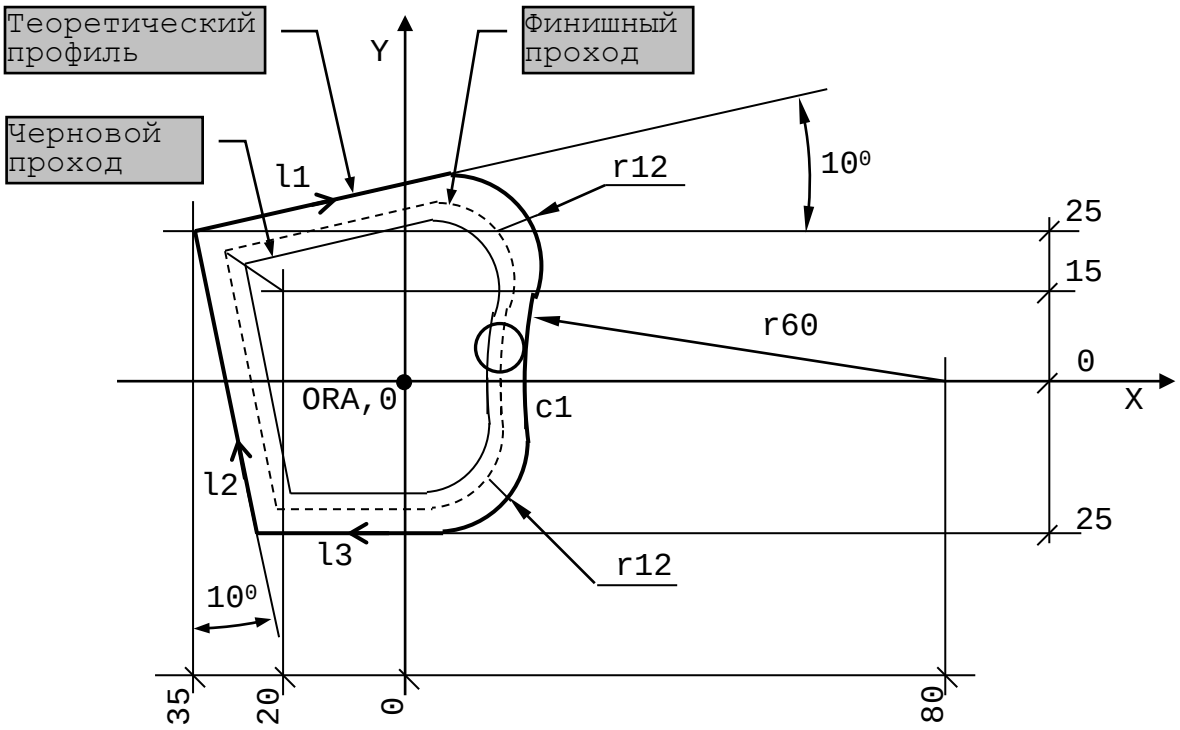


Рисунок 2.123

```
N1  l1=X-35 Y25,a10
N2  l2=X-35 Y25,a100
N3  l3=X Y-25, a180
N4  c1=180 JO r60
N5  (DIS,"ЧЕРНОВОЙ ПРОХОД ФРЕЗА D=8")
N6  F400 S2000 T4.4 M6 M3
N7  UOV=1
"START" N8
N9  G X-20 Y15
N10 Z-10
N11 G21 G42 12
N12 l1
N13 r-12
N14 c1
N15 r-12
N16 l3
N17 l2
N18 G20 G40 l1
N19 G X-20 Y15
N20 Z
"END" N21
N22 (DIS,"ФИНИШНЫЙ ПРОХОД ФРЕЗА D=8")
N23 F500 S2500 T5.5 M6 M3
N24 UOV=0
N25 (EPP,START,END)
N26 .............
```

2) Пример использования оператора EPP в операциях «от точки к точке» приведён на рисунке 2.124.

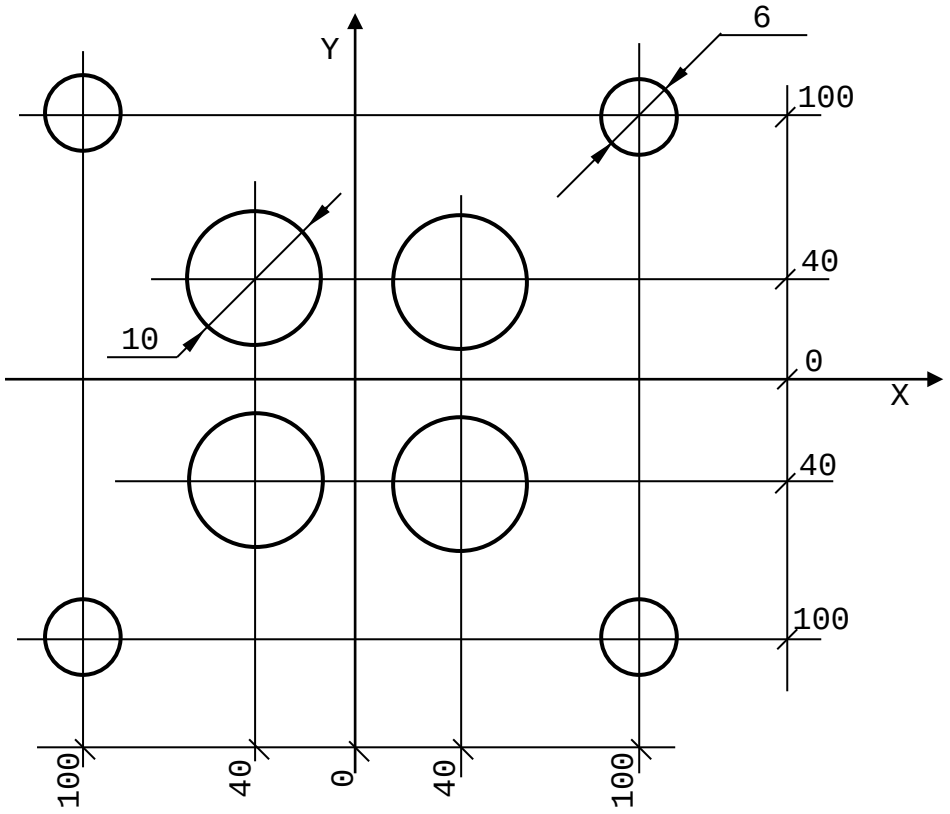


Рисунок 2.124

```
N1  (DIS,".....................")
N2  F300 S2000 T1.1 M6 M3
N3  G81 R3 Z-4
"D6" N4
N5  X100 Y100
N6  X-100
N7  Y-100
N8  X100
"D10" N9
N10 X40 Y40
N11 X-40
N12 Y-40
N13 X40
"END" N14
N15 G80
N16 (DIS,".....................")
N17 F200 S1800 T2.2 M6 M3
N18 G81 R3 Z-22
N19 (EPP,D6,D10)
N20 G80
N21 (DIS,".....................")
N22 F220 S1600 T3.3 M6 M3
N23 G81 R3 Z-24
N24 (EPP,D10,END)
N25 G80
N26 ..........................
```

3) Пример использования оператора EPP во время выполнения одинаковых фигур, имеющих различную ориентировку в плоскости, приведён на рисунке 2.125.

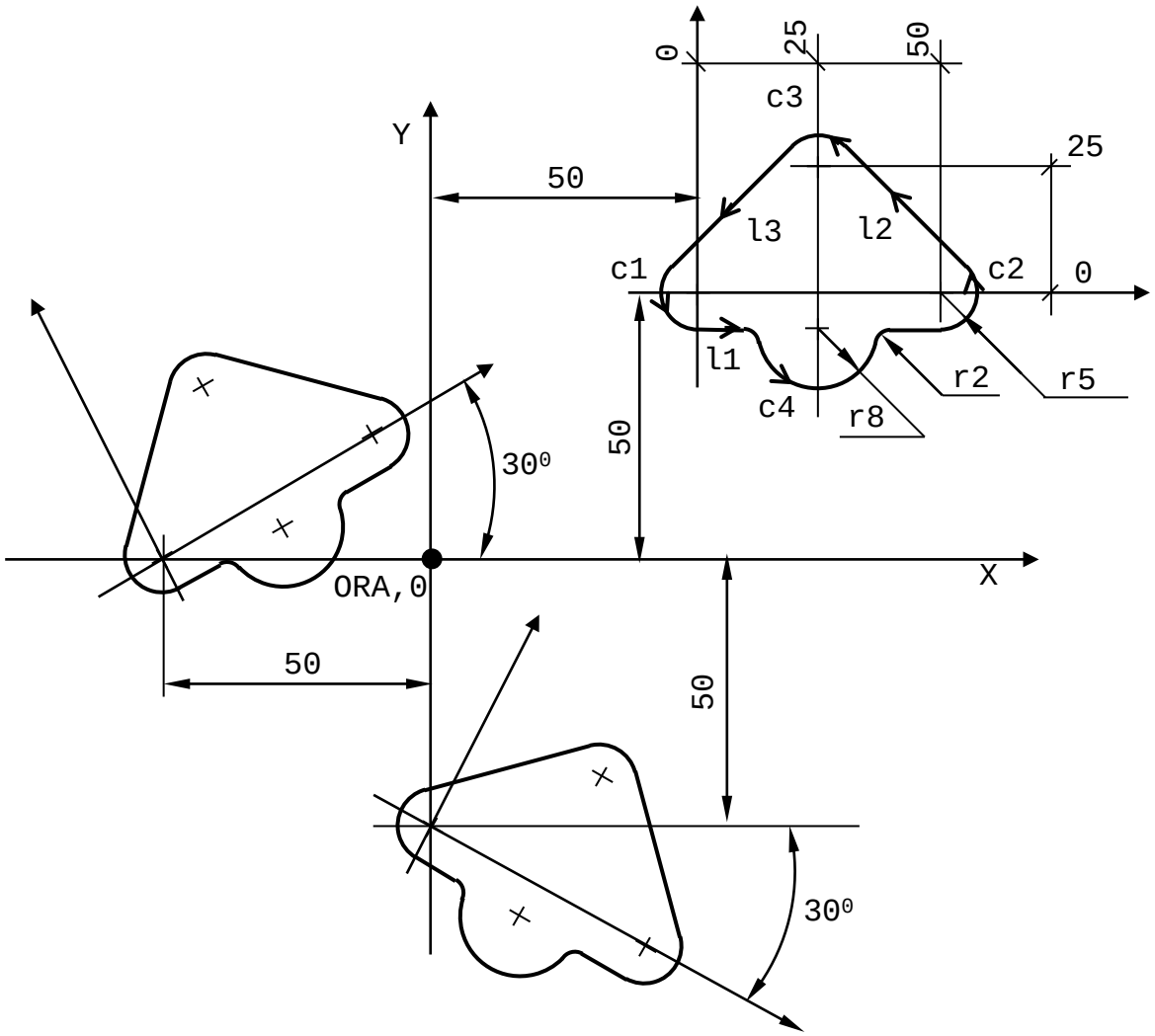


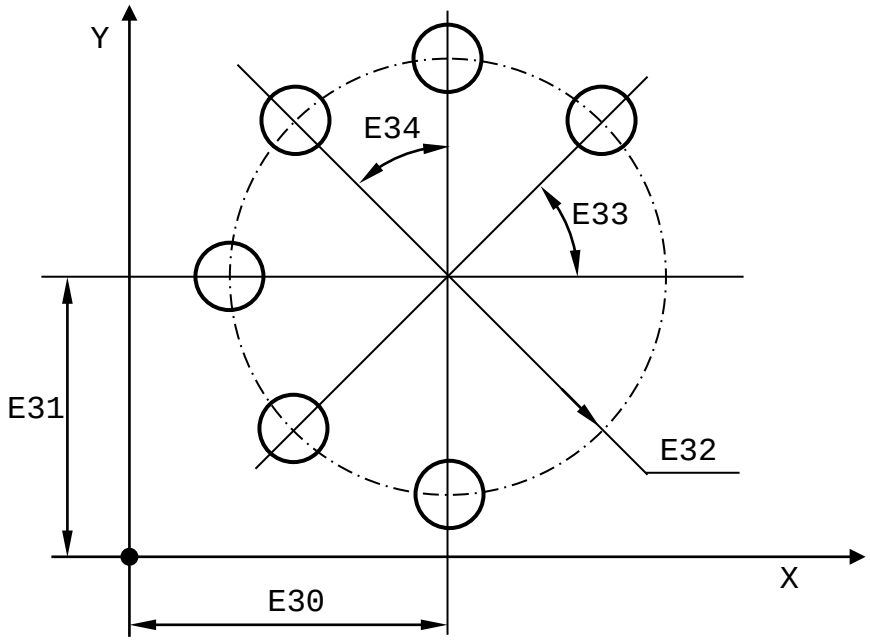
Рисунок 2.125

```
N32 F200 S1200 T1.1 M6 M3
N33 c1=I J r5
N34 c2=I50 J r5
N35 c3=I25 J25 z5
N36 l1=c1,c2
N37 l2=c2,c3
N38 l3=c3,c1
N39 c4=I25 J-5 z8
N40 (UOT,0,X50,Y50)
"1" N41
N42 G21 G41 c1
N43 Z-10
N44 l1
N45 Z-1
N46 c4
N47 Z-2
N48 l1
N49 c2
N50 l2
N51 c3
      N52 l3
      N53 c1
      N54 Z2
      N55 G20 G40 l1
      "Z" N56
      N57 (UOT,0,X-50,Y)
N58 (URT,30)
      N59 (EPP,1,2)
N60 (UOT,0,X,Y-50)
N61 (URT,-30)
      N62 (EPP,1,2)
      N63 (URT,0)
      N64 (UAO,0)
      N65 GZ 2
      ..................
```

2.13.4 Параметрические подпрограммы

Подпрограммы, т.е. последовательность кадров, которые определяют цикл обработки, вызываемые из основной программы, являются параметрическими в случае, если геометрические и технологические данные цикла обработки, т.е. величины, которые должны быть приданы различным функциям **G, S, X, Y, Z** и т.д., заданы через параметры, значения которых определены в основной программе.

Пример программирования круговой сетки с использованием подпрограммы приведён на рисунке 2.126.




```
E1 = номеру отверстия
E2 = G   постоянный цикл
E30 =  X   центр решетки
E31 =  Y   центр решетки
E32 =  радиус решетки
E33 =  начальный угол
E34 =  угловой шаг
E36 =  размер  R
E37 =  размер  Z
```

Рисунок 2.126 - Пример программирования круговой сетки

```
N43 (DIS,".............")
N44 S900 F70 T2.2 M6 M3
N45 E30=50
N46 E31=50
N47 E32=30
N48 E33=45
N49 E34=45
N50 E1=6
N51 E36=2
N52 E37=-10
N53 E2=81
N54 (CLS,SUB800)
N55 G Z20 M5
.............
```

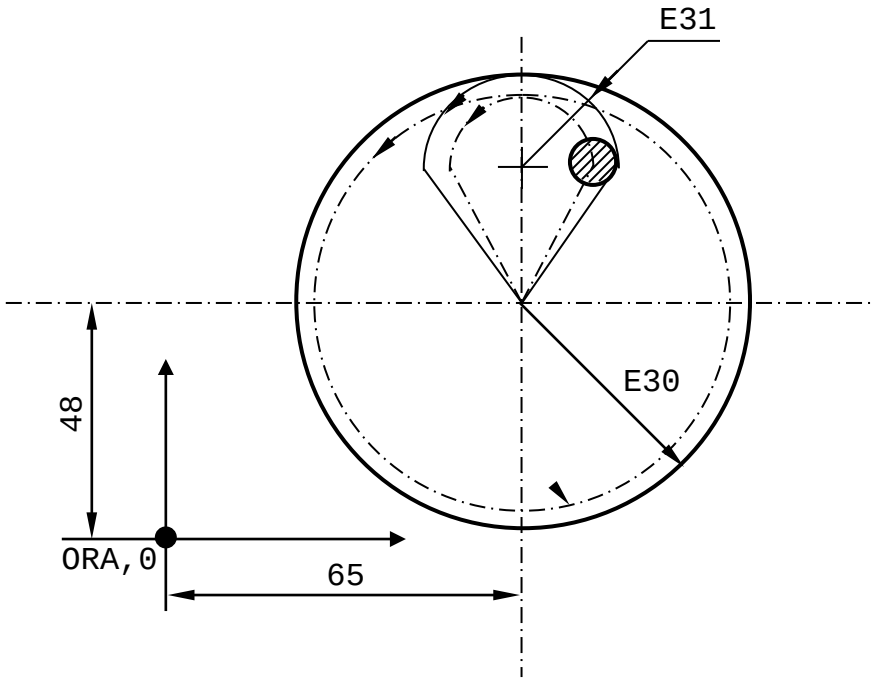
```
Подпрограмма SUB 800
N799 (DIS," круговые решения ")
N800 (UOT,0,XE30,YE31)
N801 E25=E33
N802 (RPT,E1)
N803 (URT,E25)
N804 GE2 RE36 ZE37
N805 XE32 Y
N806 E25=E25+E34
N807 G80
N808 (ERP)
N809 (UAO,0)
N810 (URT,0)
```

Примечание - Присвоение параметров различным геометрическим и технологическим функциям описано в п.2.12 «Параметрическое программирование».

2.13.4.1 Примеры параметрических подпрограмм

Примеры параметрических подпрограмм проиллюстрированы рисунками 2.127-2.129.

1) **Пример** использования параметрических подпрограмм при фрезерно-расточных операциях приведён на рисунке 126.



E30 = радиус отверстия
E31 = радиус соединения
E25 = подача работы

Рисунок 2.127

Основная программа

```
.............
N128 (DIS,".............")
N129 S330 T12.12 M6
N130 (UOT,0,X65,Y48)
N131 G X Y M3
N132 Z-30
N133 E30=62.5
N134 E31=25
N135 E25=160
N136 (CLS,SUB500)            вызов подпрограммы
N137 (UAO,0)
N138 G Z 20
.............
```

Подпрограмма SUB 500

```
N500 E35=E30-E31
N501 G1 G41 X E31 YE35 F2000
N502 G3 X YE30 I JE35 FE25
N503 I J
N504 E31=NEG(E31)
N505 G40 XE31 YE35 I JE35
N506 G1 X Y F2000
```

Примечание - Когда программируется стандартный профиль и, следовательно, параметр E30 и радиус отверстия, E31 должен быть всегда больше радиуса фрезы.

2) Линейная сетка. Этому примеру соответствует рисунок 2.128.

Основная программа

```
N30 S1000 F60 T4.4 M6 M3
N31 E25=30
N32 E30=12
N33 E31=12
N34 E32=20
N35 E1=9
N36 E34=2
N37 E35=-3
N38 E2=81
N39 (CLS,SUB600)
N40 (DIS,".........")
N41 S880 F100 T5.5 M6 M3
N42 E35=-20
N43 (CLS,SUB600)
.......................
```

Подпрограмма SUB600

```
N600 (UOT,0,XE30,YE31)
N601 (URT,E25)
N602 E3=E3-1
N603 GE2 RE34 ZE 35
N604 XY
N605 (RPT,E3)
N606 G91 XE32
N607 (ERP)
N608 G90
N609 G80
N610 (UAO,0)
N611 (URT,0)
```

Примечание - В кадрах N31 и N38 присваивается цифровое значение каждому параметру для операций центрирования. В кадрах N39 и N42 вызывается подпрограмма SUB 600.

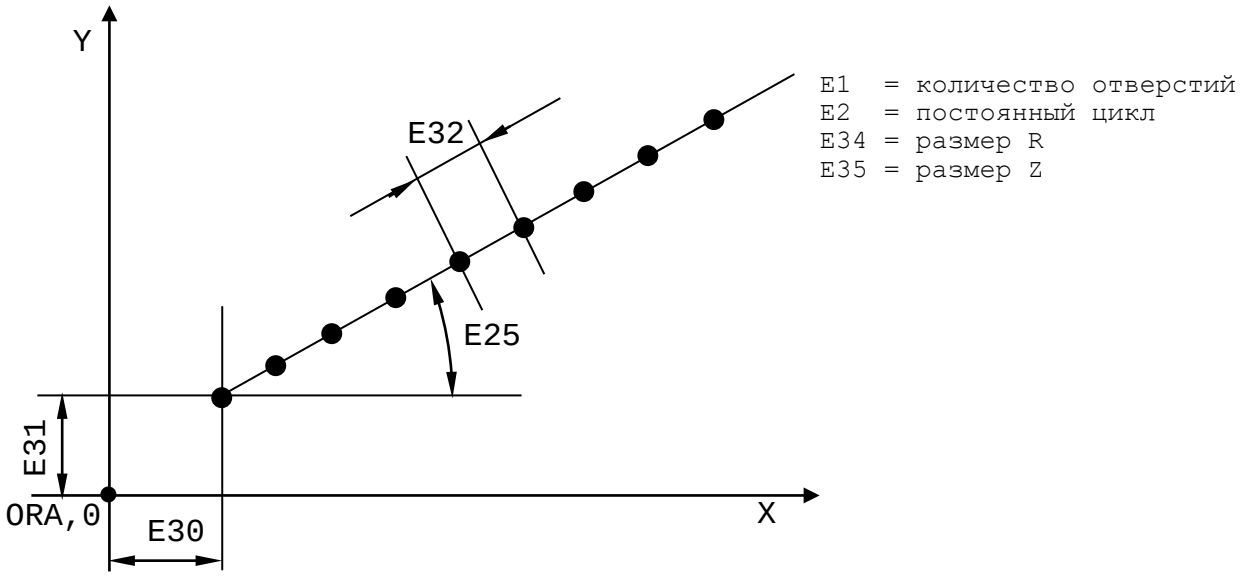


Рисунок 2.128

3) Решётка с отверстиями. Этому примеру соответствует рисунок 2.129.

Основная программа

```
N20  (DIS,".............")
N30  F9000 S3000 T2.2 M6
N40  G X20 Y25 M3
N50  G81 R3 Z-20
N60  E25=30
N70  E26=80
N80  E31=15
N90  E32=12
N100 E1=8
N101 E2=5
N102 (CLS,SUB700)
.................
```

Подпрограмма SUB 700

```
N700 E33=E31*COS(E25)
N701 E34=E31*SIN(E25)
N702 E41=0
N703 E42=0
N704 E5=E1-1
N705 (RPT,E2)
N706 G91 XE41 YE42
N707 (RPT,E5)
N708 XE33 YE34
N709 (ERP)
N710 E41=E32*COS(E26)
N711 E42=E32*SIN(E26)
N712 E33=NEG(E33)
N713 E34=NEG(E34)
N714 (ERP)
N715 G80 G90
```

Примечание - В кадрах N60 - N101 присваивается цифровое значение каждому параметру. В кадре N102 вызывается подпрограмма SUB700.

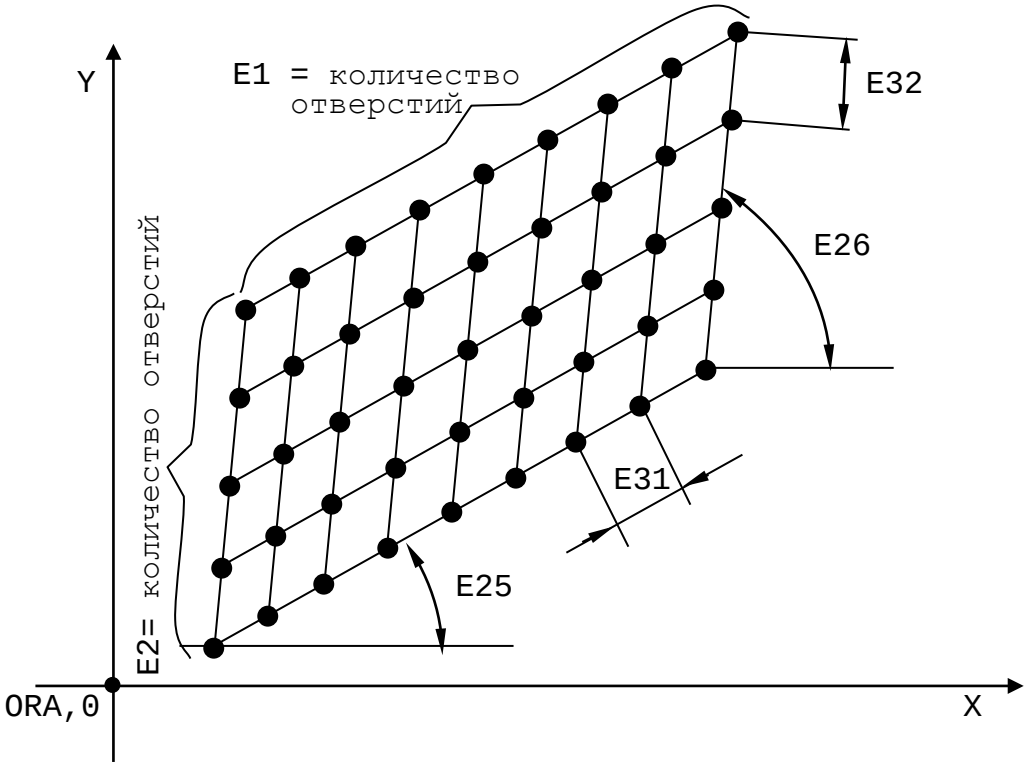


Рисунок 2.129

2.13.4.2 Примеры использования отводов, зависимых от параметров

Примеры использования отводов, зависимых от параметров, приведены на рисунках 2.130-2.131.

1) Выполнение паза. Этому примеру соответствует рисунок 2.130.

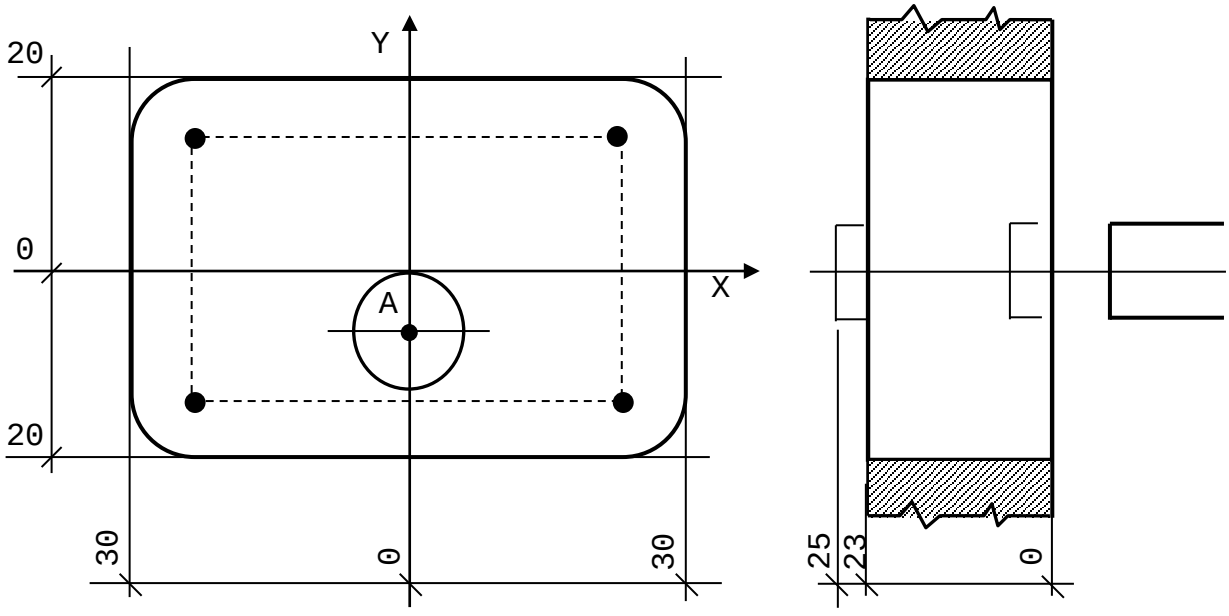
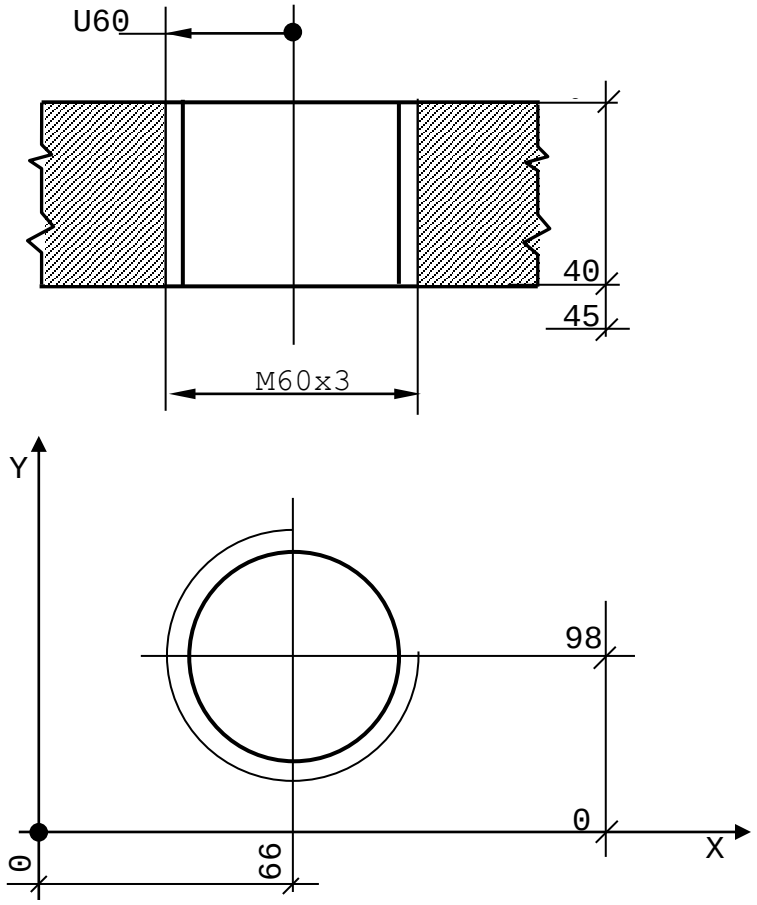


Рисунок 2.130

```
N1  (DIS,".....................")
N2  F500 S2000 T1.1 M6 M3
N3  E31=-3.5
N4  E32=-24
"START" N5
N6  G X Y-10
N7  Z E31
N8  G1 G42 X Y-20
```

```
N9  X-30
N10 Y20
N11 X30
N12 Y-20
N13 G40 X
N14 Y-10
"END" N15
N16 E31=E31-3.5
N17 (BGT,E31,E32,START)
N18 E31=-25
N19 (EPP,START,END)
N20 G Z10
N21 X Y
.....................
```

2) Выполнение цилиндрического нарезания резьбы. Этому примеру соответствует рисунок 2.131.



E30 = диаметр первого прохода
E31 = глубина прохода (по приращению)(диаметр.)
E32 = диаметр возврата
E33 = конечный диаметр

Рисунок 2.131

```
N1  (DIS,".............")
N2  S150 T5.5 M6
N3  G X66 Y98 Z5 M3
N4  E30=56.8
N5  E31=0.5
N6  E32=50
N7  E33=60
"I" N8
N9  G Z5
N10 UE30
N11 G33 Z-45 K3
N12 GUE32
N13 E30=E30+E31
N14 (BGT,E30,E33,F)
N15 (BNC,I)
```

```
"F" N16
N17 GUE32
N18 Z5
N19 UE33
N20 G33 Z-45 K3
N21 GUE32
N22 Z20
.................
```

2.13.5 Трёхбуквенные операторы смешанного типа

К этому классу принадлежат следующие операторы:

DPI - определение плоскости интерполяции;
DTL - определение величины допускаемого отклонения позиционирования;
DLO - определение рабочего поля;
CTL - переключение в режим токарного или фрезерного станка;
DSA - определение защищенных зон.

2.13.5.1 Определение плоскости интерполяции - DPI

Оператор DPI определяет абсциссу и ординату плоскости интерполяции.
Формат:

(DPI,VAR-1,VAR2) ,

где:
VAR-1 и **VAR2** - являются буквенными символами, соответствующими одному из возможных названий осей системы.

Пример
(DPI,X,A) - плоскость интерполяции, образованная осями X и A.

Две буквы должны определять два различных названия осей. Невозможно использовать пару осей, альтернативных между собой, т.е. функционально эквивалентных. Не допускается использование оператора DPI, если активизированы следующие функции:

- GTL (G21);
- компенсация радиуса инструмента (G41-G42);
- постоянные циклы (G81-G89);
- операции непрерывной обработки (G27-G28).

2.13.5.2 Определение величины допуска при позиционировании- DLT

Оператор DTL определяет для запрограммированных осей величину допускаемого отклонения позиционирования. Если запрограммированной величиной является 0, то принимается значение, объявленное в файле характеризации. Для незапрограммированных осей сохраняется значение, которое было активным ранее.
Формат:

(DTL,VAR-1 [,VAR-2...,VAR-n]) ,

где:
VAR-i - операнд типа «ось-размер».

Пример
(DTL,Z.1,X.05).

Невозможно программировать два операнда с одинаковым названием оси. Максимальное число операндов - 6. Оператор DTL вызывает ошибку, если активизированы следующие функции:

- GTL (G21);
- компенсация радиуса инструмента (G41-G42);
- непрерывная обработка (G27-G28).

Размеры допускаемого отклонения должны быть запрограммированы в системе измерения (G70/G71), активной в момент выполнения трёхбуквенного кода. Значение допускаемого отклонения позиционирования не должно превышать текущего значения «ОШИБКА ПРИВОДА».

2.13.5.3 Определение рабочего поля - DLO

Оператор DLO определяет рабочее поле для осей, запрограммированных в операторе, с учетом фактической начальной точки. Для незапрограммированных осей сохраняется значение, которое было активным ранее. Если запрограммированное значение превышает предел, объявленный в файле характеризации, оно не учитывается и значение, объявленное в характеризации принимается за допустимое.

Формат:

(DLO,VAR-i) ,

где:

VAR-i - определяет пару слов типа «ось-размер», имеющих одинаковое название оси; представляют соответственно верхний и нижний пределы рабочего поля в отношении текущей начальной точки.

Значения ограничения перемещений для активной зоны, заданные в DLO, должны находиться внутри зоны ограничения перемещений, заданных при характеризации для соответствующей зоны оси.

Может быть запрограммировано максимально 6 пар различных рабочих пределов. Размеры, приданные операндам осей, должны быть запрограммированы в системе измерения (G70/G71), которая активна в момент выполнения трёхбуквенного кода DLO. Действие команды DLO может быть отменено при помощи кнопки «СБРОС».

Пример определения рабочего поля приведён на рисунке 2.132.

```
(DLO,X-300 X300)
..............
(DLO,Y-200 Y200)
..............
```

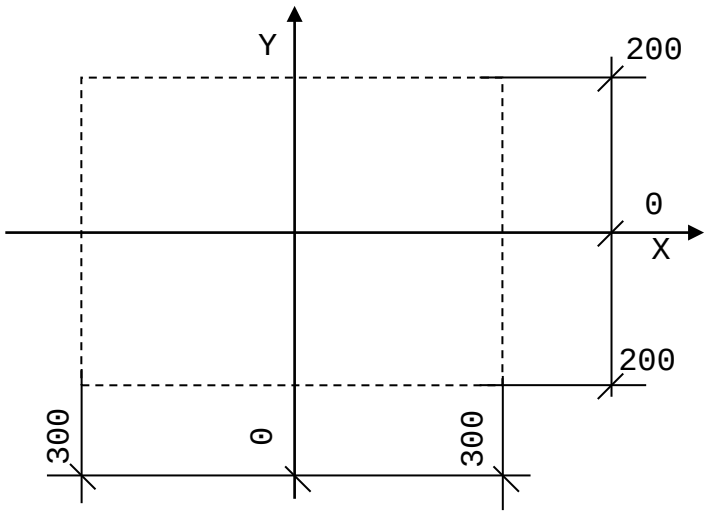


Рисунок 2.132

2.13.5.4 Переключение между конфигурациями токарного и фрезерного станка

Команда CTL позволяет выбирать конфигурацию фрезерного станка, если обе возможности установлены в файле PGCFIL.

После команды CTL управление снабжено всеми характеристиками фрезерного варианта.

Допустимый формат:

(CTL, T) - активизирует конфигурацию фрезерного варианта,

или

(CTL) - возвращает управление начальной конфигурации фрезерного варианта.

Режим «СБРОС» восстанавливает конфигурацию фрезерного варианта.

2.13.5.5 Защищённые зоны - DSA, ASC, DSC

Эти команды позволяют активизировать или деактивизировать защищённые зоны, т.е. зоны, куда вход координат запрещён.

Команды этого класса:

1) **DSA** - определяет защищённую зону;
2) **ASC** - активизирует защищённую зону;
3) **DSC** - деактивизирует защищённую зону.

Управление производит проверку относительно защищённых зон до начала движения. Из программы можно определить до трёх защищённых зон относительно текущей начальной точки.

Допустимые форматы:

(DSA, n, ОСЬ1- ОСЬ1+, ОСЬ2- ОСЬ+) ,
(ASC, n) ,
(DSC, n) ,

где:

n - номер зоны;
ОСЬ1- - имя оси и значение нижней границы по ОСИ1;
ОСЬ1+ - имя оси и значение верхней границы по ОСИ1;
ОСЬ2- - имя оси и значение нижней границы по ОСИ2;
ОСЬ2+ - имя оси и значение верхней границы по ОСИ2.

Защищённые зоны могут быть отменены нажатием кнопки «СБРОС».

Пример применения защищённых зон приведён на рисунке 2.133.

```
N1  (DSA, 1,X-300 X-100, Y-100Y100)
N2  (DSA, 2,X200X450,Y-50Y50)
N3  (ASC, 1)
N4  (ASC, 2)
N5  T1.1 M6
------------
N80 (DSC, 1)
------------
N99 M30
```

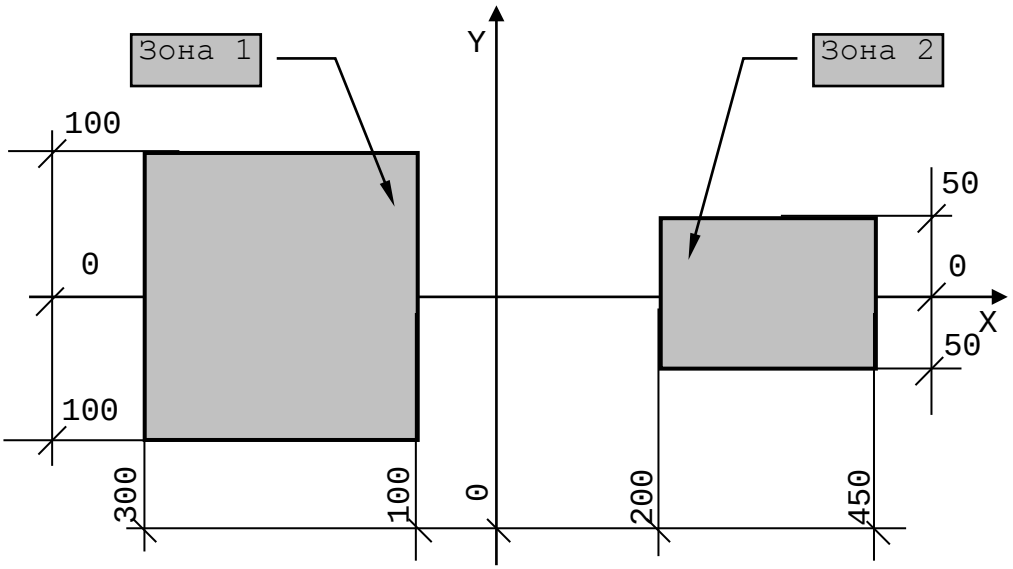


Рисунок 2.133

2.13.6 Трёхбуквенные операторы ввода/вывода

Эти команды позволяют выполнять операции входа/выхода из программы. Команды этого класса:

1) **DIS** - вывод переменной на экран;
2) **DLY** - установка задержки.

2.13.6.1 Вывод переменной на экран - DIS

Команда DIS позволяет вывести на экран значение, определяемое переменной. Желаемое значение появляется в области экрана, предназначенной для связи с оператором.

Формат:

(DIS, VAR) ,

где:

VAR может быть:

- любой код, используемый в кадрах присваивания для глобальных системных переменных (геометрических элементов, параметров); управление выводит на экран последовательность **Имя Переменной = значение**;
- сообщение для оператора; сообщение может иметь длину до 32 символов, при этом сообщение программируется в кавычках в команде DIS. Например:
  - (DIS, "ЭТО ПРИМЕР");
  - цифровая константа, например: (DIS, 100).

Примеры

1) (DIS, "c1=",c1) - выводит на экран координаты центра и радиус окружности.
2) (DIS, "l2=",l2) - выводит на экран расстояние между начальной точкой и прямой линией, а также угол, образованный прямой линией и абсциссой оси.

2.13.6.2 Выдержка времени - DLY

Оператор DLY позволяет программировать выдержку времени с определенной продолжительностью.

Формат:

(DLY, время) ,

где:

время - выражает время остановки, выраженное в сек. (макс. 32 секунды); может быть выражен как цифровая постоянная или параметр **E** типа действительное с двойной точностью (E30:...).

Этот оператор нуждается в синхронизации (символ «#» в кадре).

Пример

```
(DLY,2)
E48=2
(DLY,E48)
```

2.13.7 Управление графическим дисплеем

Этот класс команд позволяет управлять графическим дисплеем из управляющей программы. Имеются следующие команды:

1) **UCG** - определить поле графического дисплея;
2) **CLG** - стереть графический дисплей;
3) **DCG** - блокировать графический дисплей.

2.13.7.1 Определение поля графического дисплея

Эта команда инициализирует графический дисплей и устанавливает пределы, масштаб и режим дисплея.

Формат:

(UCG, n, ОСЬ1I ОСЬ1S, ОСЬ2I ОСЬ2S[,ОСЬ3])	- плоская графика движения инструмента;
(UCG, n, ОСЬ1I ОСЬ1S, ОСЬ2I ОСЬ2S[,ОСЬ3I ОСЬ3S,A,S])	- плоская графика движения инструмента по трём координатам;
(UCG, n, ОСЬ1L ОСЬ1R, ОСЬ2L ОСЬ2R[,ОСЬ3L ОСЬ3R,D,Q])	- 3D-графика ,

где:

n - определяет режим дисплея:

- **n=1** - вывод на экран, не координированный с движением осей;
- **n=2** - вывод на экран, координированный с движением осей; n может быть программировано или прямо или косвенно (параметр E типа байт);
- **n=3** - вывод на экран не координированный с движением осей (точки зелёного цвета) и координированный с движением осей (точки розового цвета) одновременно.

ОСЬ1I - определяет ось и размер для низшего предела абсциссы на экране;

ОСЬ1S - определяет ось и размер для высшего предела абсциссы на экране;

ОСЬ2I - определяет ось и размер для низшего предела ординаты на экране;

ОСЬ2S - определяет ось и размер для высшего предела ординаты на экране;

ОСЬ3 - определяет ось, перпендикулярную плоскости обработки, образованной осями ОСЬ1 и ОСЬ2, для обозначения точки на экране, в которой производится движение по ОСИ3 в режимах UAS=1 или режим дисплея n = 1;

ОСЬ3I - определяет ось и размер для низшего предела апликаты на экране;

ОСЬ3S - определяет ось и размер для высшего предела апликаты на экране;

ОСЬ1L(-1R) ОСЬ2L(-2R) ОСЬ3L(-3R) - определяют форму и размеры заготовки в 3D-графике. Эти параметры имеют различное назначение при определении объемных заготовок тел вращения и поверхностей (см. руководство оператора);

A - определяет значение угла проекции ОСИ2, откладываемое от положительного направления ОСИ1;

S - определяет значение масштаба ОСИ2;

D - определяет горизонтальное или вертикальное расположение аппликаты на экране; данный параметр может принимать два значения:

0 – устанавливает горизонтальное положение оси 3 и инструмента,
1 - устанавливает вертикальное положение оси 3 и инструмента;

Q - определяет степень детализации 3D-графики; данный параметр может принимать значения от минус 1 (минимальное качество) до минус 5 (максимальное качество).

Пример

(UCG, 2, X100 X150, Y50 Y250) - Активизирует графический дисплей режима 2. Графический дисплей показывает перемещение между X100 и X150 и между Y50 и Y250 относительно текущей начальной точки.

Управление UCG может быть запрограммировано в любой среде программирования. Однако, будучи сервисным режимом управления, которое требует значительного времени обработки, оператор UCG должен быть использован с осторожностью во время непрерывной обработки профиля. Если используется внутри программы, то этот оператор нуждается в синхронизации (символ «#» в кадре).

Графический дисплей учитывает программированные, временные или инкрементальные начальные точки. Инструкция UCG должна быть программирована после начальных точек и перед разрешением коррекции инструмента.

Пример

```
N1 (UOT, 0, CZ200)
N2 (UCG, 1, X-100 X20, Y30 Y130)
N3 T1.1 M6
```

2.13.7.2 Сброс графического дисплея - CLG

Оператор CLG осуществляет сброс текущего профиля с экрана дисплея, оставляя на экране лишь декартовые оси.

Формат:

(CLG) .

2.13.7.3 Отмена графического дисплея - DCG

Оператор DCG отменяет вывод графической информации на дисплей.
Формат:

(DCG) .

Эта команда должна быть запрограммирована после команды CLG.

Пример

```
.........
N8  (CLG)
N9  (DCG)
.........
```

2.13.8 Управление коррекцией инструмента - RQU

Эта команда позволяет управлять коррекцией инструмента из управляющей программы. Команда RQU переаттестует (изменяет и модифицирует) определенную коррекцию инструмента в соответствии с программируемыми значениями.
Формат:

(RQU, N.инстр., N.корр.,Z.., K..) ,

где:

N.инстр - номер инструмента является цифровой константой или параметром **E** типа IN (E10-E19);

N.корр - номер корректора, который будет модифицирован; номер корректора находится в диапазоне от 1 до 9999, верхняя граница зависит от количества записей, определяемого в файле коррекции;

Z - определяет инкремент (изменение) длины, который прибавляется к корректору оси Z (значение инкремента может быть задано через E параметр (E30-En));

K - определяет инкремент диаметра инструмента, который прибавляется к корректору диаметра инструмента **K** (значение инкремента может быть задано через E параметр (E30-En)). Если значение инкремента равно «0», текущее значение по длине и диаметру не изменяется.

Пример
(RQU, 10, 1, ZE40; KE41) - модифицирует инструмент 10, корректор 1. Инкремент для оси Z содержится в E40. Инкремент для диаметра **K** содержится в E41.

Если файл корректоров создан для запоминания текущих и максимальных значений коррекции, файл корректоров имеет следующий формат:

номер корректора,Z..,K..,c..,m..,c..,m.. ,

где:

c..,c.. – определяют текущие значения коррекции длины инструмента **Z** и диаметра инструмента **K** соответственно;

m..,m.. – определяют максимальные значения коррекции длины инструмента **Z** и диаметра инструмента **K** соответственно.

Если файл корректоров создан для управления текущим (c) и максимальным (m) значениями корректора, команда RQU изменяет текущее значение (c) (инструмент считается непригодным, если значение (c) превысило значение (m)). Если же вы не хотите изменить текущее значение (c), программируйте RQP с тем же форматом.
Если вы изменяете коррекцию для диаметральной оси, то управление разделит определяемое вами значение на два, перед прибавлением их к корректору.
На значения инкрементов Z, K не действует масштабирование (SCF).
Если инкрементальное значение коррекции не объявлено в цикле измерения, команда RQU требует значка синхронизации «#».

2.13.9 Определение параметров измерения - DPT

Команда **DPT** позволяет определить параметры измерения с пульта или из программы. Параметры, которые необходимо определить:

- размер подхода (**Qa**), мм;
- размер безопасности (**Qs**), мм;
- скорость измерения (**Vm**), мм/мин.

Формат команды **DPT**:

(DPT, Qa, Qs, Vm) - из программы,
DPT, Qa, Qs, Vm - с пульта.

Пример
DPT, 10, 12, 1000 - с пульта.

Когда управление выполняет цикл измерения, оно выполняет следующую последовательность движений в соответствии с рисунком 2.134:

1) движение на быстром ходу к точке подхода (PS);
2) движение на скорости измерения до точки измерения или расстояния безопасности, затем хранение размеров;
3) возврат на быстром ходу до точки начала цикла измерения.

Если щуп не переключается до достижения точки безопасности, он возвращается к начальной точке измерений, и выдаётся сообщение: «Нет касания щупа».

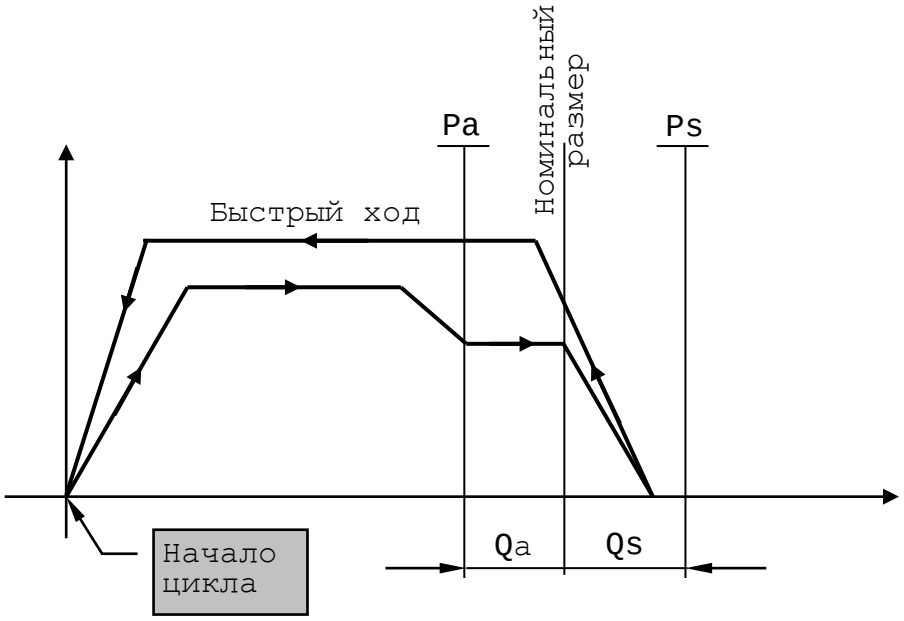


Рисунок 2.134

2.13.10 Управление стойкостью инструмента - TOF

Команда TOF позволяет объявлять инструмент негодным.
Формат:

(TOF, n) ,

где:

n - номер инструмента, который объявляется негодным; цифровая константа или параметр **E** (E10-E19).

Пример
(TOF, 22)
(TOF, E11)

Команда TOF не может быть активизирована при не присоединенных осях и запомненном поиске.

2.14 Управление расточными и обточными головками

Диаметральная или U ось позволяет выполнить следующие операции:

- растачивание конических и цилиндрических отверстий;
- круговые соединения (вогнутые или выпуклые);
- фаски;
- канавки;
- обточка;
- нарезание резьбы.

Программирование оси аналогично программированию других линейных осей. Координаты выражены в диаметрах; единица измерения - миллиметр (G71) или дюйм (G70). Движение оси U является одновременным и скоординированным с осью X,Y или Z, запрограммированной в том же кадре. Перемещение может быть выполнено на быстром ходу с функцией G00, или со скоростью обработки с функцией G01, запрограммированной в миллиметрах в минуту при помощи функции F.

Прежде чем выполнить профиль, двигая ось U, необходимо определить плоскость интерполяции, используя оператор: **(DPI,Z,U)**. Выполненные с осью U профили могут быть запрограммированы как с использованием обычной системы программирования, так и системы GTL; к ним может быть применена компенсация радиуса инструмента (радиус острия) на профиле (G41,G42) для учета радиуса острия инструмента и определения припуска (код UOV).

Значение припуска и исходных точек (временных и по приращениям) программируется на радиусе. Порядок осей при определении плоскости не может быть инвертирован. Изменением позиции осей Z и U (DPI,U,Z) определяется другая плоскость.

1) Пример программирования операции отделочной обработки приведён на рисунке 2.135:

```
N116 (DIS,".....................")
N117 F60,S630 T9.9 M6
N118 (DPI,Z,U)          Определение плоскости интерполяции
N119 (UAO,1)            Вызов абсолютной исходной точки головки
N120 (UOT,0,Z-200)      Временная начальная точка Z (плоскость детали)
N121 X Y160 T10 M3      Позиционирование  к отверстию 1
N122 G41 Z2 U51
N123 G1 Z-1 U44.98      Обработка фаски
N124 Z-44               Обработка отверстия диаметром 45 0/-0.04
N125 G G40 U40
N126 Z2 F40 S380
N127 G41 Y U106         Позиционирование к отверстию 2
N128 G1 Z-1 U99.975     Выполнение фаски
N129 Z-15               Выполнение отверстия диаметром 100 0/-0.05
N130 r5                 Выполнение соединения  R=5
N131 U60                Обработка плоскости
N132 r-3                Обработка соединения  R=3
N133 Z-40 U40           Обработка конуса
N134 G40 Z-44           Продолжение хода оси Z
N135 G U35
N136 Z100 M5
N137 (DPI,X,Y)
N138 (UAO,0)
```

Направление движения дуг окружности (запрограммированных с функциями G02 или G03, или с адресом r) и функция приведения в действие компенсации радиуса инструмента на профиле (G41 и G42) могут быть получены, если смотреть на профиль, который должен быть выполнен в плоскости Z-U. Пример приведён на рисунке 2.136.

Учитывая, что отрицательные диаметры обычно не программируются, достаточно иметь в виду первые два квадранта вышеуказанной плоскости.

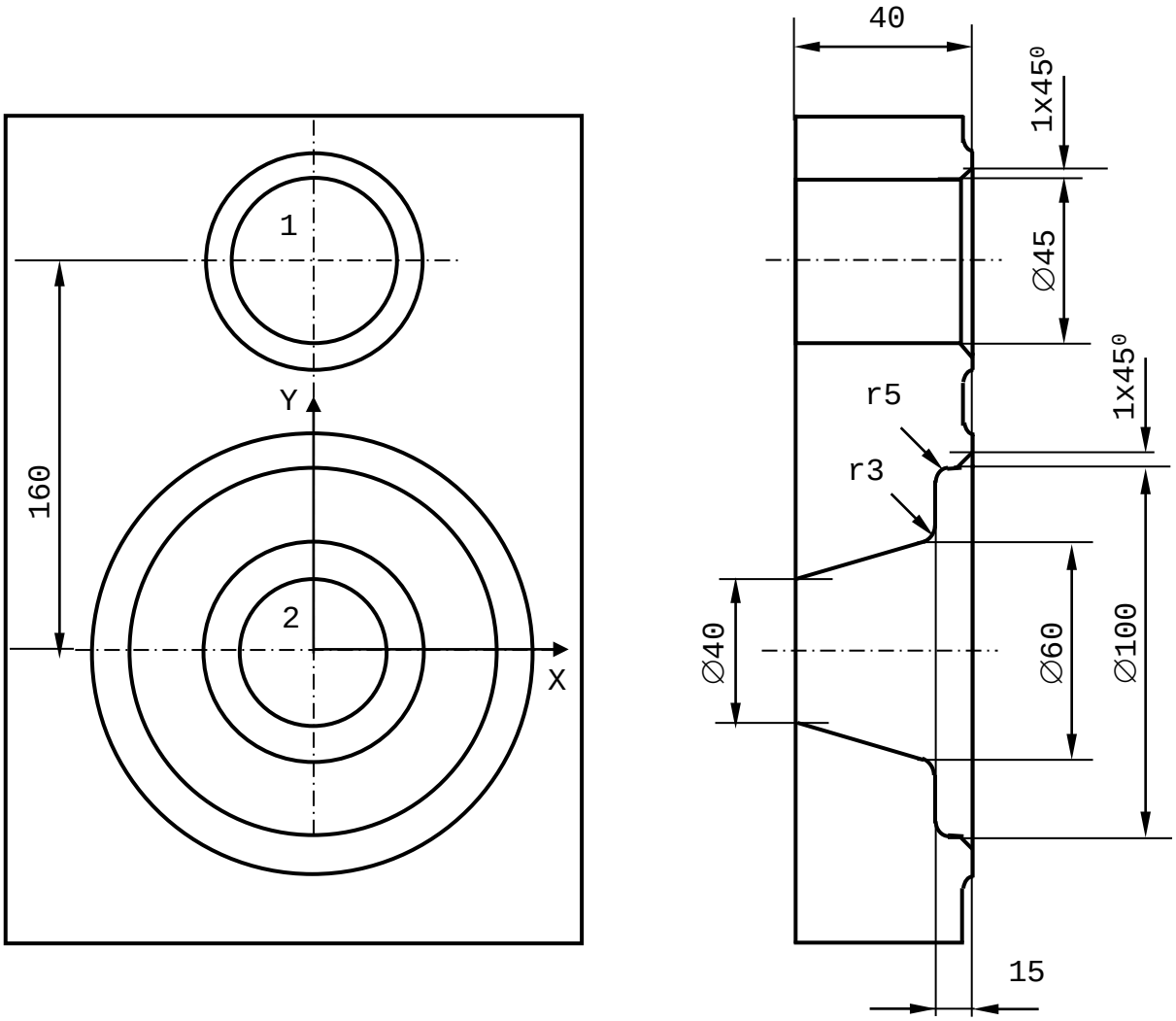


Рисунок 2.135

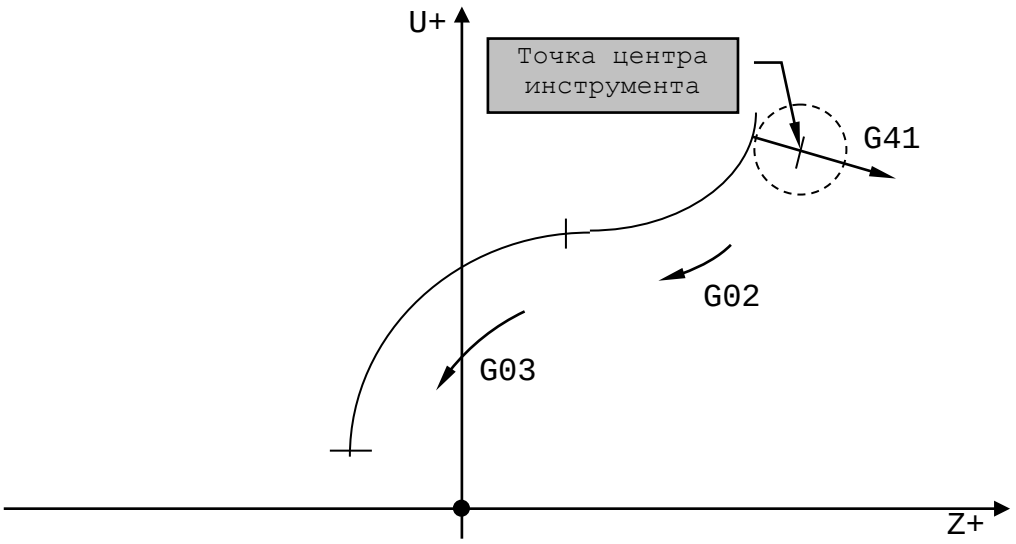


Рисунок 2.136

2) Пример обработки с диаметральной осью U профиля приведён на рисунке 2.137:

```
N1 (DIS,".................")
N2 (DPI,Z,U)
N3 p1 = Z U186
N4 l1 = p1, a-135
N5 l2 = Z U180, a180
N6 l3 = Z-50 U, a-90
N7 l4 = Z-90 U100, a-160
N8 l5 = Z U100, a180
N9 p2 = Z-100 U100
N10 S..F..T20.20 M6 M3
N11 (UAO,15)
N12 (UIO,X..,Y..,Z..)
N13 G X Y 8
N14 U190
N15 Z2
N16 G21 G41 p1
N17 l1
N18 l2
N19 b5
N20 l3
N21 r-15
N22 l4
N23 l5
N24 G20 G40 p2
N25 G U90
N26 Z2
N27 (DPI,X,Y)
N28 (UAO,0)
N29 ...
```

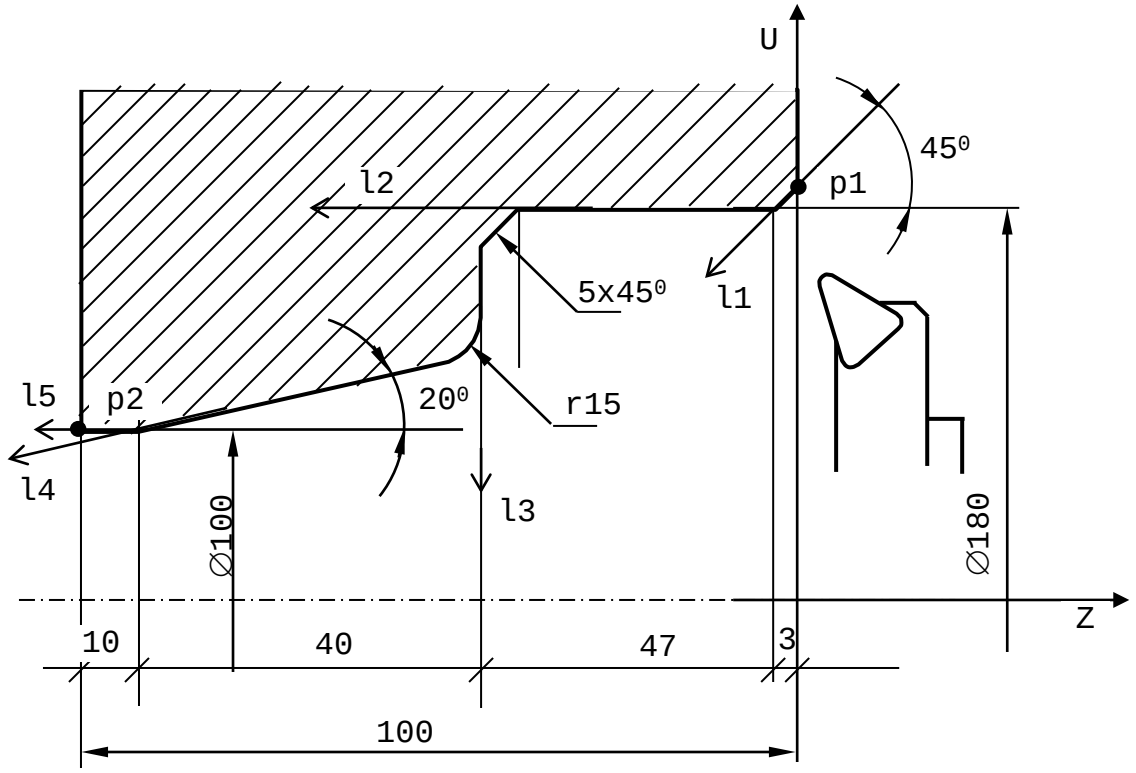


Рисунок 2.137

2.15 Управление электронным щупом

Щуп - это измерительная система, которая может быть установлена на шпинделе (рисунок 2.138). Он управляется, как любой инструмент, имеющий коррекции по длине и диаметру.

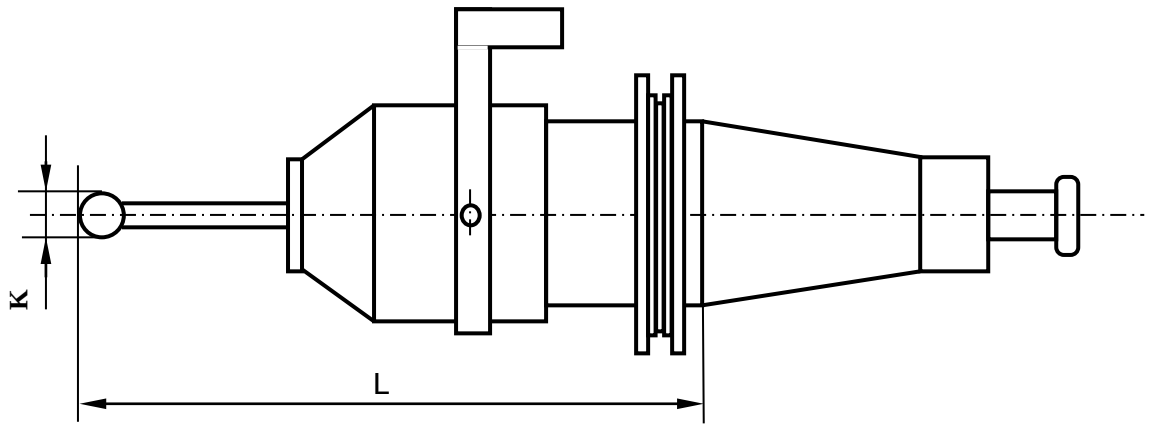


Рисунок 2.138

Посредством функций G72-G73 щуп выполняет соответственно:

- измерение координат точки в пространстве;
- измерение координат центра и радиуса окружности.

Используемый в зафиксированной позиции (компаратор с электронным датчиком), щуп посредством функции G74 измеряет отклонения от теоретических точек. Полученные значения запоминаются в параметрах **E**, определённых в цикле измерения щупом. Измерительный цикл для каждой измеряемой точки выполняется с последовательностью движений, приведённых на рисунке 2.134:

1) прибытие в точку приближения (**Pa**) со скоростью быстрого хода;
2) перемещение до точки измерения или предохранения (**Ps**) со скоростью измерения (**Vm**), прерывание и запоминание размера;
3) быстрый возврат в позицию начала цикла (центр отверстия при G73).

2.15.1 Операции предварительной установки измерительных циклов

Первый раз, когда используется щуп, или каждый раз, когда изменяются условия, необходимо:

- определить параметры измерения;
- выполнить динамическое измерение диаметра (**K**) сферы щупа;
- выполнить операцию «переквалификация щупа» относительно оси шпинделя;
- выполнить динамическое измерение длины (**Z**) щупа.

2.15.1.1 Определение параметров измерения - DPT

Для определения параметров измерения DPT необходимо задать параметры измерения с клавиатуры или из программы.
Формат:

DPT,Qa,Qs,Vm ,

где:
Qa - размер подхода, мм;
Qs - размер предохранения, мм;
Vm - скорость измерения, мм/мин.

Пример
DPT,10,12,1000 - с клавиатуры;
(DPT,10,12,1000) - из программы.

Для определения скорости измерения необходимо учитывать, что измерение, выполненное со скоростью Vm=1000 мм/мин, создает ошибку, равную 1 микрону.

2.15.1.2 Димамическое измерение диаметра шара щупа (внешний диаметр)

Диаметр (**K**) шара щупа должен быть вычислен динамически. Для этой цели может быть использовано пробное кольцо, центр которого соответствует начальной точке осей **X Y.**

2.15.1.3 Переквалификация щупа относительно оси шпинделя

Обычно центр шара щупа смещён относительно оси шпинделя. Поэтому необходима операция переквалификации щупа. Для выполнения этой операции используется пробное кольцо, в центре которого помещены начальные точки для осей X и Y.

2.15.1.4 Динамическое измерение длины щупа

Длина (**Z**) щупа должна быть измерена динамически. Для выполнения этой операции используется поверхность системы отсчёта пробного кольца, на которой помещается начальная точка для оси **Z.**
Ниже дан пример, который объединяет в одну программу выполнение всех операций предварительной установки при измерении щупом.

Предполагается, что имеется:

- пробное кольцо системы начала отсчёта;
- абсолютная исходная точка 99 для оси вращения **B** (если ось вмонтирована на индексированном поворотном столе);
- абсолютная исходная точка 99 для осей X, Y в центре отверстия пробного кольца;
- абсолютная исходная точка 99 для оси **Z** на верхней поверхности пробного кольца (поверхность начала отсчёта).

В корректоре, относящемся к щупу, накоплены следующие начальные значения:

Z = теоретическая длина щупа относительно оси сферы,
K = 0.

2.15.1.5 Пример полной переквалификации щупа

```
N1   (DIS,"DPT,RTA,RTO")
N2   T30.30 M6                - щуп в шпинделе
N3   (UAO,99)
N4   (DPT,10,12,600)          - определение параметров измерения
N5   RTA=0
N6   RTO=0
N7   E30=...                  - назначение диаметра отверстию пробного
                                кольца
N8   E31=E30/2
N9   E32=                     - назначение расстояния D от Z=0 до поверх-
                                ности измерения вдоль оси Z (обычно D=0).
N10  E33=E31+10
N11  GBO                      - (только если кольцо вмонтировано на пово-
                                ротном индексном столе)
N12  XY
N13  Z-4
N14  M...                     - активизация щупа
N15  G73   rE31 E40           - измерение координат центра окружности и
                                радиуса отверстия
N16  Z100
N17  (DIS,"RTA=",E40,"RTO=",E41)
N18  M
N19  RTA  =E40                - переквалификация абсциссы щупа
N20  RTO  =E41                - переквалификация ординаты щупа
N21  E34=(E30-E42*2)          - диаметр измерительного щупа
N22  (DIS  ,"ДИАМЕТР"=34)
N23  M
N24  (RQP ,30,30,KE34)        - запоминание диаметра сферы в корректоре K
N25  T30.30 M6                - приведение в действие нового корректора
N26  GXYE33
N27  G72  Z E32 E43           - измеряет размер Z на поверхности измерения
                                кольца
N28  E35=E34-E32              - разница между динамической и теоретической
                                величиной
N29  ZIOO
N30  (DIS"          Z=",E35)
N31  M
N32  (RQP,30,30,ZE35)         - модификация корректора длины (Z)
N33  M30
```

2.15.2 Операции, выполняемые со щупом

2.15.2.1 Измерительные циклы G72, G73

Посредством программирования измерительных циклов G72-G73 предусмотрено выполнение следующих операций:

1) модификация исходных точек:

- измерением поверхности системы отсчёта;
- посредством центрирования на отверстие.

2) проверка размеров:

- диаметров;
- плоскости и глубины отверстия.

2.15.2.1.1 Модификация исходных точек

Измерение поверхности системы отсчёта включает в себя два случая:

- модификация исходных точек при термическом дрейфе;
- модификация исходных точек посредством центрирования на отверстии.

1) Модификация исходных точек при термическом дрейфе осуществляется с применением куба корректировки, расположение которого определяется известными координатами.

Пример 1. Программирование цикла переквалификации исходной точки оси Y представлено на рисунке 2.139.

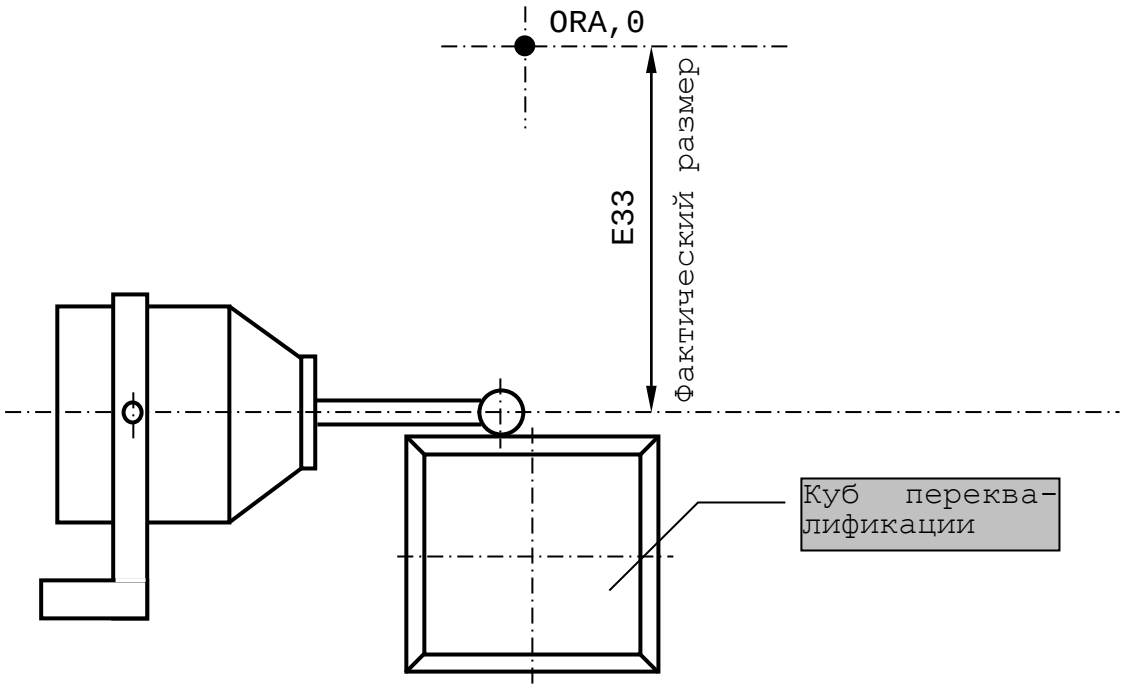


Рисунок 2.139

Главная программа

```
    .............
    .............
    N99 E33=-300
    .............
    /N100 (CLS,TAST3)
    .............
```

Подпрограмма TAST3

```
N500 (DIS,"RQU-DT")
N501  G72 YE33 E32      - измеренное расстояние, запомненное в параметре
                          E32
N502 E32=E32-E33
N503 (RQO,0,YE32)       - переквалификация исходной точки 0 для оси Y
N504 (RQO,1,YE32)       - модификация исходной точки 1 для оси Y
N505 (RQO,2,YE32)       - модификация исходной точки 2 для оси Y
```

Пример 2. Переквалификация исходной точки оси Y для обработки на новой паллете представлена на рисунке 2.140.

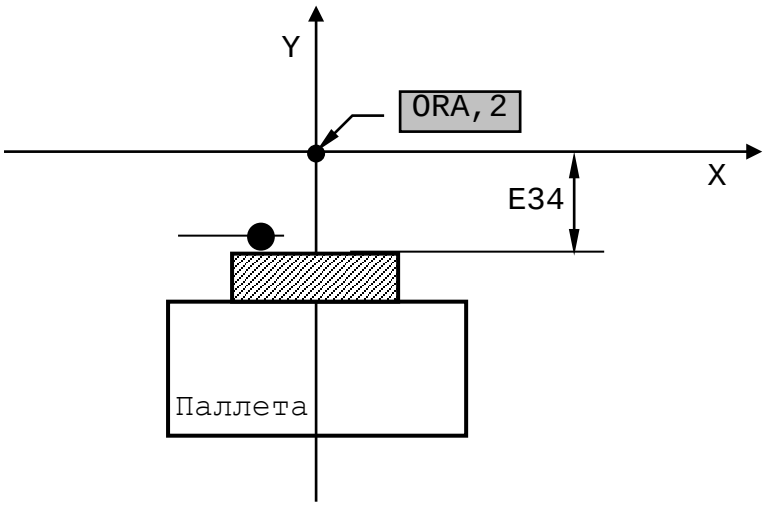


Рисунок 2.140

Главная программа

```
.....
N10 (UAO,1)
.....
.....
N194 M.....                    - замена паллет
N195 T30,30 M6                 - щуп в шпинделе
N196 (UAO,2)
N197 GXY
N198 E10=2
N199 E34=-250
/N200 (CLS,TAST4)
..............
```

Подпрограмма TAST4

```
N500  G72 YE34 E30
N501  E31=E30-E34
N502 (RQO,E10,YE31)          - модификация исходной точки 2 для оси Y
```

2) Модификация исходных точек посредством центрирования на отверстии.

До цикла корректировки следует запрограммировать позиционирование осей X и Y на оси отверстия и позиционирование щупа внутри этого отверстия.

Пример программирования:

```
 ..............
 N200 (DIS,"МОДИФИКАЦИЯ ИСХОДНЫХ ТОЧЕК ОСЕЙ X И Y)
 N201 T11.11 M6
 N202 GX180 Y60
 N203 Z-130
 N204 G73 r50 E35              - цикл измерения на отверстии диаметр 100.
                                 Координаты X и Y запоминаются в параметрах
                                 E35 и E36
 N205 E35=E35-180
 N206 E36=E36-60
 N207 (RQO,0. XE35, YE36)      - модификация исходной точки О осей X и Y в
                                 параметры E35 и E36
```

2.15.2.1.2 Проверка размеров:

Проверяют размеры:

- диаметров;
- плоскости и глубины отверстия.

1) При проверке размеров диаметров система позволяет проверить значение диаметра отверстия и, в зависимости от результата, полученного при сравнении обнаруженной величины с допущенными величинами, продолжить обработку или осуществить переход на выполнение кадра с данной меткой.

Пример 1. Программирование проверки диаметров отверстия. Номинальный диаметр равен 100 + 0.02/-0.015.

```
.............
"A1" N111
N112 GZ-150
N113 (DIS,"РАСТАЧИВАНИЕ Д = 100")
N114 F..S..T13.13 M6
N115 GX-120 Y80 M13
.............
N129 (DIS,"КОНТРОЛЬ  Д = 100")
N130 T14.14 M6
N131 GX-120 Y80              - позиционирование X Y на оси отверстия
N132 Z-85                    - позиционирование Z
N133 G73 r50 E30             - радиус, запомненный в E32
N134 E32=E32*2
N135 (DIS,E32)
N136 (BGT,E32,100.02,A3)
N137 (BLT,E32,99.985,A4)
N138 GZ150
N139 (DIS,"ДЕТАЛЬ В ДОПУСКЕ")
.............
.............
N2100 M30
"A3" N2101 (TOF,13)          - отверстие слишком большое
N2102 M00
"A4" N2103 (TOF,13)          - отверстие слишком маленькое
/N2104 (BNC,A1)
```

Отклонение, определённое между фактическим и теоретическим диаметром, сравнивается с допустимым. В зависимости от полученного результата сравнения возможны три случая:

- диаметр отверстия в пределах допустимого допуска; продолжается выполнение программы обычным образом;
- диаметр отверстия больше допустимого допуска; «Статус инструмента» (T13) автоматически обнуляется (метка A3), и программа цикл (M00) останавливается; деталь является негодной;
- отверстие меньше допустимого допуска; «Статус инструмента» (T13) автоматически обнуляется, и программа переходит к метке (A1), где осуществляется повторение цикла растачивания отверстия с альтернативным инструментом, предусмотренным таблицей продолжительности срока службы инструмента. Если этот инструмент не существует или его срок службы истек, цикл останавливается и выводится на экран сообщение, указывающее на то, что нет альтернативного инструмента.

Пример 2. Программирование цикла проверки диаметра круговой канавки, измеренного для трёх точек, приведено на рисунке 2.141.

```
.............
N154 (DIS,".................")
N155 T16.16 M6
N156 GX496.5 Y300
N157 Z-4
N158 G72 X500 Y300 E31        - запоминание X и Y – точка 1 в E31 и E32
N159 Z10
N160 X300 Y496.5
N161 Z-4
```

```
N162 G72 X300 Y500 E33          - запоминание X и Y – точка 2 в E33 и E34
N163 Z10
N164 X103.5 Y300
N165 Z-4
N166 G72 X100 Y300 E35          - запоминание X и Y - точка 3 в E35 и E36
N167 Z250
N168 p1=XE31 YE32
N169 p2=XE33 YE34
N170 p3=XE35 YE36
N171 c1=p1,p2,p3
N172 E31=FEC(1,3)*2-400         - разница между измеренным и теоретическим
                                  диаметром
N173 (BGT,E31,0.06,B1)
N174 (BLT,E31,0,B2)
..............

B1 - отверстие слишком большое
B2 - отверстие слишком маленькое
```

С этим циклом представляется возможным проверить внешние диаметры или диаметры секторов круга. Для достижения наибольшей точности вычисления три точки должны быть расположены, как можно дальше друг от друга.

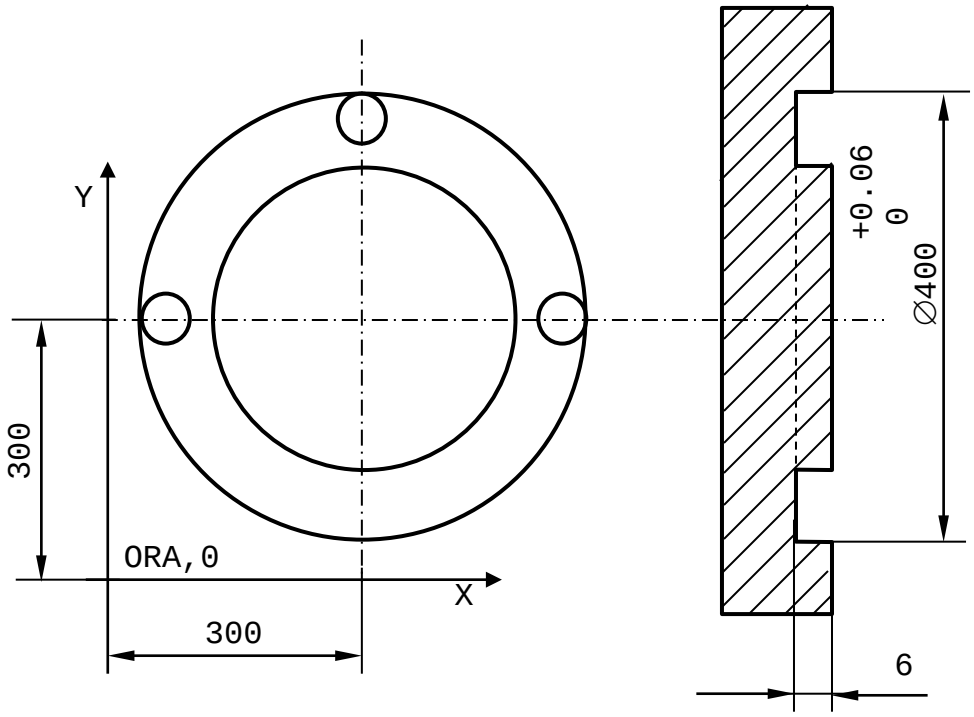


Рисунок 2.141

2) Проверка координат плоскости и глубины отверстий позволяет контролировать координаты плоскостей и глубин отверстий и, в зависимости от полученного результата при сравнении полученных и допустимых значений, продолжить обработку или осуществить переход на выполнение кадра с данной меткой.

Пример

Программирование цикла проверки координаты плоскости (требуемая координата по отношению к нулю оси Z:80+-0.1) приведено на рисунке 2.142.

```
..............
"C1" N252
N253 (DIS,".....")
N254 F..S..T23,T23 M6
..............
N268 (DIS,"КОНТРОЛЬ РАЗМЕРА 80")
N269 GX150 Y35                  - позиционирование X-Y в точке контроля
N270 G72 Z-80 E30               - измерение расстояния и запоминание в E30
N271 (BGT,E30,80.1,C2)
N272 (BLT,E30,79.9,C3)
```

```
N273 (DIS,"ДЕТАЛЬ В ДОПУСКЕ")
..............
..............
N2100 M30
"C2" N2101 (DIS,E30)          - размер слишком длинный
N2102 M00
"C3" N2103 (DIS,E30)          - размер слишком короткий
N2104 E31=E30-80
N2105 (RQU,23,23,ZE31,K)
(BNC,C1)
```

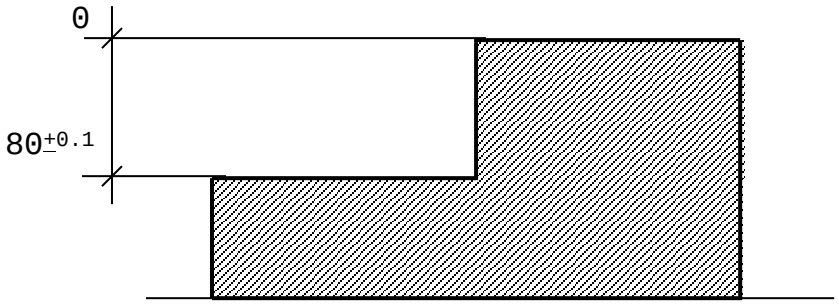


Рисунок 2.142

Отклонение, определённое между фактическим и запрограммированным размером, сравнивается с разрешённым допуском. В зависимости от полученного результата возможны три следующих случая:

- продолжение обработки;
- обработка прекращается, т.к. размер слишком большой (негодная деталь);
- размер не годится, переквалификация инструмента и повторение от «C1».

2.15.2.2 Операции, выполняемые при зафиксированном щупе

Щупом в зафиксированной позиции (например, компаратор с электронным датчиком) и инструментом, вмонтированным в шпинделе, при помощи измерительного цикла, определенного функцией G74, можно выполнить две операции:

- переквалификацию инструмента (автоматическое изменение коррекции инструмента);
- проверка целостности инструмента.

2.15.2.2.1 Автоматическое изменение коррекции инструмента

Автоматическое изменение коррекции длины инструмента осуществляется программированием трёхбуквенного кода RQU. Величина переквалификации равна величине, запрограммированной в параметрах **E**, которые используются в измерительном цикле.

Формат:

(RQU, номер инструмента, номер коррекций, ZEn, KEn) .

Параметры подробно описаны в п.2.13.8.

Если таблица коррекций была создана для запоминания текущих и максимальных значений корректировки, файл корректоров имеет вид:

номер коррекции,Z..,K..,c..,m..,c..,m.. ,

где:

c..c.. - фактические значения коррекций для Z и K;
m..m.. - максимальные значения, допустимые для Z и K.

Команда RQU корректирует фактические значения коррекций (c), объявляя инструмент непригодным, если фактическое значение превышает максимальное допустимое значение (m).

Примеры

```
1)    Модификация коррекции:
...............
 N170 GX100 Y100                 - позиционирование в точке измерения по-
                                   зиция щупа)
 N180 G74 Z-50 E30               - измерение смещения и запоминание значе-
                                   ния в параметре E30
 N190 (RQU,10,1,ZE30,K)          - модификация корректора 1 по длине (Z) в
                                   E30

2)    Модификация коррекции длины диаметра:
 ...............
 N200 GX100 Y100                 - позиционирование в точке измерения (по-
                                   зиция щупа)
 N210 G74 Z-50 E30               - измерение отклонения (ось Z) и запоми-
                                   нание значения в E30
 N220 G X150
 N230 Z-60
 N240 G74 X130 E31               - измерение отклонения (ось X) и запоми-
                                   нание значения в E31
 N250 E31=31*2
 N260 (RQU,10,1,ZE30,KE31)       - модификация корректора 1 по длине (Z)
                                   (E30) и по диаметру (K) (E31)
 ...............
```

2.15.2.2.2 Проверка целостности инструмента

Когда присутствует механизм управления сроком службы инструмента, инструмент может быть автоматически объявлен непригодным, т.к. истек срок службы или сумма корректировок превышает максимально допущенное значение.

Внутри программы можно объявить инструмент непригодным, т.к. отклонение, полученное при измерительном цикле (G74), превышает установленное значение. Для этого используются условные переходы и трёхбуквенный код TOF:

(TOF, номер инструмента).

Пример

Проверка целостности инструмента из программы:

```
 ............
 N490 T10.10 M6
 N500 GX100 Y100
 N510 G74 Z-50 E35
 N520 (BLE,E35,0.2,A1)
 N530 (TOF,10)                   - объявляет инструмент непригодным (10),
                                   если износ больше, чем на 0.2. В обратном
                                   случае, программа продолжится с метки «A»
                                   без выполнения кадров N530-N540.
 N540 MOD
 "A1" N550 (DIS,"ИНСТРУМЕНТ ОК")
 .........
```

2.15.2.3 Сообщения об ошибках при измерении щупом

Если измерение щупом не осуществилось, щуп возвращается в отправную точку, и на дисплее воспроизводится сообщение: «**Измерение щупом не произошло**».

Если не произошло освобождение стержня, воспроизводится сигнал аномалии.

2.16 Синхронизация между вычислением и движением осей

Символы **#** и **&** используют в кадрах, выполнение которых зависит от момента времени и соблюдения отдельных условий:

- указывает на запрос синхронизации;
& - указывает на отмену синхронизации.

Эти символы программируются после номера кадра, перед инструкцией. Если они не запрограммированы, принимается режим по умолчанию, который не предусматривает синхронизацию (синхронизация по умолчанию предусмотрена только для переменных **SA-SK**).

2.16.1 Запрос синхронизации

Программируя символ # в кадре, можно синхронизировать вычисление с движением осей. Синхронизация применяется при программировании команд в следующих случаях:

- когда обработка кадра зависит от результатов вычислений;
- когда в кадре значение переменной присваивается в конце запрограммированного движения.

Пример

```
N9  G X100Y80
N10  TIM1=TIM0          - принимает время часов системы при окончании движения осей, запрограммированного в кадре N9
.....
.....
N29  GXY
N30 # (UCG,2,X-50Y100,Y-20,Y-80)    - определяет графическое поле при окончании движения GXY
.....
.....
N59  GX50
N60 # (DLY,10)          - осуществляет остановку в 10 сек в конце движения оси GX50
.....
.....
N87  E30=0.2
N88 # (RQU,1,1,ZE30)    - переквалификация инструмента осуществляется, когда E30 достигает желаемого значения (0.2 в кадре N88, 0.3 в кадре N95)
N94 E30=0.3
N95 #  (RQU,1,1,ZE30)
```

2.16.2 Отказ от синхронизации

Программированием в кадре символа «**&**» отменяется синхронизация по умолчанию между вычислением и движением осей.

2.17 Виртуальные оси

Для обработки профилей на плоскости или на цилиндре при помощи оси вращения и линейной оси вводится понятие виртуальных осей.

Имеются следующие разновидности преобразований:

- способ 1: при исполнении профиля на плоскости, представляется возможным преобразование декартовых координат в полярные координаты; линейная ось перпендикулярна оси вращения;
- способ 2: при исполнении профиля на цилиндре, представляется возможным преобразование декартовых координат в цилиндрические координаты; линейная ось параллельна оси вращения.

Если одна из этих разновидностей активизирована, ось вращения позиционируется на нуле. Профиль можно программировать с помощью ISO или GTL языка в зависимости от оси (реальная или виртуальная), которая определяет декартовую плоскость.

2.17.1 Программирование первым способом

Программирование первым способом позволяет преобразовать декартовы координаты в полярные координаты.

Формат:

(UAV,1,реальная линейная ось, реальная ось вращения, виртуальная ось абсциссы, виртуальная ось ординаты),
т.е. (UAV,1,XC,UV,r) ,

где:

X - реальная линейная ось;
C - реальная ось вращения;
U - виртуальная ось по абсциссе;
V - виртуальная ось по ординате;
r - минимальный радиус; определяет область, куда запрещен ввод инструмента; при вычислении минимального радиуса необходимо указывать запрограммированную скорость так, чтобы скорость оси вращения не превышала скорости быстрого хода. Для вычисления минимального радиуса следует использовать формулу:

$$r = \frac{F}{Vcmax} * \frac{360}{2\pi}, \qquad (2.7)$$

где:

r - минимальный радиус;
F - скорость подачи, мм/мин;
Vcmax - скорость быстрого хода для оси вращения.

2.17.1.1 Пример программирования первым способом с применением GTL

Пример программирования первым способом с применением GTL приведён на рисунках 2.143, 2.144.

```
N1  T1.1 M6
N2  (DIS,"PROFILE MILLING")
N3  G X70 Z-5
N4  M21
N5  G0 C- X80 S2000 M3 M8
N6  (UAV,1,XC,UV,10)
N7  (DPI,U,V)
N8  p1=U20 V0
N9  l1=p1,a90
N10 c1=I0 J35 r-25
N11 l2=U-15 VO,a-90
N12 l3=UOV-20,a0
N13 c2=125 J-30 r-25
N14 G21 G42 p1 F300
N15 l1
N16 r3
N17 c1
N18 r4
N19 l2
N20 r5
N21 l3
N22 r5
N23 c2
N24  r3
N25  l1
N26  G40 G20 p1
N27  (UAV,0)
N28  (DPI, Z, X)
N29  GX80
N30  M20
----------------
----------------
N99  M30
```

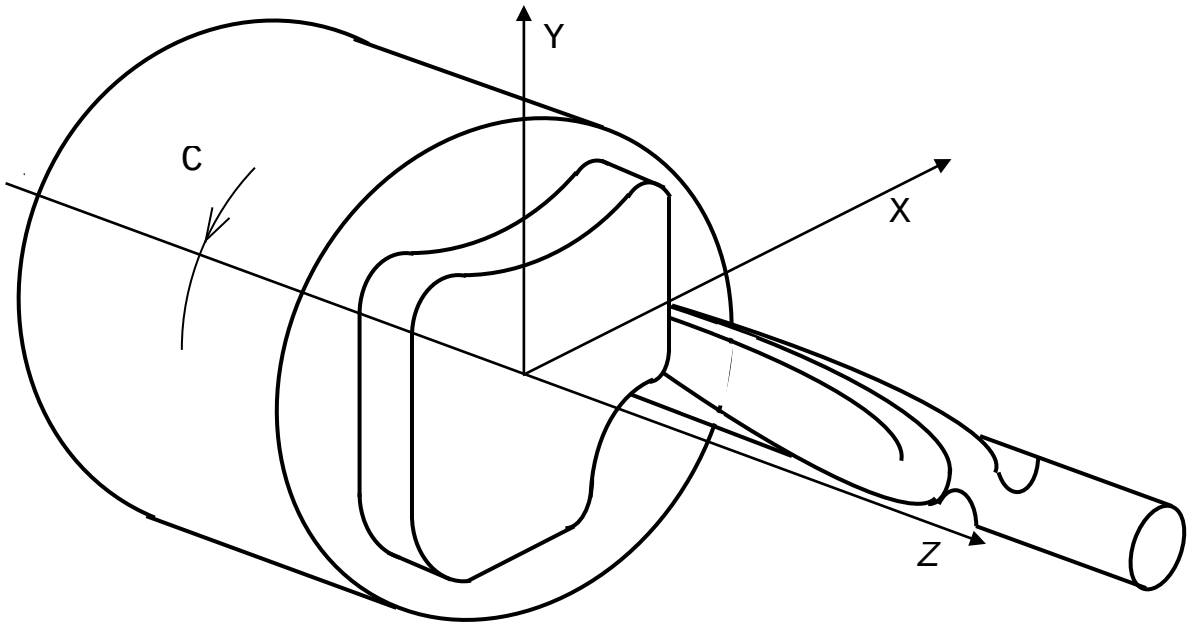


Рисунок 2.143

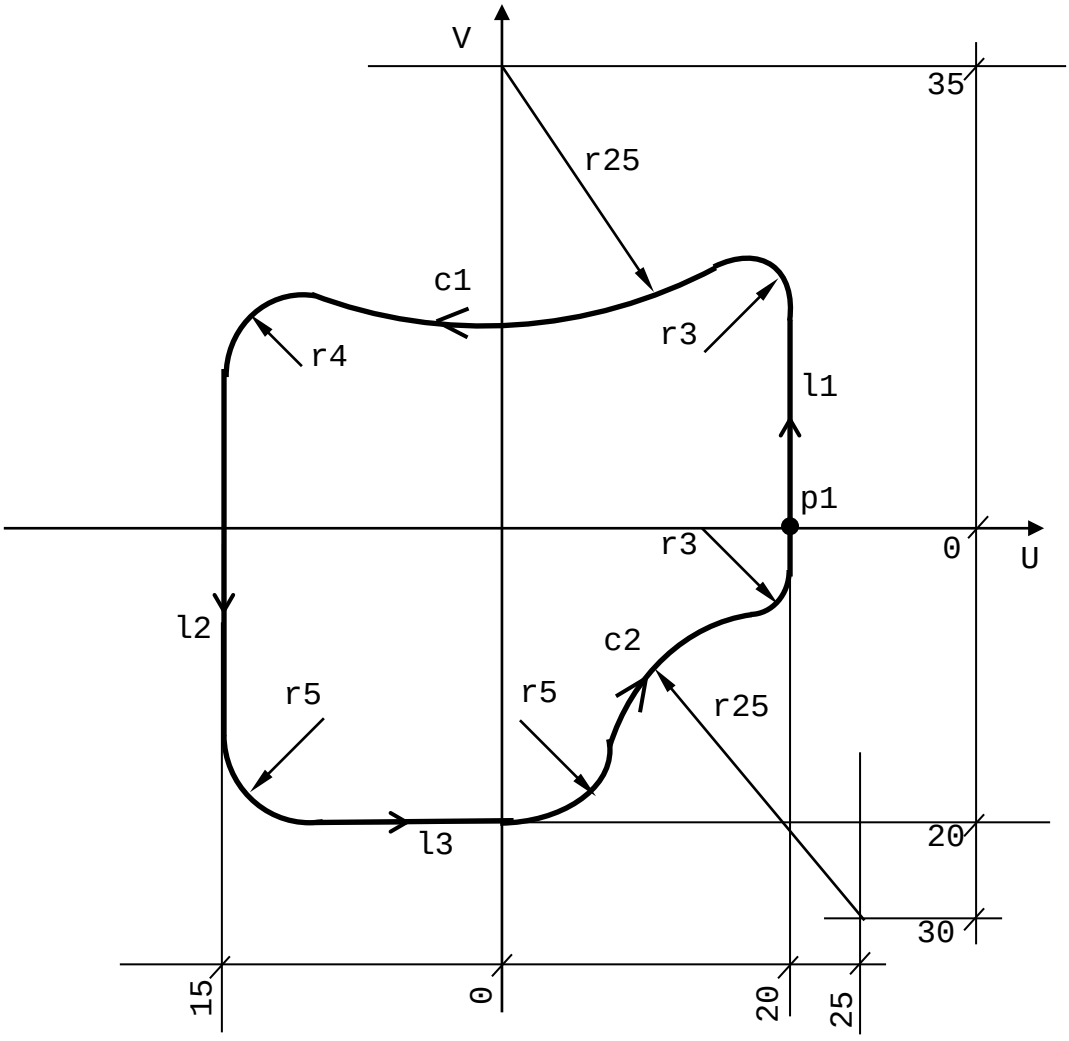


Рисунок 2.144

2.17.1.2 Пример программирования первым способом с применением ISO

Пример программирования первым способом с применением ISO приведён на рисунке 2.145.

Примечание - В примере минимальный радиус был вычислен при помощи формулы 2.7:

$$r = \frac{F \times 180}{3{,}14 \times Vcmax} .$$

```
; VIRTUALIZATION 1 ISO
T0.1 M6
GX30 Z-5
E40=110*180/(3.14159*800)
M21
GC0 X80 S1000 M3 M8
(UAV, 1, XC, UV, E40)
(DPI, U, V)
G1 G42 U20 V F110
V20
r3
U-15
b5
V-20
r5
U0
G40 G2 U20 V0 I20 J-20
(UAV, 0)
GX80
M20
M30
```

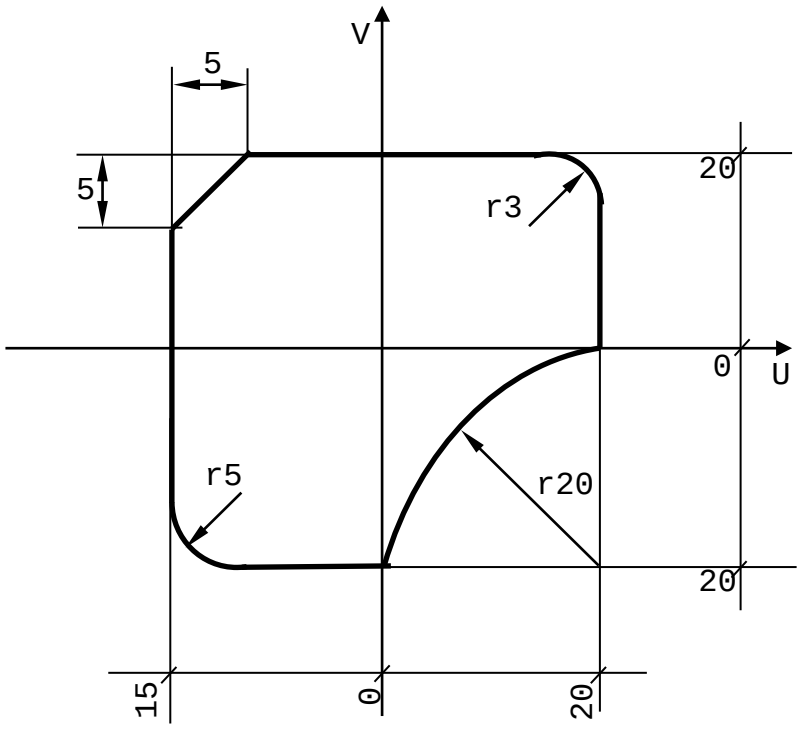


Рисунок 2.145

2.17.2 Программирование вторым способом

Этот способ позволяет преобразовать декартовы координаты в цилиндрические координаты. Профиль создаётся на декартовой плоскости, сформированной при помощи виртуальной оси вращения и реальной линейной оси.

Для программирования профиля необходимо использовать следующий формат:

(UAV,2,C,V,n) ,

где:

C - реальная ось вращения;
V - виртуальная ось;
N - радиус цилиндра, на котором профиль обрабатывается.

Пример программирования вторым способом приведён на рисунках 2.146, 2.147.

```
N1  ("DIS", "EXAMPLE")
N2  T1.1 M6
N3  GXY20 Z10
N5  G0 G94 B0 S2000 M3
N6  E30=60
N7  (UAV, 2, B, V, E30)
N8  (DPI, V, Y)
N9  p1=V0 Y20
N10 E31=2*3.1415*E30
N11 p2=VE31 Y20
N12 l1=p1, p2
N13 c1=I80 J45 r25
N14 c2=I140.71 J35 r-25
N15 c3=I180 J35 r-25
N16 l2=c1, c2
N17 l3=c2, c3
N18 c4=c3, l1, r15
N19 G21 G41 p1 F500
N19 Z-12
N20 l1
N21 c1
N22 l2
N23 c2
N24 l3
N25 c3
N26 c4
N27 l1
N28 G20 G40 p2
N29 (UAV, 0)
N30 G Z20
N99 M30
```

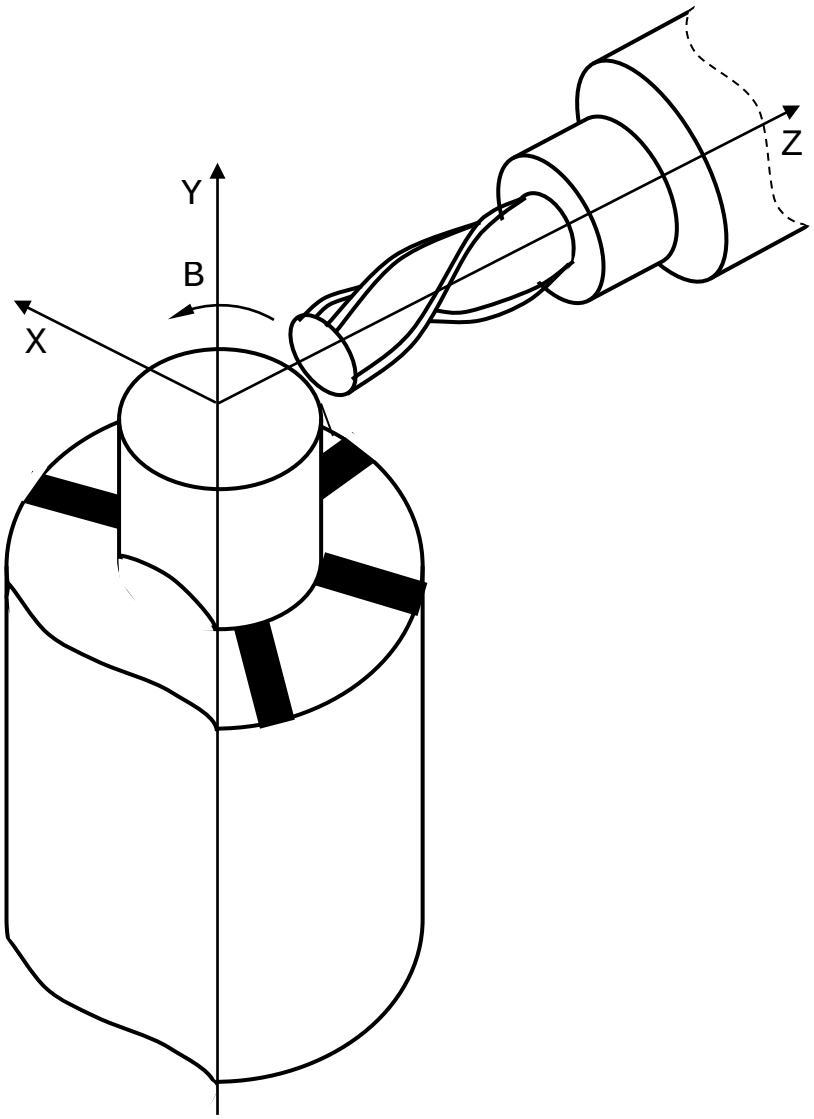


Рисунок 2.146

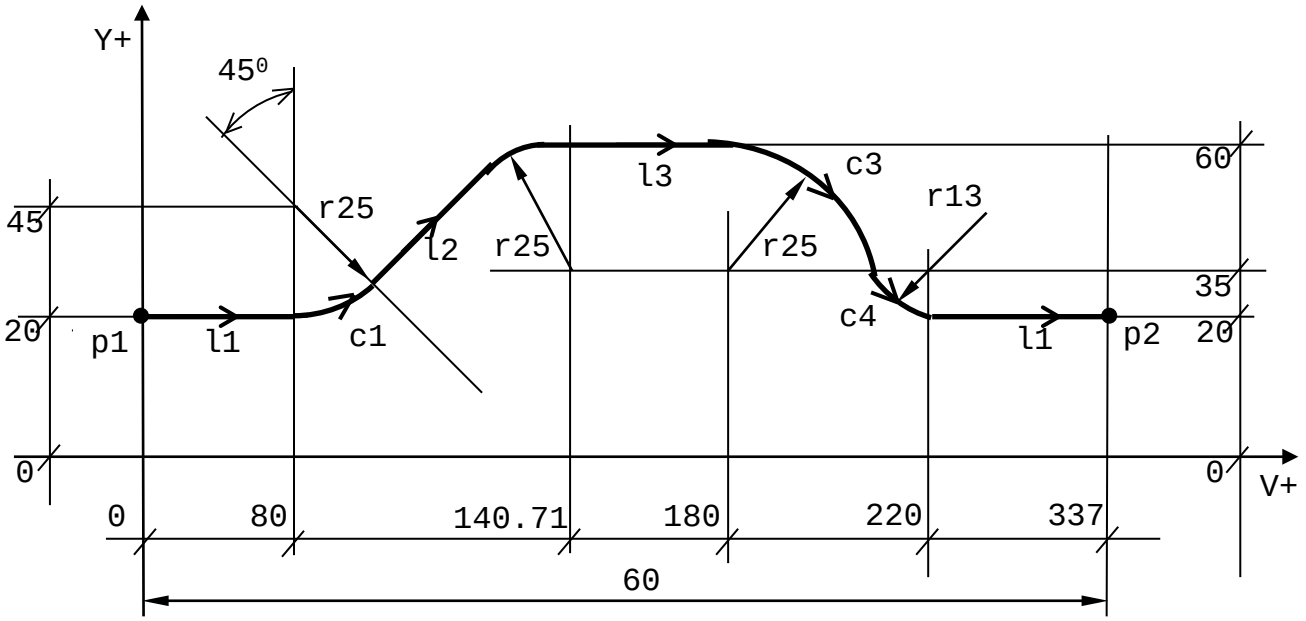


Рисунок 2.147

2.18 Параллельные оси

Трёхбуквенный код UAV позволяет, кроме определения виртуальных осей, устанавливать также параллельные оси при помощи определения ведущих и ведомых осей.

Чтобы двигать параллельные оси, необходимо программировать перемещение только ведущей оси, ведомая ось перемещается автоматически вместе с ведущей. При параллельных осях можно также программировать зеркальную обработку.

Параллельные оси устанавливаются при программировании UAV по режиму 3.

Формат:

(UAV, 3, имя ведомой, имя ведущей, соответствие ведущая-ведомая, зерк.) .

Пример
(UAV, 3, VWU, XYZ, 123, 212) ,

где:

3 - режим параллельных осей;
VWU - ведомые оси (от 1 до 3 символов);
XYZ - ведущие оси (от 1 до 4 символов);
123 - цифровой ряд, определяющий соответствие между осями; значение цифры определяет ведущую ось, а его позиция – ведомую; в примере **X** является ведущей для **V**, **Y** - для **W**, **Z** - для **U**;
212 - цифровой ряд, который характеризует тип обработки:

- 1 - нормальная обработка;
- 2 - зеркальная обработка.

В этом примере оси **V** и **U** перемещаются зеркально

Для разрешения применения параллельных осей необходимо, чтобы ведущая и ведомая оси были позиционированы в 0. Отменяется действие параллельных осей при помощи команды (UAV, 0).

2.19 Поворот системы координат в пространстве

Поворот всей системы координат применяется для произвольной ориентации плоскости интерполяции в пространстве. Для реализации данной функции в системе должны быть определены 3 виртуальные оси. Программирование обрабатываемого профиля выполняется с помощью виртуальных координат.

Использование данной функции возможно в версиях ПрО УЧПУ, обозначение которых содержит символ «С» в расширении, например, Z.15 РИВС, или в версиях ПрО, порядковый номер которых больше или равен 74. Например, Z.79.13P. Кодирование версий ПрО указано в «Руководстве по характеризации».

Формат:

(UAV,4,ось1ось2ось3,ось4ось5ось6,угол1,угол2,угол3),

где:

ось 1 -имя реальной оси 1;
ось 2 -имя реальной оси 2;
ось 3 -имя реальной оси 3;
ось 4 -имя виртуальной оси 4;
ось 5 -имя виртуальной оси 5;
ось 6 -имя виртуальной оси 6;
угол 1 -угол поворота относительно оси 1;
угол 2 -угол поворота относительно оси 2;
угол 3 -угол поворота относительно оси 3.

Направление угла вращения определяется знаком углов 1,2,3. Знак «-» соответствует повороту по часовой стрелке, знак «+» или его отсутствие соответствует повороту против часовой стрелки в соответствии с рисунком 2.148.

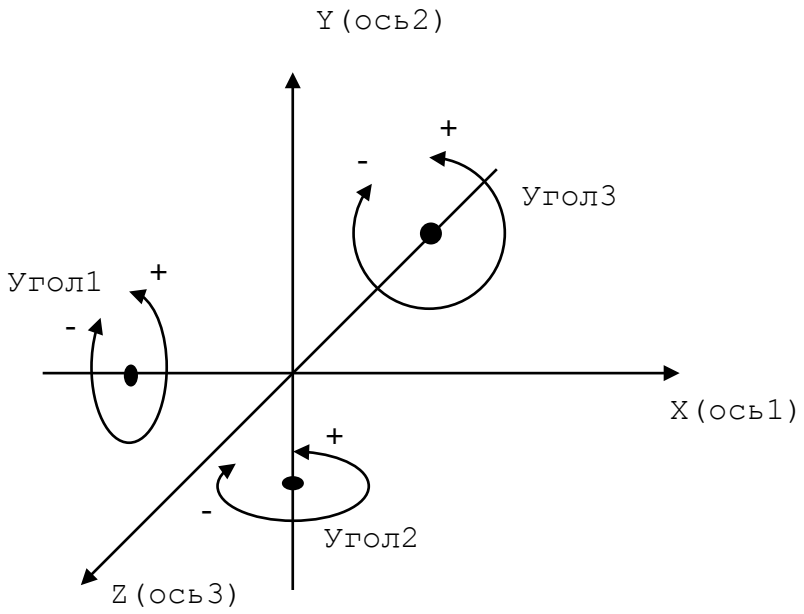


Рисунок 2.148

3 ТРЁХБУКВЕННЫЕ КОДЫ, ИСПОЛЬЗУЕМЫЕ ПРИ ПРОГРАММИРОВАНИИ УЧПУ

Трёхбуквенные коды, используемые при программировании УЧПУ, в зависимости от их функций могут быть разделены на пять групп:

1) коды, используемые при управлении файлами, приведены в таблице 3.1;
2) коды периферийных устройств приведены в таблице 3.2;
3) коды, используемые при управлении управляющими программами, приведены в таблице 3.3;
4) коды, используемые при управлении инструментом, приведены в таблице 3.4;
5) коды, используемые в кадрах управляющей программы, приведены в таблице 3.5.

Таблица 3.1 - Коды управления файлами

Код	Формат	Функция
EDI	EDI,имя/MPx	Вызов редактора для того, чтобы изменить существующую программу или записать новую программу с клавиатуры
DEL	DEL,имя/MPx	Удаляет программу из памяти
COP	COP,имя/MPx,/устройство	Копирует указанную программу из памяти на устройство
COP	COP,/устройство,имя/MPx	Копирует программу из устройства в память
REN	REN,имя/MPx, имя1/MPx	Изменяет имя программы
DIR	DIR,/MPx	Показывает список программ в памяти
FOR	FOR,имя/MPx,кол-во строк	Создает файл фиксированной длины и формирует поля файлов корректоров, продолжительности срока службы инструмента, начальных точек.
ATT	ATT,имя,100 ATT,имя,0	Защищает программу от записи. Убирает защиту.
DIF	DIF,имя/MPx, имя/MPx	Проверяет разницу между программами в памяти

Таблица 3.2 – Коды периферийных устройств

Код	Тип внешних устройств
TY	Телетайп

Таблица 3.3 - Коды, используемые при управлении управляющими программами

Код	Формат	Функция
E	EN[.тип] = значение	Определяет числовые переменные с одним из следующих типов: BY = байт; IN = целое число; LI = длинное целое число; RE = действительное; LR = длинное действительное; N - номер параметра
o	oN = значения координат или переменных	Определяет геометрический элемент как точку начала отсчета; N - номер элемента.
p	PN = значения координат или переменных	Определяет геометрический элемент как точку; N - номер элемента.
l	LN = значения координат или переменных	Определяет геометрический элемент как прямую; N - номер элемента.
c	CN = значения координат или переменных	Определяет геометрический элемент как окружность; N - номер элемента
TMR	TMR = значение	Определяет время, затрачиваемое на движение при G04 или в фиксированных циклах (выражается в секундах)
UOV	UOV=1 UOV=0	Определяет допуск припуска. Отмена припуска.
JOG	JOG=значение	Определяет величину перемещения, выполняемого в режиме ручных фиксированных перемещений.
RTA	RTA=значение	Определяет изменение величины щупа для оси X (аттестация щупа)
RTO	RTO=значение	Определяет изменение величины щупа для оси Y (аттестация щупа)
ERF	ERF=значение	Определяет допустимую ошибку формы
MCD	MCD=значение	Определяет максимальное отклонение направляющих косинусов в движении

Продолжение таблицы 3.3

Код	Формат	Функция
USB	USB=1 USB=0	Выполнение кадров с символом"/" (пропуск). Пропуск кадров с символом "/".
UVR	UVR=1 UVR=0	Выполнение программы в режиме быстрого хода. Отмена режима быстрого хода.
URL	URL=1 URL=0	Разрешение работы корректора рабочей подачи. Отмена работы корректора рабочей подачи.
USO	USO=1 USO=0	Подтверждение M01. Отмена M01.
UCV	UCV=N	Определяет тип вывода на экран осевых значений для видеокадра 1: UCV=0 рассчитанные величины осей; UCV=1 значения датчиков; UCV=2 ошибки позиционирования.
RAP	RAP=0 RAP=1	Автоматический возврат на профиль после перемещения вручную, последовавшего после «Стопа» с выбором оси. Автоматический возврат на профиль после перемещения вручную, последовавшего после "Стопа" по пути ручного перемещения.
UAS	UAS=1 UAS=0	Отключение осей (блокировка привода). Отмена вышеназванного режима.
RMS	RMS=значение	Определяет процент изменения скорости в режиме возврата при цикле резьбонарезания.
UEP	UEP=1 UEP=0	Включает использование позиционных ошибок. Отмена использования позиционных ошибок.
SA	SAN = значение	Определяет из программы значение сигнала пакета «A»; N - номер параметра.
SK	SKN = значение	Определяет из программы значение сигнала пакета «K»; N - номер параметра.
RMN	RMNpq	Переменная возвращает значение остатка пути для оси с порядковым номером «q» из процесса «p»
SYVAR	SYVARN = значение	Определяет значение переменных при записи файла из программы; N - номер параметра.
TIM	TIMN = значение	Определяет из программы системное время; TIM=0 сбрасывает часы; N - номер параметра.
TOT	TOTN = значение	Определяет из программы суммарное время; N - номер параметра.

Таблица 3.4 - Коды, используемые при управлении инструментом

Код	**Формат**	**Функция**
ORA	ORA,N,X...,Y...,Z ...	Определяет абсолютную начальную точку по осям. N - номер начальной точки. Для определения начальных точек в альтернативных единицах измерения номер должен быть взят с отрицательным знаком (-N)
CAO	CAO,N	Стирает начальную точку; N - номер начальной точки. Если N отсутствует, удаляются все записи файла начальных точек.
VOA	VOA,N	Воспроизводит начальную точку; N - номер начальной точки.
VTU	VTU,N[,T,COMPEN,T1,T2,T3,B]	Запоминает файл параметров для управления сроком службы инструмента: n: номер инструмента; T: альтернативный инструмент; COMPEN: корректировка альтернативного инструмента; T1: максимальное теоретическое время службы инструмента; T2: минимальное теоретическое время службы инструмента; T3: оставшееся время службы инструмента; B: состояние инструмента для индикации записи вводить:**VTU, n.**
CTU	CTU,N	Удаляет инструмент из файла срока службы инструментов. N: номер удаляемого инструмента; если операнд N не указан, команда удаляет все записи файла.
VOL	VOL=1 VOL=0	Активизация штурвала. Отключение штурвала.

Продолжение таблицы 3.4

Код	Формат	Функция
UCG	UCG,N,AXIS1I AXISIS, AXIS2I AXIS2S[AXIS3]	Определяет параметры инициализации для графического экрана: n=1, визуализация осей, не входящих в систему координат; n=2, визуализация осей, входящих в систему координат; ось 1I: нижний предел оси Z; ось 1S: верхний предел оси Z; ось 2I: нижний предел оси X; ось 2S: верхний предел оси X; ось S3: ось, перпендикулярная рабочей плоскости.
CLG	CLG	Очищает графический экран
DCG	DCG	Запрещает графический экран (всегда после CLG)
CAC	CAC,N	Удаляет корректор инструмента; n: номер корректора. Если n не определен, то команда удаляет весь файл.
SPG	SPG,имя	Выбирает программу.
REL	REL	Сбрасывает выбор программы.
DPT	DPT,Qa,Qs,Vm	Определяет параметры щупа: Qa: величина приближения (расстояние от условной точки щупа); Qs: величина безопасности (максимальное перемещение от точки касания щупа); Vm: скорость, мм/мин.
RCM	RCM	Разрешает запомненный поиск.
ERM	ERM	Запрещает запомненный поиск.
VIC	VIC,N	Визуализирует содержание таймерной переменной (TIMX); N: номер переменной. На дисплее визуализируется: **VIC, имя переменной, часы, минуты, секунды.**
SNC	SNC,n	Выполнение программы до кадра с номером n, например: **SNC,24**
DIS	DIS,переменная	Воспроизведение переменной
EVA	EVA, (выражение)	Вычисляет выражение и воспроизводит его на экране
UCA	UCA,n,Z,X	Модифицирует инкрементально корректора n на величину Z/X
MBR	MBR=1 MBR=0	Активизация обратного прослеживания профиля. Отмена обратного прослеживания профиля.

Таблица 3.5 - Коды, используемые в кадрах управляющей программы

Код	Формат	Функция
CLS	(CLS,имя подпрограммы)	Вызывает подпрограмму
BNC	(BNC,метка)	Выполняет безусловный переход к метке
BGT	(BGT,VAR1, VAR2,метка)	Переходит, если VAR1 > VAR2
BLT	(BLT,VAR1, VAR2,метка)	Переходит, если VAR1 < VAR2
BEQ	(BEG,VAR1, VAR2,метка)	Переходит, если VAR1 = VAR2
BNE	(BNE,VAR1, VAR2,метка)	Переходит, если VAR1 =/ VAR2
BGE	(BGE,VAR1, VAR2,метка)	Переходит, если VAR1 >= VAR2
BLE	(BLE,VAR1, VAR1,метка)	Переходит, если VAR1 <= VAR2
EPP	(EPP,метка1, метка2)	Выполняет часть программы между меткой 1 и меткой 2
RPT	(RPT,N)	Повторяет часть программы N раз (n < 99). Описание части программы начинается после блока, содержащего RPT, и заканчивается блоком, содержащим код ERP
ERP	(ERP)	Определяет границу части программы
UAO	(UAO,n)	Выбор абсолютной начальной точки; n: номер абсолютной начальной точки, ранее введен с клавиатуры
UOT	(UOT,n,X...,...,Z...)	Определяет временную начальную точку для заданных осей; n: номер абсолютной начальной точки.
UIO	(UIO,X..., Z...)	Объявляет начальную точку в приращениях относительно текущей абсолютной начальной точки
MIR	(MIR,X,Z) (MIR)	Определяет зеркальную обработку для объявленных осей. Отмена зеркальной обработки.

Продолжение таблицы 3.5

Код	Формат	Функция
URT	(URT,угол) (URT,0)	Поворачивает плоскость на угол относительно текущей начальной точки. Отмена поворота плоскости.
SCF	(SCF,n[,ось])	Масштабный коэффициент для объявленных осей; n: масштабный коэффициент. **Примечание** - Если оси не определены, масштабный коэффициент устанавливается для всех осей.
RQO	(RQO,n,ось..)	Переквалификация начальной точки для осей, определенных в программе; n: номер начальной точки
RQU	(RQU,NUT,NCOR,Z..,K..)	Переквалификация инструмента: NUT: номер инструмента; NCOR: номер корректора. Изменяет текущие корректоры и файл корректоров.
RQP	(RQP,NUT,NCOR,Z..,K..)	Изменяет корректоры Z и/или X, определенные в объявлении, файл корректоров не изменяется
DPI	(DPI,ось S1, ось S2)	Определяет плоскость интерполяции; **ось1,ось2**: оси, имена которых определяют плоскость
DTL	(DTL,ось1, ось2)	Определяет при позиционировании величину допуска для программированных осей (отличную от величин, объявленных в файле характеризации)
DLO	(DLO,ось + ось -)	Определяет программные ограничения программируемых осей (максимальный и минимальный предел)
DIS	(DIS, переменная)	Воспроизводит на экране переменную
TOF	(TOF,n)	Объявляет инструмент «вне использования»; n: номер инструмента.
UCG	(UCG,N,ось1 ось 1S,ось 2 ось 2S,[ось])	Определяет параметры графического экрана: n:1 воспроизведение с отключенными осями; n:2 воспроизведение с подключенными осями.
CLG	(CLG)	Очищает область графического экрана дисплея
DCG	(DCG)	Запрещает графический экран (должен быть запрограммирован после CLG)
DSA	(DSA,n,Z-Z+, X-X+)	Определяет пределы защищенной области: n : номер области; Z-: нижний предел оси Z; Z+: верхний предел оси Z; X-: нижний предел оси X; X+: верхний предел оси Y.
ASC	(ASC,n)	Разрешает защищенную область; n: номер области.
DSC	(DSC,n)	Запрещает защищенную область; n: номер области.
DPT	(DPT,Qa,Qs, Vm)	Определяет параметры щупа: Qa: величина подхода; Qs: величина безопасности; Vm: скорость измерения.
DLY	(DLY,n)	Определяет выдержку на указанный промежуток времени; n: выдержка времени в секундах (мах 32 с).
UAV	(UAV,1,XC,UV, r)	Определяет виртуальные оси U и V; r - минимальный радиус
	(UAV,2,C,V,r)	Определяет виртуальную ось V; r - радиус цилиндра
	(UAV,0)	Запрещает виртуальные оси

4 ПРОГРАММИРОВАНИЕ В ПРОЦЕССАХ

4.1 Параллельная синхронная работа с несколькими процессами

Информация данного раздела относится к УЧПУ, версии ПрО которых имеют в обозначении расширения буквы «…РМ», а также к УЧПУ, версии ПрО которых имеют порядковый номер 60 и выше; например, Z.60.P. Кодирование версий ПрО указано в «Руководстве по характеризации».

Для параллельного управления несколькими процессами (до пяти) вводятся следующие трёхбуквенные коды: **EXE, REL, WAI, SND**.

4.1.1 Код EXE

Код **EXE** загружает и запускает выполнение указанной программы под управлением ранее выбранного процесса.

Формат:

(EXE,n,ИМЯ ПРОГРАММЫ/MPx) ,

где:

N - номер процесса: цифровая константа или параметр типа **BY**;

/MPx - имя запоминающего устройства (x=0÷3), если оно отличается от имени запоминающего устройства, заявленного по умолчанию в файле **PGCFIL;**

ИМЯ ПРОГРАММЫ - наименование программы.

Пример

```
(EXE,2,Progl)
```

или:

```
E4=2
(EXE,E4,Progl/MP2)
```

4.1.2 Код REL

Код **REL** выгружает выполняемую программу, загруженную ранее с кодом **SPG** или **EXE**. Действие кода **REL** из управляющей программы аналогично действию трёхбуквенного кода **REL**, выполненного с клавиатуры.

Формат:

(REL)

4.1.3 Код WAI

Код **WAI** может иметь две функции.

1) Прекращает выполнение программы в процессе и заставляет данный процесс ждать команды повторного старта, подаваемой другим процессом.

Формат:

(WAI,n) ,

где:

n - номер процесса, дающего команду повторного старта; цифровая константа или параметр типа **BY.**

Пример

```
(WAI,3)
```

или

```
E1=3
(WAI,E1)
```

2) Устанавливает процесс, который выполняет команду ожидания до тех пор, пока специфическая переменная не примет требуемого значения.

Формат:

(WAI,Var=ЗНАЧЕНИЕ) ,

где:

Var - определяет наименование переменной системы типа **BL, BY** или **IN;**

ЗНАЧЕНИЕ - определяет ожидаемые значения (цифра или **E**-параметр).

Пример

(WAI,SA10.BY=2) ,

или

(WAI,SK2.IN=E11) ,

или

(WAI,SYVAR2=E5) .

Примечания

1. Если переменные типа **BY** или **IN**, то блок **WAI** может включать следующие операторы:
   > больше чем;
   < меньше чем;
   # отличный от.

   Пример

   (WAI,SYVAR>5)
   (WAI,SK2>E8)
   (WAI,SA10.BY#2)
2. Переменные **SYVAR**, **SA**, и **SK** являются общими для всех конфигурируемых процессов.
3. Код **WAI** может быть запрограммирован в любом процессе.

4.1.4 Код SND

Код **SND** даёт команду повторного старта процессу, находящемуся в состоянии ожидания.

Формат:

(SND,n) ,

где:

n - номер процесса, которому посылается команда повторного старта; цифровая константа или параметр типа **BY.**

Коды **SND** могут быть запрограммированы в любом процессе, при этом система проверяет, находится ли процесс в состоянии ожидания **WAI**, а также, соответствуют ли друг другу данный процесс и процесс, дающий команду повторного старта.

Пример

(SND,4)

или

E5=4
(SND,E5)

Для того чтобы синхронизировать **SND** с движением осей, блок должен включать символ **#.**

Пример

N25#(SND,2).

Каждый процесс для вывода на экран во время работы имеет доступ к видеокадру, выбранному функциональной клавишей «**P1**» или «**P3**», при этом на экран выводится информация, относящаяся к данному состоянию процесса.

Видеокадр **#0** используется для того, чтобы вывести на экран состояния всех процессов.

Доступ к видеокадру **#6** имеет только тот процесс, из которого он первично задан.

4.2 Режимы синхронизации между процессами

4.2.1 Условное ожидание процесса

Условное ожидание процесса - это когда один из процессов ожидает выполнение части программы другим процессом.

1) Команда повторного старта даётся кодом **SND**.

Пример

ПРОЦЕСС 1	ПРОЦЕСС 2
.....	.....
N20.....	N105
N21(SND,2)	N106(WAI,1)
M22.....	N107.....
.....	.....

Процесс 2 закончил кадр N106, когда процесс 1 выполняет команду (SND,2) в кадре N21.

2) Команда повторного старта для процесса ожидания подаётся сигналом **вх/вых**.

Пример

ПРОЦЕСС 1	ПРОЦЕСС 2
.....	.....
N130.....	N50.....
N131(WAI,SA12=1	N51.....
N132.....	.....

Процесс 1 заканчивает кадр N131, когда бит 12 в пакете SA равен «1».

3) Команда повторного старта даётся сигналом логики.

Пример

ПРОЦЕСС 1	ПРОЦЕСС 2
.....	.....
N146.....	N200.....
N147(WAI,SK250=8)	N201.....
N148.....	.....

Процесс 1 заканчивает кадр N147, когда байт 250 пакета «**К**» равен 8.

4) Команда повторного старта даётся, когда принимается значение ожидаемой переменной **SYVAR**.

Пример

ПРОЦЕСС 1	ПРОЦЕСС 2
.....	.....
N150.....	N120.....
N151 E8=10.3+20.7	N121(WAI,SYVAR2=31)
N152 SYVAR2=E8	N122.....
N153.....	N123.....
.....	.....

Процесс 2 заканчивает кадр N121, когда переменная SYVAR2 принимает значение 31 в кадре N152 процесса 1 (E8 = 31).

4.2.2 Взаимное ожидание процесса

Взаимное ожидание процесса иллюстрируется следующим примером:

ПРОЦЕСС 1	ПРОЦЕСС 2	ПРОЦЕСС 3
.....	.....	.....
N107.....	N230.....	N330.....
N108(WAI,2)	N231(SND,1)	N331(SND,1)
N109(WAI,3)	N232(WAI,1)	N332(WAI,1)
N110(SND,2)	N233.....	N333.....
N111(SND,3)	.....	.....
N112.....	.....	.....

Три процесса ждут друг друга взаимно до выполнения кадров N112 (процесс 1), N233 (процесс 2), N333 (процесс 3) одновременно.

4.3 Схема синхронизации для трёх параллельных процессов

Схема синхронизации для трёх параллельных процессов иллюстрируется следующим примером:

ПРОЦЕСС 1(PROG91)	ПРОЦЕСС 2(PROG92)	ПРОЦЕСС 3(PROG93)
"START"		
N1(EXE,2,PROG92)	N1T20.20M6S...F.	N1(WAI,1)
N2(EXE,3,PROG93)	N2GX....Y....Z..	N2T30.30M6S...F.
N3T1.1M6S...F...	N3G2X..Y..I..J..	.....
N4G1...X...Y....	.....	.....
.....	N15GXY	N19GXY.....
N24(SND,2)	N16(WAI,1).....	N20(WAI,SYVAR,1)
.....	.....	.....
.....	.....	.....
N35(SND,3)	.....	N80(SND,1)
.....	GXY	N81.....
.....	N40(WAI,SA10=1)	.....
.....	.....	.....
.....	.....	.....
N59E5=10	.....	.....
N60SYVAR1=E5	N85(SND,1).....	.....
.....	N86.....	.....
N66(WAI,2)	.....	.....
N67(WAI,3)	.....	.....
N68.....	.....	.....
.....	.....	.....
N76GXY	.....	.....
N77(WAI,2)	N98GXYZ.....	N93 XYZ.....
N78(WAI,3)	N99(SND,1)	N94(SND,1)
N100(BNC,START)	M100(REL)	N95 (REL)

В данном примере программа PROG91, рассматриваемая в качестве главной программы, активизируется командой SPG в процессе 1. Программы PROG92 и PROG93 запускаются в кадрах N1 и N2 программы PROG91.

Процесс 2 выполняет кадры до N15, а затем ожидает процесс 1, чтобы дать ему команду повторного старта, т.е. выполнить кадр N24.

Процесс 3 не начинает выполнять программу, пока процесс 1 не выполнит кадр N35. Команды WAI также могут быть входом/выходом, логическим сигналом или переменными SYVAR.

В данном примере процесс 2 ожидает (на кадре N40), чтобы бит 10 в пакете SA стал равным 1 для повторного старта.

Процесс 3 ожидает на кадре N20, чтобы процесс 1 на кадре N60 дал ему повторный старт, т.к. переменная SYVAR приняла значение 10.

Процесс 1 ожидает на кадрах N66-N67 команду рестарта с других процессов.

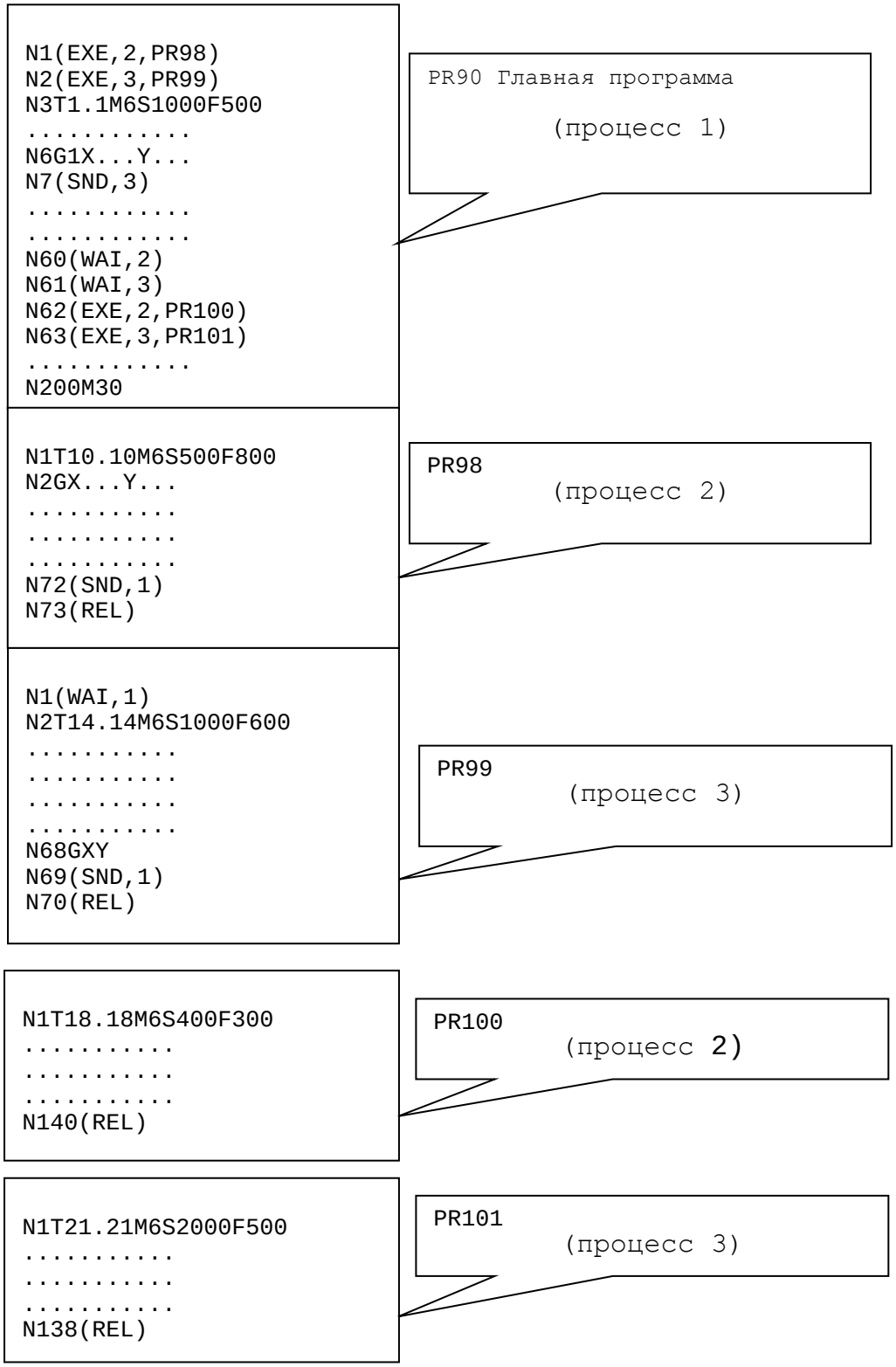
На кадрах N68-N86-N81 три процесса синхронизируются на повторном старте.

Главный процесс 1 возвращается к началу программы (BNC,START), когда два параллельных процесса 2 и 3 закончены.

4.4 Примеры программирования

4.4.1 Программирование трёх синхронизированных процессов

Программирование трёх синхронизированных процессов, один из которых рассматривается как главный, показано на следующем примере:



Пояснения к примеру:

1 Для активизации главной программы (например, PR90) из процесса (например, процесса 1) нажмите «**Р3**», а затем введите:

SPG,PR90.

2 Связь главной программы в процессе 1 с любой другой программой может быть автоматически осуществлена с помощью инструкции **EXE** из программы **PR90.**

3 Для выполнения программы используйте следующую процедуру:

- выберите видеокадр «**СИСТЕМА**»;
- установите все процессы в режим «**АВТОМАТИЧЕСКИЙ**» («**AUTO**»);
- нажмите клавишу «**ПУСК**».

4 Введённый в память поиск не может быть осуществлён на программах, содержащих инструкции синхронизации (**EXE**, **WAI**, **SND**).

4.4.2 Программирование трёх независимых процессов

Программирование трёх независимых (не синхронизированных) процессов приведено в таблице 4.1.

Таблица 4.1 – Пример программирования трёх независимых процессов

Программа	Номер программы (процесса)	Комбинирование процесс-программы с клавиатуры
N1T1.1M6S3000 N2GXY N3G1G41X100Y50F500 N4G2X...Y...I...J... .... N99G40X10Y N100GXYM30	ПРОГРАММА 10 (ПРОЦЕСС 1)	«Р3» выбирает процесс 1 SPG,PROG10
N1T1.1M6S400 N2G81R2Z-50F500 N3X-20Y10 .... N40G80XYZM30	ПРОГРАММА 11 (ПРОЦЕСС 2)	«Р3» выбирает процесс 2 SPG,PROG11
N1T1.1M6S800 N2GXY N34GXYZ	ПРОГРАММА 12 (ПРОЦЕСС 3)	«Р3» выбирает процесс 3 SPG,PROG12

Примечания

1 «**ПУСК**» может быть использован двумя способами:

- видеокадр «**СИСТЕМА**»: если все процессы находятся в режиме «**AUTO**», то выбранные программы делают автоматический старт;
- видеокадр «**ПРОЦЕССn**»: каждая программа делает независимый старт в выбранном режиме.

2 Введённый в память поиск выполняется для каждого процесса независимо.

5 ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ASSET

5.1 Назначение языка ASSET

Утилиты **ASSET** позволяют из управляющей программы обращаться к устройствам памяти **MP0, MP1, MP2, MP3, MP4, MP5, MP6,** которые могут быть расположены на FLASH DISK, HDD, FDD и других периферийных устройствах, подключённых к УЧПУ, а также создавать пользовательские видеокадры с возможностью выводить в них данные.

5.2 Хранение данных

Для хранения данных в УЧПУ используются форматированные файлы в устройствах памяти **MP0, MP1, MP2, MP3, MP4, MP5, MP6.**

Каждый файл состоит из записей фиксированной длины. Каждая запись состоит из определенного количества полей. Длина записи может быть установлена во время конфигурации.

Управление файлами включает следующие операции:

- произвольный доступ к чтению;
- произвольный доступ к записи/редактированию.

Для форматирования файла необходимо определить структуру записи. В языке **ASSET** допустимы следующие форматы:

- **CH** (символ);
- **IN** (целое);
- **RE** (действительное);
- **LR** (действительное с двойной точностью).

Соответствие между стандартными форматами УЧПУ и форматами языка **ASSET** представлено в таблице 5.1.

Таблица 5.1 - Соответствие между стандартными форматами УЧПУ и языка **ASSET**

Тип переменной	Стандартный формат	Формат ASSET
Символ	Axxx xxx - кол-во элементов	(.xCH)
Целое	I U	(.IN) нет в ASSET
Действительное	Rxxx xxx : кол-во цифр	(.RE)
Действительное с двойной точностью	Lxxx xxx: кол-во цифр	(.LR)
Байт	(.BY)	нет в ASSET
Булевское	(.BL)	нет в ASSET
Целое с двойной точн.	(LI)	нет в ASSET

Для форматирования файла необходимо осуществить следующие процедуры:

1) DEL, FORMAT /MP3;
2) EDI, FORMAT /MP3;
3) записать последовательность знаков, используя данные таблицы 5.1.

5.3 Управление файлами

5.3.1 Команды управления файлами

Для управления форматированными файлами можно использовать следующие команды:

- **OPN** - открыть файл;

- **DER** - определить запись;
- **RED** - читать запись;
- **WRT** - записать;
- **CLO** - закрыть файл;
- **CRE** - создать файл;
- **CAN** - удалить файл.

Примечание - Параметры, которые будут в дальнейшем указаны для каждой инструкции, обязательны за исключением тех, которые заключены в квадратные скобки.

5.3.2 Форматы команд управления файлами

5.3.2.1 OPN - открытие файла

Команда **OPN** открывает файл, даёт ему имя и соединяет с каналом связи.
Формат:

(OPN, номер канала, имя файла [/устройство] [,тип доступа]),

где:

номер канала - определяет номер сконфигурированного канала, в котором открывается файл; может быть константой типа (**.BY**) или **E-параметром**;

имя файла [/устройство] - имя файла (максимум 6 символов); если устройство не определено, то по умолчанию принимается **MP1**;

тип доступа - может быть:

- **R** - только чтение;
- **W** - запись/редактирование;
- по умолчанию - **R**.

Пример
(OPN, 1, FILQ/MP3, W) - Открыть файл **FILQ** на канале 1 для чтения и записи.

Используемые (допустимые) в **ASSET** каналы должны быть описаны при характеризации процесса (инструкция **CHN** файл **PGCFIL**).
При объявлении **OPN**, всегда определяется наличие свободного допустимого канала. Если такого нет, занятый канал закрывается для упорядочивания доступа к нему.
Сообщения об ошибках появляется, если:

- обозначенный канал уже занят для другого файла;
- обозначен несуществующий файл;
- устройство периферии не подключено;
- обозначенный файл защищён от записи/редактирования;

В мультипроцессной версии ПрО нельзя открыть один и тот же файл для записи/редактирования одновременно по запросу из двух различных процессов. Для разных процессов открывается файл только для чтения.

5.3.2.2 DER - определение записи

Команда **DER** позволяет определить переменные для записи файла. Перед выполнением операций с файлом сначала необходимо описать структуру, а затем установить связь между переменными в структуре и физическими данными.
Формат:

(DER, номер структуры, имя переменной [,имя переменной]...) ,

где:

номер структуры - номер записи структуры (от 1 до n). Может быть **E-параметром** или переменной типа (**.BY**). Параметр определяется при характеризации в инструкции **STR** файла **PGCFIL**. При вводе номера структуры, уже описанной,

новая структура заменит старую;

имя переменной - имена переменных, связанных с записью структуры.

Каждая запись структуры может быть связана с 17 переменными (**E, SYVAR, SA** и другие допустимые типы переменных) со следующим форматом **IN, RE, LR, CH**. Формат цифровых переменных определяет также и длину поля записи, в то время как в символьных переменных длина поля определяется числом, предшествующим формату переменной.

Пример
SYVAR. 5CH - определяет установку 5 символов из **SYVAR**.

Синтаксис, используемый для спецификации символьных переменных, позволяет также поддерживать строку записи данных.

Пример
Допустим, что файл FILE/MP3 состоит из следующих форматов:

```
IN - для поля 1
CH -     "    2 (длиной 10 символов)
LR -     "    3
CH -     "    4 (длиной 3 символа)
LR -     "    5
```

Программирование в данном случае будет иметь вид:

(OPN, 3, FILE/MP3,W); открывается файл для записи на канале 3

(DER, 2, E10, SYVAR.10CH, E40,SYVAR10.3CH,E41); определение для структуры двух следующих переменных:

- поле 1 - E10
- поле 2 - SYVAR
- поле 3 - E40
- поле 4 - SYVAR10
- поле 5 - E41

(RED, 1, 2, 10); чтение записи 10 из файла, открытого на 1-ом канале. Когда операция выполнена, содержание полей присваивается переменной, описанной в записи структуры 2.

5.3.2.3 RED - чтение записи

Команда **RED** позволяет читать файл, открытый командой **OPN.** Команда читает данные и назначает их переменным, описанным в записи структуры. Если структура записи не была до этого определена, будет сообщение об ошибке.

Система проверяет на совместимость формат записи, определенный при создании файла, и структуру записи, определённую командой чтения. Чтение выполняется в двоичном коде.

Формат:

(RED, номер канала, номер структуры записи [,запись]) ,

где:

номер канала - номер канала, содержащего файл; цифровая константа типа (**.BY**) или **параметр E;**

номер структуры - номер структуры для чтения; цифровая константа типа (**.BY**) или **параметр E;**

запись - номер требуемой записи (константа от 1 до 32767 или **параметр E** типа **IN**); если номер записи не определен, обращение будет к записи, следующей непосредственно после последнего **RED** или **WRT,** если файл был только что открыт, будет выполняться чтение первой записи.

Пример
Чтение файла исходных точек, структура которого IN, CH, LR, CH, LR.
Имя файла - FI1EOR/MP3.

Сначала следует определить структуру для копирования:

```
(DER, 2, E10, SYVAR.CH, E41, SYVAR2.CH, E42)
```

Последующее программирование будет иметь вид:

```
(OPN, 1, FILEOR/MP3,R); открытие файла только для чтения.
(RED, 1, 2, 1); чтение записи 1 (начальная точка 0) из файла, открытого на канале 1.
```

Содержание полей записи записывается в переменные, описанные в структуре 2. Ошибки будут, если:

- запись не существует;
- канал не был открыт;
- структура записи не была определена;
- формат файла не сравним с запрограммированной структурой записи.

5.3.2.4 WRT - запись в запись

Команда **WRT** осуществляет запись данных в запись файла, открытого в выбранном канале. Запись возможна, если файл был открыт для записи/редактирования. Команда присваивает данные переменным, описанным в структуре записи. Система проверяет на совместимость формат записи, определённый при создании файла, и структуру записи, определённую командой **WRT.**

Формат:

(WRT, номер канала, номер структуры записи [,запись]) ,

где:

номер канала - номер канала, содержащего файл; может быть константой типа (**.BY**) или **E-параметром**;

запись - номер записи (константа от 1 до 32767 или **E-параметр** типа **IN)**; если номер записи не определён, обращение будет к записи, следующей непосредственно после последнего **RED** или **WRT,** если файл был только что открыт, запись будет выполняться в первую запись;

номер структуры - номер структуры записи, связанной с командой **WRT**.

Пример

Отредактировать 3-е поле 2-го файла начальных точек.

```
      Имя файла - ORIG/MP3.
; - определение структуры записи
(DER, 2, E10, SYVAR10.CH, E45, SYVAR11.CH, E46, SYVAR12.CH, E47)
;
; - открытие файла для записи
      (OPN, 1, ORIG/MP3, W)
;
; чтение 3-ей записи (2-ая начальная точка) из файла ORIGIN (поле 3 установлено в E45)
(RED, 1, 2, 3)
;
E45 = E45+0.5
;
; перезапись после редактирования поля 3
(WRT, 1, 2, 3)
;
; закрытие канала
(CLO, 1)
```

Ошибки будут, если:

- запись не существует;
- канал не открыт;
- структура записи не определена;
- формат файла не сравним с запрограммированной структурой записи.

5.3.2.5 CLO - закрытие файла

Команда **CLO** позволяет закрывать файл, который был открыт ранее командой **OPN**. Эту команду необходимо использовать перед открытием другого файла на этом же канале.

Формат:

(CLO, [, номер канала]) ,

где:

номер канала - константа типа (.**BY**) или **E-параметр**; если номер канала не определён, закрываются все занятые каналы.

Ошибки будут, если специфицированный канал не занят.

5.3.2.6 CRE - создание файла

Команда **CRE** создаёт файл, структура записи которого уже определена с **DER**. Создание файла предполагает существование логического канала. После создания файла можно получить к нему доступ командой **OPN**.

Формат:

(CRE, N структуры, имя файла [/устройство], число записей) ,

где:

N структуры - номер структуры записи;

/устройство - имя периферийного устройства, на котором создается файл; если оно не указано, определяется по умолчанию;

число записей - длина файла по количеству записей; может быть константой типа **IN** от 1 до 32767 или **E-параметром**; область памяти, занимаемая файлом, определяется количеством и форматом записей.

Пример

```
(DER, 2, SYVAR.CH, E40, SYVAR 1. CH, E41)
```

(CRE, 2, PUNTXZ, 20); создаётся файл, названный PUNTXZ, содержащий до 20 записей, каждая из которых содержит следующее:

имя оси 1 размер 1 имя оси 2 размер 2

Размеры являются измеренными величинами.

```
(OPN, 1, PUNTXZ, W)
SYVAR.CH = "X"; поле 1 = "X"
SYVAR1.CH="Z";  поле 3 = "Z"
;специфицирует номинальные координаты точки на профиле.
(RPT,20);
; измерение реальных координат точки X и Z и установкой их в E40, E41.
G72X размер Z размер E40
; запись в запись, следующую непосредственно за последней записью. Если предыдущая запись отсутствует, то содержание полей, описанных в структуре 2, записывается в первую запись.
(WRT, 1, 2)
(ERP)
(CLO, 1)
```

Ошибки будут, если:

- нет допустимого канала;
- структура записи не была определена;
- файл уже существует;
- память переполнена.

5.3.2.7 CAN - удаление файла

Команда **CAN** позволяет удалить файл, уже созданный командой **CRE**. Область памяти, занятая файлом, освобождается. Команда **CAN** предполагает существование логического канала.

Формат:

(CAN,имя файла [/ устройство]) **,**

где:

имя файла - имя удаляемого файла;
/устройство - имя периферийного устройства, содержащего файл.

Пример
(CAN, PUNTXZ/MP3)

5.4 Управление ошибками ввода/вывода

При помощи определённых команд **ASSET** (**OPN, RED, WRT. CLO, CRE, CAN**) можно управлять ошибками ввода/вывода или автоматически, или из управляющей программы. Для этого необходимо установить параметр **ERR**.

При конфигурации система автоматически сбрасывает управление ошибками, т.е. устанавливает **ERR=0**.

Посредством установки **ERR=1** активизируется управление ошибками от программы. При обнаружении ошибки программирования системная переменная **IOSTA** приобретает значение, соответствующее коду ошибки, и визуализирует сообщение. Исполнение в данном случае не прерывается при обнаружении ошибки. Однако в этом случае необходимо проверить содержимое **IOSTA** с целью тестирования правильности программирования кадров. При устранении ошибки **IOSTA** устанавливается в «**0**».

При **ERR=0** в случае обнаружения ошибки система прерывает цикл отработки. Возможные коды ошибок ввода/вывода показаны в таблице 5.2.

Таблица 5.2 - Коды ошибок ввода/вывода в языке ASSET

Коды ошибок	Описание
1	канал уже открыт/канал еще закрыт
2	недопустимый номер канала
3	нет допустимого канала
4	файл не обнаружен
5	файл уже существует
6	незаконная операция
7	файл уже открыт
8	файл не открыт
9	защищенный файл
10	ошибки формата
11	запись не обнаружена
12	аппаратные ошибки
13	конец файла
14	ошибки логического вх/вых
15	недопустимое устройство
16	структура записи не определена
17	переполнение памяти
18	область ASSET не сконфигурирована

5.5 Доступ к клавиатуре

При помощи **ASSET** можно обращаться из управляющей программы к клавиатуре с целью:

- ввода данных для последующей их установки в переменные;
- ввода параметров для индикации на 8-ом видеокадре.

Для активизации клавиатуры используется команда **INP.**
Формат:

INP, [комментарий], имя переменной [,поле ввода] **,**

где:

комментарий - может быть:

- строкой в кавычках;
- номером поля, содержащего строку комментарий, предварительно определённую командой **DEF** (для индикации на экране); номер поля может быть константой типа **IN** или **E-**параметром;

имя переменной - имя переменной, куда будет записана введённая информация; формат ввода определяется форматом переменной; для ввода не адресуемого к экрану процесса формат переменной принимает начальное значение;

поле ввода - ограничивает ввод для индикации на 8-ом видеокадре определением номера поля экрана; может быть типа (**.BY**) или **E-**параметром.

Для обращения из управляющей программы к клавиатуре рекомендуется использовать следующие процедуры:

1) записать необходимую инструкцию **INP**. Цикл (или процесс в мультипроцессной системе) остановится и будет перезапущен после того, как процедура ввода завершится. Временная приостановка процесса будет указана кодом **INP** на видеокадре **#0** в графе «Сост. Проц.»/
   2-ая строка на экране воспроизведёт строку комментария и начальное значение. Если ввод выполняется для установки в переменную, то начальное значение содержит переменную для установки. Если ввод выполняется для индикации - начальное значение специфицирует текущее поле экрана. Для сброса индикации инструкции **INP** надо нажать клавишу «**СБРОС**»;
2) нажать клавишу «**ENTER**» для указания того, что ввод для доступа к клавиатуре закончен. После этого можно выполнять новые манипуляции по установке переменной, индикации на экране и перезапускать цикл.

Пример

```
.........
E40 = 1000
(INP, "FEED = ", E40)
```

- выполнение останавливается, и на второй строке экрана воспроизводится: FEED = 1000. Теперь можно изменять значение, установленное в E40. При нажатии клавиши «**ENTER**» визуализируемое значение запоминается в E40 и цикл продолжит выполнение.

5.6 Видеокадр 8 процесса (экран пользователя)

5.6.1 Структура видеокадра 8 процесса

В мультипроцессных системах для каждого процесса выделен видеокадр (экран пользователя), но можно пользоваться одним видеокадром одновременно с другим процессом.

Видеокадр 8 процесса (экран пользователя) имеет 19 строк по 78 знака в каждой. Внутри каждой строки можно задать определённое количество полей, каждое из которых может содержать в себе буквенные строки (комментарии) или цифровые строки (содержит переменные).

Для обращения пользователя к видеокадру из управляющей программы можно использовать следующие команды:

- **SCR** - разрешение/запрещение видеокадра 8;
- **DEF** - определение полей видеокадра 8;
- **OUT** - высвечивание полей видеокадра 2.

5.6.2 Форматы команд обращения к видеокадру 8 процесса

5.6.2.1 SCR - разрешение/запрещение видеокадра 8

Команда **SCR** разрешает/запрещает видеокадр 8. Если экран пользователя в монопроцессной системе уже был назначен для процесса, то команда **SCR** сбросит поля экрана, очистит его и подготовит к использованию другим процессом. Если в мультипроцессной системе экран пользователя был уже выделен процессу, то эта команда

генерирует сообщение об ошибке.

Формат:

(SCR,ON) или (SCR,OFF)

5.6.2.2 DEF - определение полей экрана пользователя

После **SCR** следует определить поля экрана, используя для этого команду **DEF**. Для изменения полей необходимо сначала сбросить экран, используя команду **(SCR,OFF)**. Команда **DEF** используется исключительно для определения полей, а не для высвечивания их.

Формат:

(DEF, N поля, строка, [столбец], начальное значение [,формат] [,R]),

где:

N поля - определяет номер поля; может быть константой типа **INTEGER** или **E-параметром**; верхний предел зависит от конфигурации процесса;

строка - идентификатор строки экрана, содержащей поле; может принимать значение от 1 до 20 или быть **E-параметром**;

столбец - номер начального столбца поля; может быть числовой константой в диапазоне от 1 до 64 или **параметр E**, если он не определен, то по умолчанию принимается 1;

начальное значение - определяет начальное значение и, следовательно, длину поля; это может быть номер, ряд алфавитно-цифровых символов или имя переменной; если не определено, то по умолчанию будет поле с пустым пространством; соотношение между форматом начального значения и длиной поля экрана приведено в таблице 5.3;

Таблица 5.3 - Соотношение формата начального значения и длины поля экрана

Формат	Длина (в символах)
Булевское (BL)	1
Байт (BY)	1
Целое (IN)	6
Целое двойной точности (LI)	9
Действительное (RE)	11
Действительное двойной точности (LR)	11
Символ	Изменяется в соответствии с длиной строки знаков или форматом инициализации переменной

формат - определяет формат поля; может быть число типа **(.BY)** со значением от 1 до 7 или **E-параметр**; назначается, когда начальное значение определено не однозначно или не специфицировано; формат должен быть конгруэнтен с начальным значением;

R(A) - разрешение инверсного изображения для определённого поля. Разрешение цветного изображения задаётся символом **«A»** и кодом от **0** до **255**, который определяет цвет фона и цвет текста, например: **A179.**

Номер цвета определяется порядком расположения цвета в файле **cnc.ini**, расположенном в каталоге **C:\CNC\MP0** (нумерация цвета начинается с нуля). В таблице 24 приведён рекомендуемый порядок расположения цвета и код цвета по палитре **RGB**. Код с буквой «**A**» формируется с использованием таблицы 5.4 следующим образом:

1. формируем код цвета фона из таблицы цвета:

код цвета фона = 16 * номер цвета;

2. формируем код цвета символа из таблицы цвета:

   код цвета символа = номер цвета;

3. формируем код с буквой «**A**»:

   код с «**A**»=код цвета фона+код цвета символа.

Таблица 5.4 - Порядок расположения цвета и код цвета по палитре **RGB**

Номер цвета	**R** красный	**G** зелёный	**B** синий	Цвет	Использование цвета в других видеокадрах
0	05	15	05	тёмно-зелёный	цвет фона в редакторе и др.
1	63	63	00	жёлтый	координаты в поле «ФАКТ»
2	16	00	16	тёмно-фиолетовый	«УПРАВЛЕНИЕ СТАНКОМ», надписи на клавишах «1»-«8»
3	00	00	35	тёмно-синий	заголовки «ФАКТ», «ПРОГРАМ»
4	63	42	42	розовый	цвет точек траектории инструмента в видеокадре #6
5	00	00	00	чёрный	сетка в видеокадре #6
6	30	36	30	тёмно-серый	Светотени
7	45	51	45	серый	Светотени
8	53	59	53	светло-серый	Светотени
9	88	88	88	резерв	
10	0	63	48	голубой	цвет в поле «ПРОГРАМ»
11	12	37	12	зелёный	курсор в редакторе
12	63	00	00	красный	сообщения об ошибках
13	42	21	00	коричневый	цвет координат в кадре #6
14	00	63	00	ярко-зелёный	Т, корректор, кнопка «ПУСК»
15	63	63	63	ярко-белый	фон в строке ввода

5.6.2.3 OUT - индикация полей

Команда **OUT** позволяет визуализировать поля, определённые пользователем с **DEF** (макс. 6 полей). Для одиночного поля команда **OUT** может также присваивать новые значения этому полю, предполагая, что они совместимы с форматом поля. С командой **OUT** поля визуализируются, начиная с начального столбца, определенного в **DEF**. Маска каждого поля зависит от ее формата. Соответствие между форматом поля и маской представлено в таблице 5.5.

Таблица 5.5 - Соответствие между форматом поля и маской

Формат	Длина (в символах)
Булевское (BL)	X
Байт (BY)	XXX
Целое (IN)	[-] XXXXX
Целое двойной точности (LI)	[-] XXXXXXXX
Действительное (RE)	[-] XXXXX.XXXX
Действительное двойной точности (LR)	[-] XXXXX.XXXX
Символ	X...X (изменяется в соответствии с длиной, определённой в DEF)

Формат:

(OUT [, N поля [, значение]]...) ,

где:

N поля - определяет номер индицируемого поля; может быть числовой константой типа **IN** или **параметром E**; если знак минус **«-»** предшествует номеру поля, индикация происходит в режиме инверсии, если номер поля не определён, то все поля, определённые в команде **DEF** и не индицируемые до сих пор, появятся на экране;

значение - определяет значение для индикации в специфицированном поле; может быть числовой константой, алфавитно-цифровой стро-

кой или именем переменной в формате, совместимом с форматом этого поля; если опущено, то на поле индицируется его старое значение.

Пример приведён на рисунке 5.1.

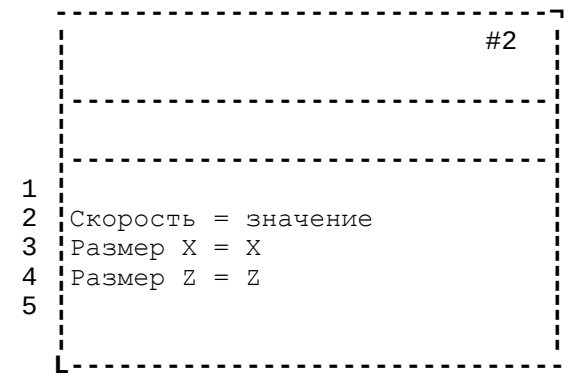


Рисунок 5.1

(SCR, ON); разрешение экрана пользователя для выбранного процесса

(DEF, 1, 3, 10, "Размер X = "); определение имени поля для X.

(DEF, 2, 3, 19, 0, 6); определяет формат и значение инициализации для X

(DEF, 3, 4, 10, "Размер Z = "); определение имени поля для Z;

(DEF, 4, 4, 19, 0, 6); определяет формат и значение инициализации для Z

(DEF, 5, 5, 10, "Скорость = "); определение имени поля для скорости подачи

(DEF, 6, 5, 19, 0, 5); определяет формат и значение инициализации для скорости подачи

(OUT); индикация специфицированных полей, изображена на рисунке 5.2.

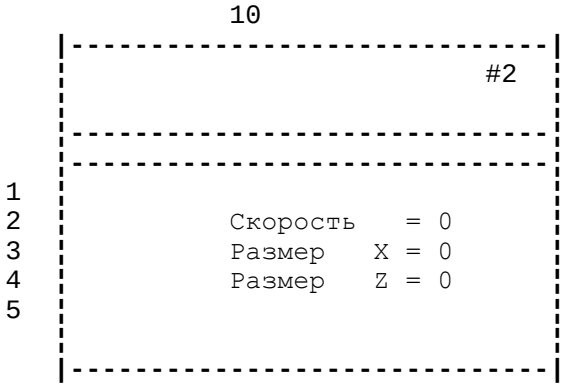


Рисунок 5.2

(INP, 1,E40, 2); ввод значения для поля 2 (X), связанное с именем программируемого поля. Значение хранится в E40 и появляется на экране

(INP, 3, E41, 4); ввод значения для поля 4 (Z). Значение хранится в E41 и появляется на экране

(INP, 5, E29, 6); ввод значения для поля 6 (скорость), значение хранится в E29 и появляется на экране

(BEQ, E29, 0, END); повтор всей последовательности до тех пор, пока не введется скорость, равная 0

```
G1 XE40 ZE41 FE29
(BNC, ENTRY)
"END" G M30
```

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

ОЗУ - оперативное запоминающее устройство;
ПЛ - программа логики станка;
ПрО - программное обеспечение;
УЧПУ - устройство числового программного управления;

PLC – программируемый логический контроллер.

ПЕРЕЧЕНЬ ТАБЛИЦ