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 ТС

Санкт-Петербург
2019г

АННОТАЦИЯ

Руководство программиста (В2.1) предназначено для ознакомления с правилами и методами составления управляющих программ для 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, работающим 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

- 2-16 управляемых осей: 8 осей в линейной интерполяции, 2 оси с перемещением «от точки к точке», 1 ось шпинделя;
- управление одновременно до 8-ми 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;
- текущее время;
- 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. Введенные символы в память программы могут быть воспроизведены на видеоэкране и модифицированы посредством удаления, модификации или вставления кадров. Эти операции могут осуществляться во время обработки детали на станке.

1.1.6 Режимы работы

Режимы работы выбираются клавишами со станочной панели. Они могут быть:

- выполнение кадров введенных 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/оборот при безразмерных ручных перемещениях;
   - 0,1 мм/оборот 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 на видеостранице #6;
- 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. При включении станка, перемещая любую ось 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

Автоматическая компенсация 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 максимально определить до 100 (от 0 до 99) абсолютных исходных точек. Например: ORA, n, Z.., X... Исходные точки активизируются 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 К=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

Можно 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 надежности 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 введённого 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%;
- скорость вращения шпинделя от 75-125%.

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 абсолютное 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, определённую конфигурируемыми осями 1 и 2;
- G18 выбирает плоскость интерполяции, определённую конфигурируемыми осями 3 и 1;
- G19 выбирает плоскость интерполяции, определённую 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, обороты/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. 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

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;
- фикции 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).

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 определённых точках обработки.

1.2.20 Сообщения программы

На видеостранице #1, в зоне сообщений могут быть воспроизведены: сообщения, переменные, константы, которые программируются посредством трёхбуквенного кода DIS.

Примеры
(DIS, "ИНСТРУМЕНТ=12")
(DIS, E37)
(DIS, U0V).

1.2.21 Коэффициент масштабирования

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

Пример
(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, которые надо применить к паре осей, для их коррекций, вычисляются автоматически. Корректировка отменяется функцией **G40**.

1.2.24 Определение припуска

Кодом U0V можно определить припуск в операциях контурной обработки. Заданный в программе или введённый с клавиатуры код U0V временно модифицирует значение корректировки на величину, равную установленному значению. Отмена припуска программируется установкой кода U0V=0.

Пример
U0V=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 **(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,Z)
.........
(MIR,X,Z)
```

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 нем - еще один (до трех 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 целые или десятичные числа
LR (действительное с двойной точностью)	E30...En	±16 целые или десятичные числа

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 приведены в таблице 1.3.

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

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

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

Посредством щупа представляется возможным выполнить три цикла измерения, программируя следующие 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 с инструментом, установленном в шпинделе, относительно закрепленного щупа, и заносит данные измерений в параметры E как и при 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, заключённой между двумя метками.

Пример
"НАЧАЛО"
....
....
"КОНЕЦ"

```
(EPP,НАЧАЛО,КОНЕЦ)
........
```

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

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:

(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 в случае, 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 путём определения в пользовательской программе параметров входа/выхода через переменные интерфейса логики пакета «К» (переменные SK) и пакета «А» (переменные SA). Системные структуры данных:

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

Примеры присвоения:

SA12=SK.BL - придаёт биту №12 пакета «А» значение первого бита структуры пакета «К».
SK5=SK7 - придаёт байту №5 пакета «К» значение байта №7 этого же пакета.

```
SA128=1          - устанавливает сигнал (бит) №128 пакета «А».
SK7.3CH="RIF"    - записывает инструкцию RIF, начиная с байта №7 пакета
                 «К».
SA3.BY=255       - придаёт значение 255 байту №3 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 пределы рабочего поля, запомненные в файлах системы, по любой оси.

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 защищённых зон

Посредством трёхбуквенного 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 определении автоматических пересечений между элементами профиля и на использовании 4 типов геометрических элементов:

- точки начала отсчёта;
- точки;
- прямые линии;
- 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 программный модуль PLC, система позволяет записывать, исправлять и проверять непосредственно на УЧПУ, при 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, систему УЧПУ-станок более надёжной.

Используя интерфейс, можно воспроизводить на видеоэкране 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 «C»-сплайн. «C»-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 отсчёта	on=Zxa	
Точки	pn=(om) Z X	точка в декартовых координатах
	pn=(om) m a	точка в полярных координатах
	pn=+lm,+lp	точка пересечения двух прямых линий
	pn=+lm,+cp(,s2)	точка пересечения линия/окружность
	pn=+cm,+lp(,s2)	точка пересечения окружность/линия
	pn=+cm,+cp(,s2)	точка пересечения 2 окружностей
Прямые линии	ln=(om)I J r,(op)I J r	линия, касательная к 2 окружностям
	ln=(om)Z X,(op)Z X	линия, проходящая через 2 точки
	ln=(om)I J r,(op)Z X	линия, касательная к 1 окружности и проходящая через 1 точку
	ln=(om)Z X,(op)I J r	линия, проходящая через 1 точку и касательная к 1 окружности
	ln=(om)I J r,a	прямая, касательная к 1 окружности и образующая угол с абсциссой
	ln=(om)Z X,a	прямая, проходящая через точку и образующая угол с осью абсциссы
	ln=+cm,+cp	линия, касательная к 2 окружностям
	ln=+cp,pm	линия, касательная к 1 окружности и проходящая через 1 точку
	ln=pm,+cp	линия, проходящая через 1 точку и касательная к 1 окружности.
	ln=pm,pq	линия, проходящая через 2 точки
	ln=+cm,a	линия, касательная к 1 окружности и образующая угол с осью абсциссы
	ln=pm,a	линия, проходящая через 1 точку и образующая угол
	ln=+lm,d	линия, параллельная другой, на расстоянии d
Окружности	cn=(om)I J r	окружность в декартовых координатах
	cn=(om)m a r	окружность в полярных координатах
	cn=+lm,+lp,r	окружность определенного радиуса, касательная к 2 прямым линиям
	cn=+lm,+cp,r	окружность, касательная к 1 прямой и к 1 окружности определенного радиуса
	cn=+cp,+lm,r	окружность определенного радиуса, касательная к 1 окружности и 1 прямой
	cn=pm,+lp,r	окружность данного радиуса, проходящая через точку и касательная к 1 прямой
	cn=+lp,pm,r	окружность данного радиуса, касательная к 1 прямой и проходящая через 1 точку
	cn=+cm,+cp,r	окружность данного радиуса, касательная к 2 окружностям
	cn=pm,+cp,r	окружность данного радиуса, проходящая через точку и касательная к окружности
	cn=+cp,pm,r	окружность данного радиуса, касательная к окружности и проходящая через точку
	cn=pm,pq,r	окружность данного радиуса, проходящая через 2 точки
	cn=pm,+lp	окружность с центром в 1 точке и касательная к 1 прямой
	cn=pm,pa,pr	окружность, проходящая через 3 точки
	cn=pm,r	окружность данного радиуса с центром в одной точке
	cn=+cm,+d	концентрические окружности с данными величинами расстояния
	cn=pm,+cp(,s2)	окружность с центром в 1 точке и касательная к 1 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. рисунки А.1-А.2).

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;
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 Информационные коды

Программа, которую необходимо выполнить, должна быть занесена в память системы. Ввод программы в память может осуществляться с клавиатуры или же посредством телетайпа. Кодом информации, перфорированным на ленте, является ISO. Символы, которые используются УЧПУ в соответствии с вышеуказанными стандартами, представлены 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 Информация управляющих программ

2.4.1 Символ

Символ - это число, буква или знак, используемые для выражения информации. Используемые символы должны соответствовать тем, которые описаны в таблице 2.1.

Пример
I,G,%,3,X,LF...

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,Z50.5, X-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. Символы, используемые для составления кадра, приведены в таблице 2.1. 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 следующей в обязательном порядке: 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 обратном 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, максимальная длина которой 6 символов, заключенная в знак « » (кавычки). 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. Номер кадра 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 4 типа кадров:

1) комментирующие кадры;
2) кадры ISO;
3) кадры назначения;
4) кадры с трёхбуквенными 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 Z500 X20 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 3 класса:

1) кадры назначения с переменными вычисления, например, E30 = 28.5;
2) кадры назначения с геометрическими переменными, например, p2 = Z10 X25;
3) кадры назначения с глобальными переменными системы, например: UOV=1.5 .

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

Пример
```
(URT,45)
```

2.6 Начало и конец программы

Если программа введена с клавиатуры, то символы начала и конца программы не обязательны. Если программа перфорирована на ленте, то первым и последним символами должны быть «%» (ISO). 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
%
```

Возможно, вставить во внутрь программы сообщение, заключённое в кавычки и предназначенное для оператора станка. Это сообщение программируется трёхбуквенным 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 по абсциссе и J по ординате. 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, определенного 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 (если в G71) или в дюймах/оборот (если в G70), если это предусмотрено в характеризации.

Функция **G93** - определяет обратное время в минутах выполнения участка, определенного 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 резания 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 (К) инструмента. Корректировка длины инструмента может быть применена к любой оси станка. Выбор зависит от названия оси, к которой присоединена корректировка длины.

Пример
Z55, X20.

Корректировка длины приводится в действие без использования других подготовительных функций. Корректировка диаметра инструмента, вызванная одновременно с корректировкой длины, приводится в действие при помощи функций G41/G42 компенсации радиуса инструмента (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		M10	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 4 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 занесен 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 запрограммировать 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 2 цифры (максимум). 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. В одном кадре можно программировать несколько операторов G, если они конгруэнтны. Таблица 2.4 «Конгруэнтность операторов G в кадре» 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 [ ] должны рассматриваться как необязательные;
2) все элементы, заключённые в { } 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 8);

[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 обратном случае, подается сигнал ошибки.

В отношении других полей действительно всё изложенное в предыдущем параграфе.

Пример линейной интерполяции см. рисунок А.3.

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 2.1 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 2.1 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).

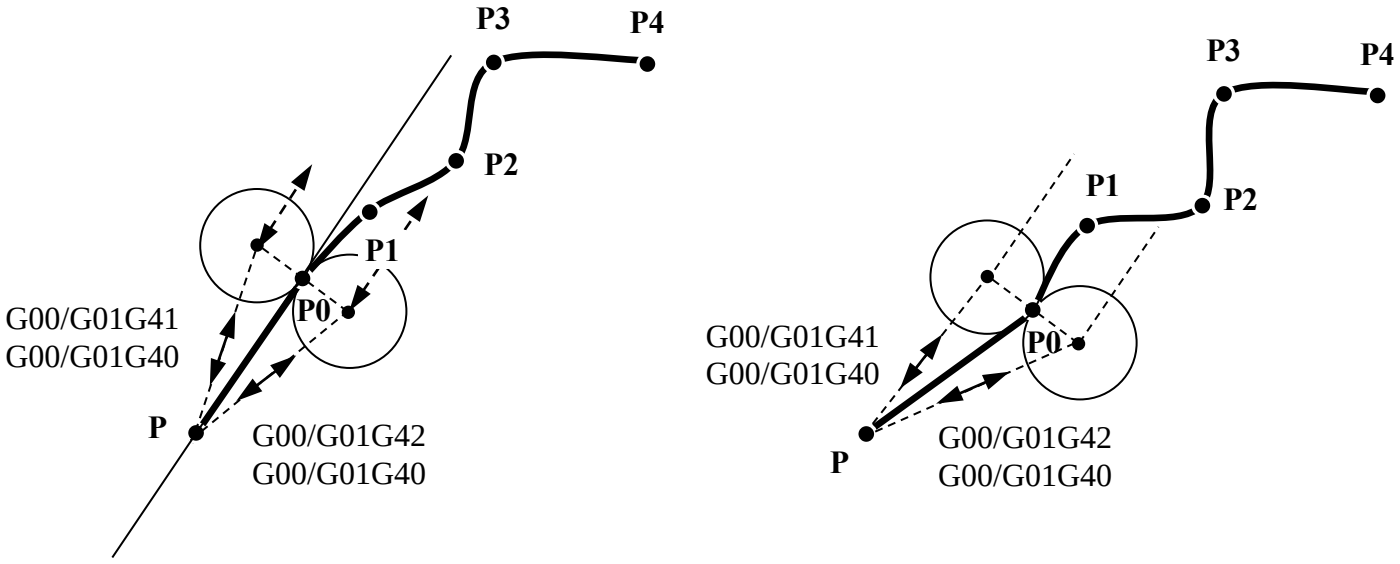


Рисунок 2.1

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

где:

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

[СКОРОСТЬ ПОДАЧИ] - при операторе G01;

[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, то она может быть опущена;

I и J - являются адресными словами, выражающими координаты центра окружности (I по абсциссе; J по ординате), 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 приведён на рисунке А.4.

Примечание - Максимальная программируемая дуга - 360 градусов.

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 179.999⁰,
- \- - для дуги от 180⁰ до 359.999⁰.

Пример программирования дуги менее 360 градусов приведён на рисунке А.5.

2.8.1.5 Плоскость интерполяции

Плоскость интерполяции определяется заранее при помощи функций G17-G18-G19 или же посредством определения плоскости, образованной парой осей, установленных кодом DPI (см. трехбуквенные коды) (G17 в любом случае присутствует при включении).

Координаты начальной точки (запрограммированные в предшествующем кадре), конечной точки и центра окружности должны быть вычислены таким образом, чтобы разница между начальным и конечным радиусом была бы меньше, чем 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.

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

2.8.1.6 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] K [I] [R] ,

где:

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

K - 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 А.6:

а) фронтальная резьба;

б) цилиндрическая резьба;
в) коническая резьба;
г) цилиндрическо-коническая резьба.

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

Примеры нарезания резьбы с переменным шагом см. на рисунке А.7:

а) цилиндрическое нарезание резьбы с возрастающим шагом;
б) коническое нарезание резьбы с возрастающим шагом;
в) 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:

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

где:
I - максимальное изменение шага;
K - начальный шаг;
ZK - координата конечной точки;
ZN - координата начальной точки;
(ZK-ZN) - длина нарезания резьбы.

Пример нарезания резьбы с 3 заходами:

```
.............
N37 G33 Z3 K6        первая нарезка
.............
.............
N41 G33 Z3 K6 R120   вторая нарезка
            .............
.............
N45 G33 Z3 K6 R240   третья нарезка
.............
```

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 отличии от других систем, в которых для осуществления многозаходной резьбы необходимо сместить начальную точку каждой нарезки на величину, равную шагу, разделенному на количество заходов.

2.8.1.7 Функция нарезания резьбы G34 (версии ПрО 1.41.31Р, 2.31Р, 2.31РИВ, 3.31Р, 4.ХХР)

Для расширения возможностей резьбонарезания параллельно существующей функции нарезания резьбы G33 установлена функция G34.

2.8.1.7.1 Нарезание резьбы с постоянным или переменным шагом (G34)

Нарезание резьбы с постоянным или переменным шагом (G34) определяет цикл цилиндрической или конической резьбы с постоянным или переменным шагом. Это движение координируется с вращением шпинделя. Запрограммированные в кадре параметры определяют тип резьбы, которую следует осуществить.

Формат:

G34 [ОСИ] К+ [I] [R] ,

где:
[ОСИ] - представлены символом имени оси и цифровым значением в явной или неявной форме (параметр Е);
К+ - шаг резьбы.

Знак величины шага определяет ось, вдоль которой выполняется резьба:

- **«+»** - вдоль оси абсцисс;
- **«-»** - вдоль оси ординат.

В случае конической резьбы знак для величины шага устанавливается в зависимости от величины перемещения по осям, определяющим конус:

- **«+»** - перемещение больше вдоль оси абсцисс;
- **«-»** - перемещение больше вдоль оси ординат.

В отличие от функции **G**33, где шаг конической резьбы задаётся по образующей конуса, для функции G34 шаг задаётся вдоль оси абсцисс или ординат.

Для фрезерного станка значение шага резьбы всегда положительно.

В случае резьбы с переменным шагом «**K**» представляет начальный шаг. Должен присутствовать всегда.

[I+] - представляет изменение шага. Для нарезания резьбы с возрастающим шагом «I» должно быть положительным, для нарезания резьбы с уменьшающимся шагом - должно быть отрицательным;

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

Примечания

1. Функция G34 программируется только с датчиком в шпинделе.
2. Во время нарезания резьбы выведены из состояния работы:
   клавиша «СТОП»;
   переключатель коррекции подачи «F»;
   переключатель скорости вращения шпинделя «S».

Для улучшения динамики выхода из резьбы по функциям G33 и G34 допускается в одном кадре с ними программировать функцию G09. В этом случае необходимо программировать конечную точку резьбы так, чтобы участок торможения был полностью в конечном пазе, иначе на конечном участке резьбы шаг будет отличным от заданного (переменным).

Примеры нарезания резьбы с постоянным шагом см. на рисунке А.8.

а) фронтальная резьба;
б) цилиндрическая резьба;
в) коническая резьба;
г) цилиндрическо-коническая резьба.

Примечание - Все параметры могут быть выражены цифровым значением в явной или неявной форме.

Примеры нарезания резьбы с переменным шагом аналогичны примерам на рисунке А.7.

а) цилиндрическая резьба с возрастающим шагом;
б) коническая резьба с возрастающим шагом;
в) цилиндрическая резьба с уменьшающимся шагом.

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

$$I <= \frac{K^2}{2*(ZK - ZN)}, \qquad (2.2)$$

где:

I - максимальное изменение шага;
K - начальный шаг;
ZK - координата конечной точки;
ZN - координата начальной точки;
(ZK-ZN) - длина нарезания резьбы.

Пример нарезания резьбы с 3-мя заходами:

```
.............
```

```
.............
N37 G34 Z3 K6        первая нарезка
.............
.............
N41 G34 Z3 K6 R120   вторая нарезка
.............
.............
N45 G34 Z3 K6 R240   третья нарезка
.............
.............
```

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 Определение плоскости интерполяции (G17-G18-G19)

Плоскость интерполяции определяется тремя функциями:

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**).

Оси 1-2-3 являются первыми тремя осями, объявленными в файле характеризации (по умолчанию, соответственно **X-Y-Z**).
Формат:

{G17}
{G18}
{G19}

Эти функции допустимы, если объявлены в кадре без другой информации.

2.8.3 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, определенные 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».

Графическое изображение динамики движения от кадра к кадру по функциям G27, G28, G29 представлено на рисунке А.12.

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. 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 и функций H и S,T недопустимо. Непрерывное функционирование временно прекращено движением по G00, которое является частью профиля. Если необходимо запрограммировать вспомогательные функции M, S, T, то программирование осуществляется в последующем кадре после G00. Примеры программирования контура при непрерывном режиме и в режиме **G00** рассматриваются на рисунке A.13.

Пример
```
N20 G1 Z-200
N21 G X200
N22 M5
```

Внутри непрерывной обработки G28 можно программировать замедление в конце кадра при помощи G09 (см. рисунок A.13).

2.8.4 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 языка GTL;
G20 - устанавливает конец геометрического профиля на базе языка GTL.

Формат:

{G20} {pn};
{G21} [ДРУГИЕ G] {ln} [s2] [ОСИ] [СКОРОСТЬ ПОДАЧИ] [ВСПОМ. 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 то же значение, что и описанное для функций G1.

Примечание - Примеры и спецификацию см. в п.2.12 («Геометрическое программирование высокого уровня GTL»).

2.8.5 Компенсация радиуса инструмента (G41-G42-G40)

На рисунке A.13 изображён теоретический профиль детали. E=0.41 x радиус инструмента. При контурной обработке профиля теоретический край P инструмента должен следовать другим путём от теоретического профиля так, что центр инструмента может расположиться на равном расстоянии от профиля детали. Результирующий профиль совпадает с теоретическим профилем только вдоль поверхностей, параллельных осям.

Используя компенсацию радиуса инструмента, вы должны только программировать теоретический профиль, т.к. управление (система) вычисляет путь теоретического края инструмента в соответствии с радиусом и ориентацией инструмента. Эти два параметра помещаются в таблицу смещений.

Значение ориентирующих кодов (от 0 до 8):

0 = P точка совпадения с центром инструмента;
1 = P точка в направлении X-Z+;
2 = P точка в направлении X-;
3 = P точка в направлении X-Z-;
4 = P точка в направлении Z-;
5 = P точка в направлении X+Z-;
6 = P точка в направлении X+;
7 = P точка в направлении X+Z+;
8 = P точка в направлении Z+.

Коды ориентации приведены на рисунке А.14.

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.

Применение функций G41 и G42 показано на рисунке А.15.

Допустимый формат:

```
G41
G42       [другие коды] [операнды]
G40
```

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

1) первое перемещение должно быть линейным, т.е. при быстром ходу или при скорости обработки (G00- G01);
2) операции черновой обработки, резьбонарезания и нарезания пазов не могут программироваться внутри цикла;
3) блоки с функциями M,H,S и T не могут программироваться внутри цикла;
4) профиль может обрабатываться в непрерывном режиме (G27-G28) или в режиме от точки к точке (G29) в автоматическом или кадровом режиме;
5) компенсация радиуса инструмента деактивизируется при помощи G40, который должен программироваться в последнем кадре профиля;
6) G00 не исключает компенсацию.

На первой и последней точке профиля центр инструмента позиционируется перпендикулярно профилю на программируемой точке. Следовательно, край теоретического инструмента зависит от кода ориентации.

На рисунке А.16 иллюстрируются начало и конец профиля с компенсацией конца инструмента. Обозначения на рисунке А.16:

P1 = программируемая точка;
P2 = позиция центра инструмента;
P3 = теоретическая позиция точки и воспроизводимая точка.

При программировании выпуклого пути перемещением против часовой стрелки радиус (r), связывающий линии, должен иметь положительную величину; при перемещении по часовой стрелки программируется отрицательный радиус. Радиус r=0 оптимизирует путь инструмента путем генерирования радиуса, равного нулю на детали (см. рис. А.18). Программирование r возможно только внутри блока G41,G42 - G40. Пример программирования сопряжения без радиуса скругления приведён на рисунке А.18.

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 А.19). Программирование b возможно только внутри блока G41,G42 - G40.

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

Пример программирования функций G41/G42/G40 приведён на рисунке А.17.

```
N29 (DIS,"FINISH")
N30 SSL = 1500
N31 T4.4 M6
N32 S100 M3M7
N33 G X64 Z7
N34 G1 G42 X50 Z4 F.1
N35 Z-20
N36 Z-5
N37 X70
N38 b2
N39 Z-55
N40 X100 Z-70
N41 r5
N42 Z-90
N43 G40 X116
```

```
N44 G X.. Z..
```

2.8.6 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]

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

2.8.7 Постоянные циклы (G80-G89)

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

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 присутсвует
G82 растачивание	рабочая подача	да	нормальное	Ускоренное перемещение к R1 или R2, если присутсвует
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 присутствует
G85 рассверливание или нарезание резьбы метчиком	рабочая подача	нет	нормальное	Рабочая подача к R1 ускоренное перемещение к R2, если присутсвует
G86 развертывание	рабочая подача; начало вращения шпинделя	нет	остановка	Ускоренное перемещение
G89 развертывание с растачиванием	рабочая подача	да	нормальное	Рабочая подача к R1 ускоренное перемещение к R2, если присутствует
G80 отмена постоянных циклов				

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:

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

Пример

Постоянный цикл с R2=R1 и R2 не равно R1 показан на рисунке A.20.

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.

ВНИМАНИЕ! ИНФОРМАЦИЯ ОБЩЕГО ХАРАКТЕРА, ИМЕЮЩАЯ ОТНОШЕНИЕ КО ВСЕМ ПОСТОЯННЫМ ЦИКЛАМ:

1. В КАДРЕ, СОДЕРЖАЩЕМ 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;
2. ПОСЛЕ ВЫПОЛНЕНИЯ ПЕРВОГО ЦИКЛА ДЛЯ ТОГО, ЧТОБЫ ВЫПОЛНИТЬ ПОСЛЕДУЮЩИЕ ЦИКЛЫ, ИДЕНТИЧНЫЕ ПЕРВОМУ, ДОСТАТОЧНО ЗАПРОГРАММИРОВАТЬ КООРДИНАТЫ ТОЧЕК ОТВЕРСТИЯ;
3. ПРОДОЛЖИТЕЛЬНОСТЬ ВЫДЕРЖКИ ВРЕМЕНИ ПРОГРАММИРУЕТСЯ ТРЁХБУКВЕННЫМ КОДОМ TMR;
4. НЕ ПРЕДСТАВЛЯЕТСЯ ВОЗМОЖНЫМ ПРОГРАММИРОВАТЬ G8X, ЕСЛИ СОСТОЯНИЕ ПРОФИЛЯ ЯВЛЯЕТСЯ АКТИВНЫМ ОТНОСИТЕЛЬНО ГЕОМЕТРИЧЕСКОГО ПРОГРАММИРОВАНИЯ И/ИЛИ ОТНОСИТЕЛЬНО КОМПЕНСАЦИИ РАДИУСА ИНСТРУМЕНТА;
5. ФУНКЦИИ G8X ЯВЛЯЮТСЯ МОДАЛЬНЫМИ. НЕВОЗМОЖНО ПРОГРАММИРОВАТЬ НОВЫЙ ПОСТОЯННЫЙ ЦИКЛ БЕЗ ЗАКРЫТИЯ ПРЕДЫДУЩЕГО ПОСТОЯННОГО ЦИКЛА С G80.

2.8.7.1 Постоянный цикл сверления (G81)

Кадр программирования: **G81 R.. Z..**

Пример цикла G81 (сверление) приведён на рисунке A.21.

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, использующая команду TMR.

Пример
```
N33 TMR=2
N34 G82 Z-100 F0.2
N35 X0
N36 G80
```
Выдержка времени в данном примере равно 2 секунды.

Пример цикла G82 для обработки пазов приведён на рисунке А.22.

2.8.7.2 Постоянный цикл глубокого сверления (G83)

Программируется с такими кадрами, как:

G83 [R]] Z..I..[K..] [J..] ,

где:

[R] - начальная координата отверстия (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) Случай, при котором были запрограммированы I, K, J, цикл имеет следующие шаги:

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

Примечание - Для сохранения параметра I неизменным (постоянное приращение) необходимо 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 оси отверстия для обработки;
- быстрый подход к размеру R;
- рабочая подача к точке R=R+I;
- выдержка времени, запрограммированная с TMR;
- подход по другой величине I.

Три последних шага следуют один за другим до достижения запрограммированного размера глубины.

Пример программирования постоянного цикла G83 приведён на рисунке А.23.

2.8.7.3 Постоянный цикл нарезания резьбы метчиком (G84)

Формат кадра цикла G84:

G84 R.. Z.. K,

где:

G84 - код цикла нарезания резьбы метчиком;

R - размер быстрого подхода и возврата на рабочей скорости;
Z - конечный размер нарезания резьбы;
K - шаг резьбы.

Пример

```
N90 (DIS,"TAP M8")
N91 G97 S280 T4.4 M6 M3
N92 G84 R5 Z-15 K2
N93 X0
N94 G80
```

Пример относится к резьбонарезанию при покое метчика и вращении детали с датчиком на шпинделе.

Если шпиндель имеет моторизованный инструмент, то возможно через вспомогательную функцию M включить шпиндель с моторизированным инструментом в револьверной головке и нарезать с моторизованным инструментом при остановке детали. В этом случае имеется две возможности: моторизированный инструмент с датчиком и без датчика.

Если моторизированный инструмент имеет датчик, цикл резьбонарезания программируется, как указано выше.

Если моторизированный инструмент не имеет датчика, программирование кадра имеет вид:

G84 R.. Z.. F.. ,

где:

G84 - фиксированный цикл нарезания резьбы;
R - размер быстрого подхода и возврата на рабочей скорости;
Z - конец размера резьбонарезания;
F - рабочая скорость.

2.8.8 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}
```

Пример программирования по приращениям приведён на рисунке А.24:

```
N12 TMR=2                          N19 Z-15
N13 G0 X50 Z-20                    N20 X-20
N14 G1 G91 G94 X-20 F...           N21 G4 X20
N15 G4 X20                         N22 G0 G90 X80
N16 Z-15                           N23 G79 X0 Z0
N17 X-20
N18 G4 X20
```

Если ни одна из этих функций не запрограммирована, то 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.

2.8.9 Характеристики динамического режима

К этому классу принадлежат следующие 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. Функция действительна только в том кадре, в котором запрограммирована;

G09 - устанавливает скорость, равную 0 в конце кадра, где она была запрограммирована, но не изменяет состояние профиля, если он находится в процессе обработки. Она действительна только в том кадре, в котором запрограммирована.

2.8.10 Измерительные циклы (G72-G73-G74)

Функциями G, определяющими измерительные циклы, являются:

G72 - измерение координат точки прямолинейным движением (с корректировкой радиуса);
G73 - измерение параметров отверстия;
G74 - измерение координат точки (без корректировки радиуса).

2.8.10.1 Измерение координат точки прямолинейным движением

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

Формат:

G72 ось [ось] [ось] En,

где:

ось - максимально три оси. Осями являются запрограммированные оси. Перемещения осуществляются в номинальных величинах;

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

Пример
G72 Z100 X50 E32
В E32 и E33 запоминаются соответственно вычисленные величины для Z,X.

2.8.10.2 Измерение параметров отверстия

Функция G73 измеряет при помощи щупа параметры отверстия в данной плоскости интерполяции и заносит их в память системы как параметры E, определённые 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 r 100 E55

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

2.8.10.3 Измерение координат точки

Функция G74 - цикл модификации. 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 три одновременных оси;
En - определяет параметр, с которого начинается запоминание измеренных смещений.

Пример

```
G74 X60 E41
E41 =Pt – Pm    ,
```

где:

Pm - измеренная точка;
Pt - теоретическая точка.

Цикл состоит из тех же фаз, что и цикл G72. Разница заключается в вычислении полученных размеров: не учитывается корректировка радиуса, и в запрограммированный параметр заносится смещение от теоретического размера.

2.8.11 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

$$F = \frac{\text{скорость подачи}}{\text{расстояние}} \qquad (2.3);$$

2) круговая интерполяция

$$F = \frac{\text{скорость подачи}}{\text{радиус}} \qquad (2.4),$$

где:

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

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

Пример

```
G93 G1 Z...X...F...
Z...X...F...
```

2.8.12 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.
В случае 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 Остановка вращения шпинделя с угловой ориентацией (M19)

При использовании вспомогательной функции M19 представляется возможным осуществить остановку шпинделя с угловой ориентацией. 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, то сначала выполняется функция, а затем - движение.

2.10 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, то в этом случае заблокированная ось будет пытаться двигаться, вследствие чего подается сигнал «СБОЙ ПРИВОДА».

Примеры

```
1) N8  G G29 Z100 X100
N10 G1 Z-100 M10 F250
N11 X-100
N12 Z100
N13 X100
N14 G X.. X.. M11
.............

2) N9  G G27 Z100 X100
N10 G1 Z-100 X100 M10 F250
N11 X-100
N12 Z100
N13 X100
N14 G Z.. X.. M11
.............
```

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

2.11 Кадры назначения глобальных переменных системы

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

1) кадры назначения переменных вычисления (см. п.2.13 «Параметрическое программирование»);
2) кадры назначения геометрических переменных (см. п.2.14 «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, указан в таблице 17). Эти переменные, значения которых могут быть получены из программы, определяют параметры, используемые во время цикла обработки.

2.11.1 Определение выдержки времени - TMR

Позволяет назначить выдержку времени в конце кадра, которая приводится в действие в кадрах с функциями G04 и/или в постоянных циклах, и/или в цикле нарезания 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 с);
- G95 значение паузы, записанное в TMR, выражено в количестве оборотов шпинделя (максимум 30000 оборотов).

TMR может присутствовать в любой части программы.

Пример

```
TMR = 12.5      назначает выдержку времени, равную 12.5 с.
E32 = 13.4
TMR = E32       назначает выдержку времени, равную 13.4 с.
```

2.11.2 Определение припуска - UOV

Определяет величину припуска, который необходимо оставить вдоль профиля. 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.

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

Пример

```
UOV = 0.5   назначает припуск равный 0.5
E30 = 1.5
UOV = E30   назначает припуск равный 1.5
```

2.11.3 Определение переменной скорости возвращения при нарезании резьбы метчиком - RMS

Определяет процент изменения скорости возврата в цикле нарезания резьбы в присутствии датчика.

Формат:

RMS = ВЕЛИЧИНА,

где:

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

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

Пример

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

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

Функции назначения переменных вх./вых. по отношению к логике интерфейса PLC позволяют установить значения непосредственно из программы в буферы данных для связи между программой и логикой интерфейса. Для этой цели в УЧПУ существует два типа структур данных, известных как пакет **«А»** и пакет **«К»**.

Пакет **«А»** определяет все физические (электрические) сигналы типа «ВКЛ.»/«ВЫКЛ.», соединяющих систему управления с металлорежущим станком. Эта структура находится в памяти и состоит из 1024 байтов.

Пакет **«К»** определяет все переменные связи между программой пользователя и логикой интерфейса по отношению к обрабатывающему центру. Более подробная информация по структурам пакета **«А»** и пакета **«К»** приведена в документе «Программирование интерфейса PLC».

Формат:

[индекс] = ВЕЛИЧИНА,

где:

[индекс] - является величиной, определяющей переменную назначения, которой должно быть присвоено значение. Её значение строго зависит от формата (формат по умолчанию бит для SA и байт для SK);

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

Пример

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

2.11.5 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 (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"
```

Если предусмотрено в фазе характеризации системы, то возможен сброс переменных при каждом включении системы. По умолчанию можно определить 200 SYVAR формата байт.

2.11.5.1 Адресация глобальных переменных системы SA-SK-SYVAR

Адресация глобальных переменных системы связана с двумя основными моментами:

- индексом программирования переменной;
- форматом обращения 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

Адресация глобальных переменных для различных форматов приведена в таблице 2.7.

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>
</table>

При программировании индекса и формата переменной во избежание наложения памяти следует учитывать количество байтов, занятых предыдущей переменной.

Если формат переменной - CH, число элементов, которое предшествует формату, указывает число символов, адресованных со стороны переменной. Значение по умолчанию - 1, максимальное значение - 32.

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

Примечание - В таблице 11 переменная S = SYVAR/SA/SK.

Пример
Адресация переменной SYVAR, начиная с SYVAR0.BY:

```
SYVAR0.BY
SYVAR1.CH
SIVAR1.IN
SIVAR1.LI
SIVAR2.RE
SYVAR2.LR
.........
```

2.11.6 Определение времени системы - TIM

Определяет группу переменных, используемых программистом для определения времени на базе системного таймера в определённых моментах обработки. Эта группа имеет 7 таймеров (от 0 до 6). Первый таймер (TIM0) зарезервирован и содержит таймер системы, который получает значение не за счёт простого назначения, а при использовании трёхбуквенного 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=TIM0       ;придаёт TIM1 время системы (часы)
...........
...........
N89 GO.....
N90 # TIM2=TIM0-TIM1  ;вычисляет время обработки от кадра N10 до кадра N90
...........
...........
(DIS,TIM2)                ;воспроизводит на видеокадре вычисленное время.
```

2.11.7 Определение общего времени - TOT

Определяет группу переменных, используемых программистом для суммирования частичного времени циклов обработки, полученного в определённые 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 времён. 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 # TOTO=TIMO-TIM1
N210 TOT1=TOTO*E1
N220 TOT2=TOT1/3600
N230 (DIS,TOT2)
```
- указывает в часах время, необходимое для выполнения партии из 75 деталей.

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

2.11.8 Предельная скорость шпинделя (SSL) при контроле постоянства скорости резания (G96)

Команда SSL определяет предельную скорость шпинделя. Это является необходимым в случае, когда система выполняет контроль постоянства скорости резания (G96).
Формат:

SSL = ВЕЛИЧИНА ,

где:

ВЕЛИЧИНА - может быть константой или параметром такого же формата.

Пример

SSL = 200 - устанавливает максимальную скорость шпинделя 200 об/мин
SSL = 1500 - устанавливает максимальную скорость шпинделя 1500 об/мин
SSL = E32 - устанавливает максимальную скорость шпинделя из параметра E32.

ВНИМАНИЕ! ПРИ ОБРАБОТКЕ В РЕЖИМЕ ПОСТОЯНСТВА СКОРОСТИ РЕЗАНИЯ (G96) НЕОБХОДИМО ВСЕГДА ПРОГРАММИРОВАТЬ SSL ДО ПЕРВОГО ПРОГРАММИРОВАНИЯ ФУНКЦИИ G96 СОВМЕСТНО С ФУНКЦИЕЙ S.

Пример

```
G97S500M3          - вращение шпинделя по ч.с. со скоростью 500 об.мин
.........
G00X70Z0           - диаметр заготовки 70 мм (размер от оси ее вращения), для
                     которого задаётся поддержание скорости резания в кадре с G96
SSL = 2000         - предельные обороты вращения шпинделя
G96 G94 S40 M3     - скорость резания 40 м/мин, при этом шпиндель автоматически
                     перейдёт на скорость вращения V*1000/π/D; 40*1000/3.14/70
                     =182 об/мин.
G1X0F60            - движение к оси вращения приведёт к уменьшению диаметра и,
                     как следствие, к увеличению скорости вращения шпинделя до
                     максимально допустимой скорости, заданной в SSL.
X70M5
```

Условия для поддержания скорости резания:

1) шпиндель должен быть с датчиком и ЦАПом;
2) поддержание скорости резания выполняется вдоль оси, указанной при описании шпинделя в инструкции ASM (секция 2 файла характеризации AXCFIL);
3) ноль детали вдоль оси, указанной в инструкции ASM, должен быть на оси её вращения.

2.11.9 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**=1÷5); «**q**» -порядковый номер оси в интерполяторе скоординированных осей процесса «**p**».

Пример

```
E25=RMN12/20
```

Примечание - 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.

2.11.10 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 ось образуют угол, не равный 90 градусов. В этом случае используется код **UAV=5**, для программирования косоугольной системы координат с помощью декартовых виртуальных координат (см. п.2.25 «Косоугольная система координат»).

Формат:

UGF = ВЕЛИЧИНА ,

где:

ВЕЛИЧИНА - определяет угол косоугольной системы координат (градусы). 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
```

2.12 Геометрическое программирование высокого уровня (GTL)

В 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).

2.12.1 Векторная геометрия

Определение профиля с использованием GTL основано на использовании 4 типов геометрических элементов:

- точки начала отсчета;
- 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 требует, кроме параметров, необходимых для установления позиции в плоскости, также назначение направления движения.

Например, прямая линия проходит через точки A и B (рисунок A.25), двигаясь от A к B, или прямая линия l', лежащая на l, но проходящая от B к A.

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 назначение 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 (см. рисунок А.26).

Направление должно быть придано также и окружностям. Условно принимается за положительное направление движение против часовой стрелки и за отрицательное - по часовой стрелке (см. рисунок А.26).

По договоренности придается положительное значение радиуса окружностям с направлением движения против часовой стрелки и отрицательное - в обратном случае.

Направление, данное элементу, обычно соответствует направлению движения по профилю. Однако возможно направление элемента во время определения профиля, если это направление противоположно остальным элементам профиля (см. рисунок А.27).

2.12.2 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;
- точки, числа пересечений;
- скоса.

Необходимость использования для этой информации строчных символов вызвана тем, что эти же заглавные символы используются в языке ЧПУ 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 отсчёта;
2) **p** - для определения точки;
3) **l** - для определения прямой;
4) **c** - для определения окружности.

ИНДЕКС - определяет номер переменной геометрического элемента. Этот номер заключен между 0 и 255. Максимальный предел определяется при конфигурации;

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 = Z30 X30 a45
p1 = o1 Z15 X15
p2 = Z60 X30
L1 = p1, p2
L2 = Z30 X50, a45
c1 = L1, L2, r15
L3 = Z0 X0, Z100 X60
p3 = L3, c1
c2 = p3, r8
```

```
.....
```

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

Примеры

```
p1 = Z30 X30
c1 = I10 J20 r30     разделитель не требуется
L1 = Z20 X20, Z100 X-10
      |         |
      точка     точка

L2 = I30 J20 r10, Z80 X80
      |           |
      окружность   точка

L3 = Z100 X100, a45
       |         |
      точка     угол

c3 = L1, L2, r18
      |   |   |
      |   |  радиус
      | прямая
    прямая
```

Дискриминатор s2 служит для выбора второго пересечения (s2). Иллюстрация примера приведена на рисунке А.28.

2.12.3 Список возможных геометрических определений

Точки начала отсчёта:

pn= Z.. X.. a..

Точки:

pn = [om] Z.. X..
pn = [om] m.. a..
pn = Lm, Lp
pn = [-]Lm, cp [,s2]
pn = cm, [-]Lp [,s2]
pn = cm, cp [,s2]

Прямые линии:

Ln = [om] Z.. X.., [op] Z.. X..
Ln = [om] Z.. X.., a..
Ln = [om] I.. J.. r.., [op] I.. J.. r..
Ln = [om] I.. J.. r.., a..
Ln = [om] I.. J.. r.., [op] Z.. X..
Ln = [om] Z.. X.., [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] l.. 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..
```

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

2.12.4 Определение точек начала отсчёта

Функция определения точек начала отсчета 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 = Z.. X.. a..,
```

где:

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

Пример приведён на рисунке А.29.

2.12.5 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 отсчёта состоит из начальной точки, названной полюсом, из которой начинается ось X, названная полярной осью (рисунок А.30).

Любая точка плоскости может быть определена при помощи длины отрезка (P), названной модулем, который соединяет её с полярной точкой при помощи величины угла, который образует отрезок прямой с полярной осью (рисунок А.31).

Формат:

1) Прямой

Точка в декартовых координатах (рисунки А.34-А.35):

```
pn = [on] Z.. X..
```

Точка в полярных координатах (рисунок А.36):

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

2) Косвенный

Точка пересечения двух прямых, определённых ранее (рисунок А.37):

```
pn = lm,lp
```

Точка пересечения прямой и окружности, определённых ранее (рисунок А.38):

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

Точка пересечения двух окружностей (рисунок А.39):

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

где:

Pn - определяет название точки индекса n (n - число, заключённое между 1 и максимально конфигурируемым числом);
Z.. X.. - координаты точки;
[on] - начальная точка отсчёта индекса n, определённая ранее, к которой относятся координаты Z и X;
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 - возможно изменить направление, вставляя знак «-»;
[,s2] - индикатор второго пересечения.

В случае пересечения прямая-окружность или наоборот, существуют два возможных решения (рисунок 47): 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). Пример пересечения прямая-окружность показан на рисунке А.38.

Также в случае пересечения окружность-окружность существуют два возможных решения: окружности c1 и c2 пересекаются в точках p1 и p2 (рисунок А.48). 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).

Пример пересечения окружность-окружность приведён на рисунке А.39.

2.12.6 Определение прямой линии

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

Направление прямой всегда от первого ко второму определённому элементу. 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 направления движения геометрических элементов показана на рисунке А.40. Совместимость направления движения показана рисунке А.41.

Формат:

1) **Прямой в явном виде**

Прямая, проходящая через две точки (рисунок А.42):

ln = [om] Z..X.., [op] Z..X..

Прямая, проходящая через одну точку и образующая угол с осью абсциссы (рисунки А.44-А.45):

ln = [on] Z..X..,a..

Прямая, касательная к одной окружности и образующая угол с осью абсциссы (рисунки А.46-А.47):

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

Прямая, касательная к двум окружностям (рисунки А.48-А.49):

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

Прямая, касательная к окружности и проходящая через точку (рисунок А.43):

ln = [om] I..J..r.., [op] Z..X..
ln = [om] Z..X.., [op] I.. J..r..

2) **Косвенный**

Прямая, проходящая через две точки (рисунок А.50):

ln = pm, pq

Прямая, проходящая через точку и образующая угол с осью абсциссы (рисунок А.54):

ln = pm, a..

Прямая, касательная к двум окружностям (рисунки А.52-А.53):

ln = [-]cm, [-] cp

Прямая, касательная к одной окружности и образующая угол с осью абсциссы (рисунок А.55):

ln = [-] cm, a..

Прямая, касательная к окружности и проходящая через точку (рисунок А.51):

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

Прямая, параллельная прямой на расстоянии **d** (рисунки А.56-А.57):

ln = [-] lm, d..,

где:

Ln - определяет название прямой с индексом n (n - число, заключённое между 1 и максимально конфигурируемым номером);
Z..X.. - 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 индексом m и q;
[-] cm [-] cp - предопределённые элементы окружности с индексом m и q.

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

[-] lm - предопределённый элемент прямой с индексом m;
d.. - расстояние между двумя прямыми положительное, если прямая находится слева, отрицательное - в обратном случае, смотря в направлении предопределённом прямой.

2.12.7 Определение окружностей

Функция определения окружностей 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 направление элементов не учитывается, то, задавая окружность известного радиуса и прямую линию, возможно от 1 до 8 решений получения окружности, касательной к двум элементам.

Окружности, касательные к прямой и окружности показаны на рисунке А.58.

Учитывая совместимость направлений движения предопределённых элементов и окружности, которую следует определить, количество возможных решений уменьшается до двух.

Для различения двух возможных окружностей, имеющих одно и то же направление и один и тот же радиус, необходимо учитывать направление элементов, к которым относится определение, и две дуги окружности, в которой, если необходимо, элемент делится точками касания с предраспределёнными элементами: GTL всегда производит окружность с направлением от первого ко второму элементу, с дугой, имеющей меньший центральный угол. Окружности, касательные к меньшей дуге см. рисунок А.59.

Окружность c3 получается при введении в определение прямой l1 в первой позиции и окружность 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 введении в определение 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 предопределённым окружностям: в этом случае возможны от 1 до 8 решений см. рисунок А.60.

Окружности, касательные к двум предопределённым окружностям, показаны на рисунке А.61.

Совместимость направлений движения предопределённых 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 разделён точками, касательными к элементам начала; GTL создаёт окружность, двигаясь от первой окружности ко второй, с дугой, имеющей меньший центральный угол (рисунок А.61). 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.

2.12.7.1 Формат прямого программирования

Окружность с декартовыми координатами центра и радиуса (рисунки А.62-А.63):

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

Окружность с полярными координатами центра и радиуса (рисунок А.64):

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

где:

cn	- устанавливает название окружности индекса n (n - номер, заключенный между 1 и макс. 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 индексом m, p. Может принять противоположное направление при использовании знака «-»;
[s2]	- атрибут для наибольшей из двух возможных окружностей;
d..	- расстояние между двумя окружностями положительное, если, смотря из [-] **cm**, **cn** находится слева от неё, отрицательное, если справа.

2.12.7.2 Формат косвенного программирования

Окружность с данным радиусом и касательная к двум предопределенным прямым (рисунок А.65):

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

Окружность с данным радиусом и касательная к прямой и окружности (предопределенным) (рисунки А.66-А.68):

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

Окружность с данным радиусом, проходящая через предопределенную точку и касательная к предопределенной линии (рисунок А.69):

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

Окружность с данным радиусом и касательная к двум предопределенным окружностям (рисунки А.70-А.71):

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

Окружность с данным радиусом, проходящая через одну предопределенную точку и касательная к предопределенной окружности (рисунок А.72):

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

Окружность с данным радиусом, проходящая через две предопределенные точки (рисунок А.73):

cn = pm, pq, r..

Окружность с центром в предопределенной точке и касательная к предопределенной прямой (рискнок А.74):

cn = pm, [-] lp

Окружность с центром в предопределенной точке и касательная к предопределенной окружности (рисунок А.75):

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

Окружность, проходящая через три точки (рисунок А.76):

cn = pm,pq,pr

Окружность с данным радиусом и с центром в точке (рисунок А.77):

cn = pm,r..

Окружность концентрическая к предопределенной окружности и отдаленная от нее на данную величину (рисунок А.78):

cn = [-] cm,d..,

где:

cn	- устанавливает название окружности индекса n (n - номер, заключённый между 1 и макс. 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 индексом m, p. Может принять противоположное направление при использовании знака «-»;
[s2]	- атрибут для наибольшей из двух возможных окружностей;
d..	- расстояние между двумя окружностями, положительное, если, смотря из [-] cm, cn находится слева от неё, отрицательное, если справа.

2.12.8 Определение профиля

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

2.12.8.1 Начало и конец профиля

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

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

2.12.8.2 Открытый профиль

Если профиль открытый, то 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, программируя в кадре функции G21 G41/G42, и закрыта на последней точке с функциями G20 G40. Пример приведён на рисунке А.79.

2.12.8.3 Закрытый профиль

Если профиль закрытый, то сначала следует запрограммировать последний элемент (см. рисунки А.80-А.81), а затем, после последнего элемента - вызвать первый элемент профиля. Первая точка скорректированного профиля - пересечение первого и последнего смещенных элементов (первая точка = последней точке). Компенсация радиуса должна быть в начале профиля в кадре вызова последнего элемента, программируя функции 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, то возможны два пересечения. Если не дается 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 А.81).

2.12.8.4 Примеры открытых и закрытых профилей

Если в кадре начала профиля запрограммировано pn, то это означает, что профиль открытый; pn может быть запрограммировано в начале и в конце профиля, но не внутри профиля.

Открытый профиль (см. рисунок А.79):

```
.............
l1 = ZX 25,a
p1 = Z-20 X25
p2 = Z90 X25
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.

Закрытый профиль (см. рисунок А.80):

```
.............
l5 = Z X-15,a180
.............
l1 = Z-30 X-15,a135
.............
G21 G42 15          - последний элемент
l1                  - первый элемент
.............
l5                  - последний элемент
G20 G40 l1          - первый элемент
```

Закрытый профиль (см. рисунок А.81):

```
c1 = I.. J.. r..
.............
l1 = Z.. X..,a90
.............
l5 = Z.. X..,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.

2.12.8.5 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                       .........
                         .........
```

2.12.9 Соединение геометрических элементов

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

2.12.9.1 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 А.82. В случае пересечения прямой с окружностью первое и второе пересечения определяются направлением движения прямой линии. В случае пересечения окружности с окружностью, как показано на рисунке А.83, первое пересечение то, что слева от прямой, соединяющей центр первой окружности с центром второй, а второе пересечение то, что справа от той же прямой линии.

2.12.9.2 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 А.84.

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 закончиться 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 (r=0). Примеры приведены на рисунках А.84-А.85.

2.12.9.3 Скосы

Можно определить скосы между прямолинейными элементами, программируя значение скоса без знака, рассматриваемое как расстояние от точки пересечения. Пример приведён на рисунке А.86.

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 с высоким значением.

2.12.10 Примеры программирования при помощи GTL

Рассмотрим следующие примеры:

1) примеру соответствует рисунок А.87:

```
N1  (UCG,2,Z-90Z5,X-5X125)
N2  p1=Z0 X50
N3  l1=p1,a180
N4  c1=I-40 J30 r20
N5  l2=Z-40 X30,a100
N6  p2=Z-85 X120
N7  l3=p2,a180
N8  T1.1 M6 S… F… M3
N9  G X100 Z10
N10 G21 G42 p1
N11 l1
N12 r-8
N13 c1
N14 r-10
N15 l2
N16 r8
N17 l3
N18 G20 G40 p2
N19 G X150 Z55
```

2) примеру соответствует рисунок А.88:

```
(UCG,2,Z-85Z5,X-5X75)
p1=Z0X60
l1=p1,a90
l10=Z0 X61.7,a195
l2=l10,d-3.5
c1=I-42.2 J28 r20
l3=c1,a215
c2=I-67.6 J64 r2
l4=c2,a105
p2=Z-80 X68
l5=p2,a180
T1.1 M6 S… F… M3
G0 X65 Z2
G21 G42 p1
l1
r 0.5
l2
r-3
c1
l3
r-5
l4
c2
l5
G20 G40 p2
G0 X100
```

2.13 Параметрическое программирование

Используя коды E, можно параметрически 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.

Таблица 2.8 – Описание параметров E для различных форматов

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

Примечание – В таблице 2.8 (*)означает максимальное число параметров E, определённое во время конфигурации.

Параметры E получают значения в кадрах назначения.
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).

Арифметические операции:

1) **+** сложение;
2) **-** вычитание;
3) ***** умножение;
4) **/** 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 индекса B (1,2,3) из геометрического элемента линии (прямой) индекса A (1 = синус угла, 2 = косинус, 3 = расстояние от прямой линии до начальной точки);
- **FEP (A,B)** - извлекает элемент индекса B (1,2) из геометрического элемента точки индекса A (1 = абсцисса точки, 2 = ордината);
- **FEC (A,B)** - извлекает элемент индекса B (1,2,3) из геометрического элемента окружности индекса A (1 = абсцисса центра, 2 = ордината, 3 = радиус окружности).

Значения для (A) и (A,B) могут быть параметрами E или числовыми константами. Выражение вычисляется с учетом 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 придает результат параметру E37.
"LAB1"E51=-0.00000124+5	- выполняет вычисление выражения и придает результат параметру E51.
E40=TAN(35)	- извлекает тангенс 35 градусов и придает результат параметру E40.
/E35=FEP(37,1)	- извлекает абсциссу точки p37, ранее занесенной в память и придает 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 кон-
                      станты 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=...
```

Обычно параметры остаются запомненными при выключении станка, 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)	функции T	TE14.E15
E20..E24 (LI)		
E25..E29 (RE)	функции F коды URT коды SCF индексированные оси код UGF	FE27 (URT,E25) (SCF,E26) PE29 UGF=E28
E30...(*) (LR)	координаты осей C X Z координаты R составные операции IJK глобальные переменные системы: TMR UOV	 XE32 RE33 KE34 TMR=E38 UOV=E40

2.14 Кадры с трёхбуквенными операторами

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

- операторы, изменяющие систему начала отсчёта осей;
- операторы, изменяющие последовательность выполнения программы;
- смешанные операторы;
- операторы ввода/вывода;
- операторы контроля инструмента;
- операторы видеографического управления;
- операторы управления коррекциями.

2.14.1 Трёхбуквенные операторы, модифицирующие систему отсчёта осей

Ими являются все операторы, позволяющие изменять в плоскости декартовую систему отсчёта, по отношению, к которой был запрограммирован профиль. К этому классу принадлежат следующие операторы: UAO,UOT,UIO,MIR,URT,SCF,RQO.

2.14.1.1 Использование абсолютных исходных точек - UAO

Выбирает одну из абсолютных исходных точек, ранее определенных командой ORA.
Формат:

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

где:

N - определяет номер исходной точки, которую надо выбрать. Может быть цифровой постоянной или параметром Е типа целый (от E10-E19);

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

Пример

(UA0,1) - абсолютная исходная точка 1, приведенная в действие для всех осей.

..... - программа, отнесенная к исходной точке 1 для всех осей.

(UA0,2,Z,X) -абсолютная исходная точка 2, приведенная в действие для осей Z и X.

(UA0,3,B) - абсолютная исходная точка 3, приведенная в действие для оси B.

..... - программа, отнесенная к исходной точке 2 для осей ZX, к точке 3 для оси B и к точке 1 для всех остальных осей.

(UA0,0) - возвращение в нулевую начальную точку для всех осей.

При включении и после команды «СБРОС» автоматически приводится в действие нулевая исходная точка для всех осей. Максимально могут присутствовать 8 «названий осей». Не могут быть определены одинаковые «названия осей». Если требуется привести в действие различные исходные точки для различных осей, необходимо программировать столько кадров с этими операторами, сколько имеется исходных точек. Если выбранная начальная точка (-n) загружена в файл альтернативной системы измерения, она автоматически переводится в текущую систему измерения.

2.14.1.2 Определение и использование временных исходных точек - UOT

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

Формат:

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

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 исходной точке для той оси. Для необъявленных осей остаётся в силе текущая начальная точка.

Пример

(UA0,0) - активизируется абсолютная исходная точка 0. Программа, отнесенная к абсолютной исходной точке 0 для всех осей:

1) (UOT,0,Z100,X100)
..... применяется временная исходная точка к исходной точке 0 с корректировками Z100 и X100 (временная исходная точка).

2) (UOT,1.Z-250,X-50)
..... применяет временная исходная точка к абсолютной исходной точке 1 с корректировками Z-250 и X-50.

(UA0,0) активизируется абсолютная исходная точка 0 для всех осей.

Пример приведён на рисунке А.89.

По крайней мере, должен присутствовать один операнд оси. Максимально могут присутствовать 8 осей. Не могут быть определены операнды осей с одним и тем же названием. Временная исходная точка остаётся активной до того, как определяется новая временная исходная точка или до вызова абсолютной исходной точки, или до команды «СБРОС». Размер в операторе UOT необходимо программировать в текущей размерности (G70/G71).

2.14.1.3 Определение и использование исходных точек по приращениям - UIO

Эта команда позволяет инкрементально переместить текущую исходную точку для всех программированных в команде осей.
Формат:

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

где

VAR-i - представляет ось и размер. Система берет размер как абсолютное смещение и прибавляет его к абсолютной исходной точке для данной оси. Для необъявленных осей текущая исходная точка остается в силе.

Пример приведён на рисунке А.90.

Инкрементальная исходная точка остаётся в силе до её переопределения с новой командой UIO, или восстанавливается абсолютная исходная точка при помощи (UAO,O) или [СБРОС].
Если команда UIO относится к оси X, то программируемое значение является радиальным, а не диаметральным.

2.14.1.4 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

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

Зеркальная обработка на программированную ось начинает действовать с первого движения данной оси после команды MIR. Инверсия осуществляется вокруг текущей начальной точки. Максимально может быть запрограммировано 8 осей. Не представляется возможным программировать 2 раза одну и ту же ось. Если присутствуют команды вращения (URT) и зеркальной обработки (MIR), то они устанавливаются в следующем порядке: MIR и URT.

2.14.1.5 Поворот плоскости - URT

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

(URT , ОПЕРАНД) ,

где:

ОПЕРАНД - представляет величину угла, выраженную в градусах и десятичных долях градуса; может быть выражен явно или неявно (параметр E типа от E25 до E29). Если операндом является «0», то 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), то они устанавливаются в следующем порядке: MIR и URT.

Пример поворота плоскости изображён на рисунке А.91:

```
(UOT,0,Z100,X50)
(URT,30)
..........
..........
..........
(UAO,0)
(URT,0)
```

Пример поворота профиля в случае программирования на языке GTL при виртуальных осях (рисунок А.92):

```
N75  (DIS,"MILL D=8")
N76  M21
N77  GX30
N78  G94 G97 C0 Z5
N79  (UAV,1,XC,UV,10)
N80  (DPI,U,V)
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  F170 S800 T1.1 M6 M3
N88  E26=0
N89  (RPT,6)
N90  (URT,E26)
N91  G U15 V15
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)
N105 (UAV,O)
N106 CX80
N107 (DPI,Z,K)
N108 M20
N109 GG79 X Z M30
```

Пример программирования для повторения профиля (8 раз) при виртуальных осях (рисунок А.93):

```
N1  (DIS,"GTL WITH ROTATION")
N2  M21
N3  G0 G94 G97 C0 X120 Z5
N4  (UAV,1,XC,UV,30)
N5  (DPI,U,V)
N6  F.. S.. T2.2 M6
N7  UOV=2
N8  p1=U50 V0
N9  C1=I0 J0 r50
N10 C2=I0 J0 r10
N11 l1=C2,a180
N12 l3=U0 V0,a45
N13 l2=C2,a45
N14 p2=l3,C1,S2
```

```
N15 U60 V0
N16 Z-10
    "START" N17 E25=0
N18 (RPT,8)
N19 (URT,E25)
N20 G21 G42 p1
N21 c1
N22 r3
N23 l1
N24 r-3
N25 l2
N26 r3
N27 c21
N28 G20 G40 p2
N29 E25=E25+45
N30 (ERP)
N31 (URT,0)
    "END" N32
N33 UOV=0
N34 (EPP,START,END)
N35 (UAV,0)
N36 (DPI,Z,X)
N37 G0 M20
N38 Z5
N39 G79 X Z M30
```

2.14.1.6 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 применен. Может быть запрограммирован как явно, так и неявно при помощи параметра E типа RE (от E25 до E29);

VAR-1 - символ, который представляет одну из осей, для которой приведён в действие коэффициент масштабирования. Для необъявленных осей масштабирование отменяется. Если к SCF не присоединен никакой операнд, то масштабирование отменяется для всех осей.

Пример

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

Примечание - Может быть запрограммировано максимально 6 названий осей.

2.14.1.7 Модификация исходной точки - RQO

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

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

где:

N - определяет номер модифицируемой исходной точки. Его величина заключена между 0 и 99 и тесно связана с числом записей, определённых во время создания файла исходных точек. Может быть выражена 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 исходной 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 8 операндов (осей). Не может быть определено больше одного операнда с одним и тем же названием оси. Исходная точка изменяется как в файле исходных точек (следовательно, результат модификации постоянный), так и в памяти пользователя, если эта исходная точка активна для оси в момент модификации.

В файле начальных точек, модификации придаются 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.

2.14.2 Трёхбуквенные операторы, которые изменяют последовательность выполнения программы

К этому классу принадлежат следующие операторы:

- **RPT** - повторить набор кадров;
- **ERP** - определить конец повторяющего набора;
- **CLS** - вызвать подпрограмму для выполнения;
- **EPP** - выполнить подпрограмму;
- **BNC** ¦
- **BGT** ¦
- **BLT** ¦
- **BEQ** > операторы ветвления
- **BNE** ¦
- **BGE** ¦
- **BLE** ¦

2.14.2.1 Повторение частей программы - RPT

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

Формат:

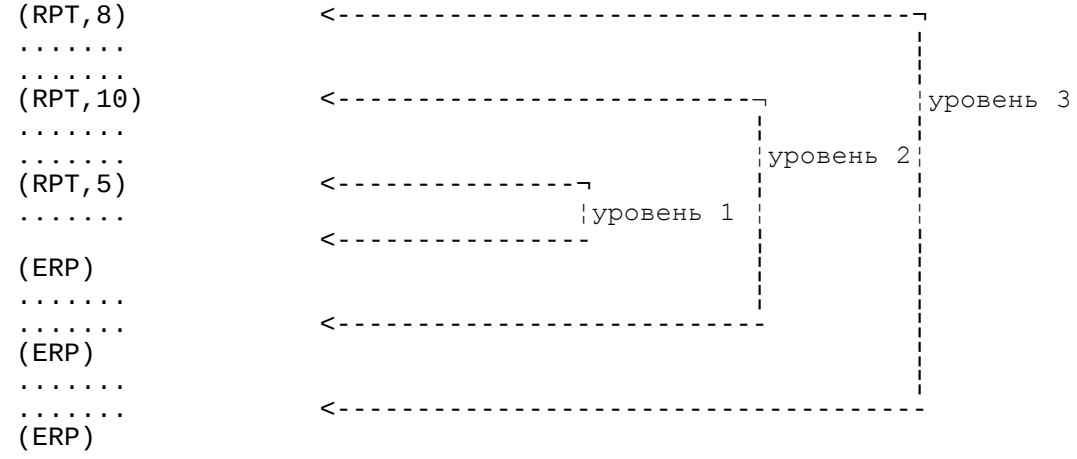
(RPT,n) ,

где:

n - число повторений; должно быть целым числом, заключённым между 0 и 99 и может быть представлен параметром E типа байт (от E0 до E9). Последующая схема иллюстрирует, как может быть определён цикл обработки.

Допущено 3 уровня повторений, т.е. можно программировать до 2 операторов RPT внутри другого оператора RPT.

Пример



2.14.2.2 Использование подпрограммы - CLS

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

Формат:

(CLS, ФАЙЛ [/MPx]) ,

где

ФАЙЛ - название программы для вызова;

MPx - название памяти MPx (x=0÷3), которое содержит программу. Если определение MPx отсутствует, то применяется то, которое объявлено по умолчанию в секции 4 файла PGCFIL.

Синтаксические обязательства: - Алфавитно-цифровая последовательность, которая заменяет файл, может иметь максимально 6 символов, отделено от устройства символом «/». Устройство заменяется последовательностью двух или трёх алфавитно-цифровых символов. Алфавитные символы, которые используются для **«ФАЙЛ»** и **«УСТРОЙСТВО»,** могут быть только заглавными; первым символом должна быть буква.

Пример

```
N1 (CLS,P800/MP2)
```

Передает управление подпрограмме P800, расположенной в памяти MP2 (если MP2 определена по умолчанию, название не записывается).

```
Основная программа                Подпрограмма P800
N16 ...........
N17 (CLS,P800)    ----------------> N500 ..........
N18 ...........            |       N501 ..........
...............            |       N502 ..........
N67 (CLS,P800)    -----------       N503 ..........
N68 ...........                     ...............
```

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.

2.14.2.3 Выполнение части программы - EPP

Оператор **EPP** выполняет часть программы, заключенную между двумя метками, определенными в операторе.

Формат:

(EPP, МЕТКА1,МЕТКА2) ,

где:

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

МЕТКА - это алфавитно-цифровая последовательность, состоящая из максимально 6 символов, заключенная 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 используются один и тот же набор 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.

2.14.2.4 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 в УП

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

где:

- **VAR1, VAR2** - являются переменными, на базе которых проверяется отношение; могут быть параметрами, сигналами логики станка, глобальными переменными системы, цифровыми значениями или последовательностью символов.
- **METKA** - для него действительны те же правила, что 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 переменная 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, то его значением по умолчанию принимается значение «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, то 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)
```

2.14.3 Примеры программирования

2.14.3.1 Программирование повторений

Пример программирования повторений изображён на рисунках А.93-А.94.

Используя код RPT совместно с инкрементальной исходной точкой UIO, можно получить несколько одинаковых элементов профиля, программируя только первый из них относительно его нулевой точки. Пример приведён на рисунке А.95.

2.14.3.2 Использование параметрического программирования и команд RPT

Пример использования параметрического программирования и команды RPT приведён на рисунке А.96.

Выполнение прямоугольной резьбы, глубиной 10 мм, при помощи десяти 1 мм проходов.

```
-------------------
N18 (DIS, "TEST RPT")
N19 G97 S300 T2.2 M6 M3 M8
N20 E31=98
N21 G0 X104 Z10
N22 (RPT, 10)
N23 G0 XE31
N24 G33 Z-200 K20
N25 G0 X104
N26 Z10
N27 E31=E31-2
N28 (ERP)
--------------------
```

Этот цикл может также быть запрограммирован с использованием обычной команды переходов, приписывая текущий диаметр параметру E31, конечный диаметр - параметру E32 и глубину прохода - параметру E33.

Пример

```
---------------------------
N18 (DIS,"TEST CONDITIONAL JUMPING")
N19 G97 S300 T2.2 M6 M3 M7
N20 E31=98
N21 E32=80
N22 E33=1.8
N23 G Z10
"CONT" N24 XE31
N25 G33 Z-200 K20
N26 G0 X104
N27 Z10
N28 E31=E31-E33
N29 (BGT, E31, E32, CONT)
N30 XE32
N31 G33 Z-200 K20
N32 G0 X104
N33 Z..
---------------------------
```

2.14.3.3 Подпрограмма без параметров

Примеры определения цикла обработки паза приведены на рисунках А.97-А.98.

2.14.3.4 Подпрограмма стандартного резьбонарезания

Пример подпрограммы стандартного резьбонарезания приведён на рисунке А.99:

Параметры:

E30 - внешний диаметр;

```
 E31 - внутренний диаметр;
 E32 - начало Z;
 E33 - конец Z;
 E34 - угол инструмента;
 E35 - число проходов;
 E36 - шаг;
 E37 - расстояние безопасности.

Определение подпрограммы резьбы:

 E40=(E30-E31/2)
;E47=диаметр возврата
 E47=e30+2xE37xE35/ABS(E35)
 E44=TAN(E34/2)
 E1=ABS(E35)
;E41=глубина первого прохода
 E41=E40/SQR(E1)
 E42=0
 (RPT, E1)
 E42=E42+1
 E43=E41xSQR(E42)
;E45=X REAL  E46=Z REAL
 E45=E30-2xE43
 E46=E32-E43xE44
 G0 ZE46
 XE 45
 G33 ZE33 KE36
 G0 XE47
 (ERP)
 G0 ZE32
```

Вызов подпрограммы резьбы:

```
-------------------
-------------------
-------------------
N24 T3.3 M6
N25 G97 S1200 M3 M7
N26 E30=E40
N27 E31=37.4
N28 E32=3.
N29 E33=-30
N30 E34=60
N31 E35=8
N32 E36=2
N33 E37=1
N34 (CLS, THREA)
-------------------
```

2.14.3.5 Параметрическая подпрограмма треугольной или прямоугольной резьбы

Пример применения параметрической подпрограммы треугольной или прямоугольной резьбы (рисунок А.100):

```
 Параметры:
 E30 = внешний диаметр
 E31 = внутренний диаметр
 E32 = начало Z
 E33 = конец Z
 E34 = угол инструмента
 E35 = число проходов (отрицательное для внутреннего резьбонарезания)
 E36 = шаг
 E37 = расстояние безопасности (радиальное значение)
 E38 = ширина инструмента
 E39 = ширина резки

Определение подпрограммы "TRAPEZ"
   ;E40 = глубина резьбы
    E40 = E30-E31
```

```
;проверка параметров
 E11 = (E30-E31)*E35
(BLT, E11, 0, ERR)
E45 = E39-ABS (E40)*TAN (E34/2)
(BLT, E45, E38, ERR)
E41 = E40/ABS (E35)
;E42 = диаметр возврата
E42 = E30+2xE37xE35/ABS (E35)
E43 = ABS (E41)xTAN (E34/2)
E1 = ABS (E35)
(RPT, E1)
E30 = E30-E41
E39 = E39-E43
;E2 = число проходов для выполнения резьбы
E2 = INT ((E39/E38)-0.001)+1
E44 = (E39-E38)/(E2-1)
;E32 = реальное начало Z
E32 = E32+E43/2
;E45 = реальное Z
E45 = E32
(RPT, E2)
G0 ZE45
XE30
G31 ZE33 KE36
G0 XE42
E45 = E45+E44
(ERP)
(ERP)
G0 ZE32
(BNC, END)
"ERR" (DIS, "WRONG PARAMETERS")
M0
"END"
```

Вызов "TRAPEZ"

```
--------------------
--------------------
N10 (DIS, "ACME THREAD")
N11 T4.4 M6
N12 G97 S300 M3 M7
N13 E30=100
N14 E31=90
N15 E32=5
N16 E33=-100
N17 E34=30
N18 E35=12
N19 E36=15
N20 E37=1
N21 E38=8
N22 E39=15
N23 E40 (CLS, TRAPEZ)
--------------------
--------------------
```

2.14.3.6 Использование команды EPP для черновой и чистовой обработки профиля

Пример использования команды EPP для черновой и чистовой обработки профиля (рисунок А.101):

```
N0  (DIS, "PREFINISHING")
N10 S150 T1.1 M6 M3 M7 F0.3
UOV=0.3
"START" G0 G42 X.. Z.. (точка 1)
----------------------
----------------------
"END" G40 X.. Z..      (точка 5)
----------------------
----------------------
N70 (DIS, "FINISHIHG")
```

```
N80 S200 T2.2 M6 M3 M7 F0.15
N81 UOV=0
N82 (EPP, START, END)
```

2.14.3.7 Использование параметрического программирования и подпрограмм для обработки параболических профилей

Пример использования параметрического программирования и подпрограмм для обработки параболических профилей (рисунок А.102):

Параметры:
E31 = расстояние фокуса (двойной фокус)
E32 = приращение по Z
E33 = начало Z
E34 = конец Z

Подпрограмма PARAB:

```
G1 G42 X Z E33
"START" E33=E33-E32
(BLT, E33, E34, END)
E35=2*SQR(2*E31*ABC(E33))
XE35 ZE33
(BNC, START)
"END" E35=2*SQR(2*E31*ABS(E34))
G40 XE35 ZE34
E35=E35+10
G XE35
```

Главная программа:

```
N   T1.1 M6 S.. F..
N.. G X Z5
E31 = 52
E32 = 2
E35 = 0
E34 = -168.8
(CLS, PARAB)
G Z
------------------
------------------
```

2.15 Трёхбуквенные операторы смешанного типа

К этому классу принадлежат следующие операторы:

1) **DPI** - определение плоскости интерполяции;
2) **DTL** - определение величины допускаемого отклонения позиционирования;
3) **DLO** - определение рабочего поля;
4) **CTL** - переключение в режим токарного или фрезерного станка;
5) **DSA** - определение защищенных зон.

2.15.1 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).

2.15.2 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 - 8. Оператор DTL вызывает ошибку, если активизированы следующие функции:

- GTL (G21);
- компенсация радиуса инструмента;
- 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 **ОШИБКИ ПРИВОДА**.

2.15.3 Определение рабочего поля - DLO

Оператор **DLO** определяет рабочее поле для осей, запрограммированных в операторе, с учетом фактической исходной точки. Для незапрограммированных осей сохраняется значение, которое было активным ранее. Если запрограммированное значение превышает предел, объявленный в файле характеризации, то 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 исходной 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 8 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 (рисунок А.103):

```
(DLO,Z-10 X-70)
..............
(DLO,Z-55 X10)
..............
```

2.15.4 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, F) - активизирует конфигурацию фрезерного варианта,

или:

(CTL) - возвращает управление исходной конфигурации токарного варианта.

Кнопка «**СБРОС**» восстанавливает конфигурацию токарного варианта.

2.15.5 Защищенные зоны - DSA, ASC, DSC

Эти команды позволяют активизировать или деактивизировать защищённые зоны, т.е. зоны, куда вход координат запрещён.

Команды этого класса:

- **DSA** - определяет защищённую зону;
- **ASC** - активизирует защищённую зону;
- **DSC** - деактивизирует защищённую зону.

Управление производит проверку относительно защищённых зон до начала движения. Из программы можно определить до 3-х защищённых зон, относительно текущей исходной точки.

Допустимые форматы:

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

где:

n - номер зоны;
Z - нижняя граница по Z;
Z+ - верхняя граница по Z;
X- - нижняя граница по X;
X+ - верхняя граница по X.

Защищённые зоны могут быть отменены при помощи кнопки «**СБРОС**».

Пример применения защищённых зон (рисунок А.104):

```
N1  (DSA, 1,Z-5 Z100, X0 X50)
N2  (DSA, Z, Z250 Z-200, X0 X180)
N3  (ASC, 1)
N4  (ASC, 2)
N5  T1.1 M6
------------
N80 (DSC, 1)
------------
N99 M30
```

2.16 Трёхбуквенные команды ввода/вывода

Эти команды позволяют выполнять операции входа/выхода из программы. Команды этого класса:

- **DIS** - вывод переменной на экран;
- **DLY** - установка задержки;
- **USS** - вращение моторизированнного инструмента.

2.16.1 Вывод переменной на экран - DIS

Команда **DIS** позволяет вывести на экран значения, определяемого 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. 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, "THIS IS AN EXAMPLE");
- цифровая константа. Например, (DIS, 100).

Примеры

(DIS, "C1=",C1) - выводит на экран координаты центра и радиус окружности.
(DIS, "l2=",l2) - выводит на экран расстояние между исходной точкой и прямой линией и угол, образованный прямой линией и абсциссой оси.

2.16.2 Выдержка времени - DLY

Оператор DLY позволяет программировать выдержку времени с определённой продолжительностью.

Формат:

(DLY , время) ,

где:

время - выражает время остановки в секундах (максимум 32 с). Может быть выражено 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)
```

2.16.3 Вращение моторезированного инструмента - USS

Эта команда позволяет вращать моторезированный инструмент, установленный в револьверную головку при помощи принудительного форсирования аналогового выхода своего преобразователя в то время, когда операнд S продолжает указывать запрограммированные величины главного шпинделя.

Допустимый формат:

(USS, ИМЯ ОСИ, [+-i]) ,

где:

ИМЯ ОСИ - является именем оси, конфигурированной в качестве шпинделя;
i - скорость вращения (об./мин). Знак указывает направление вращения (+ = по часовой стрелке). Если **i** не определена или равна нулю, то ось не будет вращаться; **i** может быть как цифровой константой, так и параметром E типа LR.

Примеры

(USS, S500) - вращение по часовой стрелке;
(USS, S0) - нет вращения;
(USS, S-500) - вращение против часовой стрелки.

2.17 Управление графическим дисплеем

Этот класс команд позволяет управлять графическим дисплеем из управляющей программы.

Имеются следующие команды:

- **UCG** - определить поле графического дисплея;
- **CLG** - стереть графический дисплей;
- **DCG** - блокировать графический дисплей.

2.17.1 Определение поля графического дисплея - UCG

Эта команда инициализирует графический дисплей и устанавливает пределы, масштаб и режим дисплея.

Формат:

(UCG, n, ОСЬ1I ОСЬ1S, ОСЬ2I ОСЬ2S[,ОСЬ3]) – плоская графика движения инструемента,

(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 (апликаты), для обозначения на экране точки, 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 объёмных 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, Z100 Z150, X50 X250) - Активизирует графическую видеостраницу #6 режима «**УПРАВЛЕНИЕ СТАНКОМ**». Графическая видеостраница показывает перемещение между Z100 и Z150 и между X50 и X250 относительно текущей исходной 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 исходные точки. Инструкция **UCG** должна быть программирована после исходных точек и перед разрешением коррекций инструмента.

Пример
```
N1 (UOT, 0, CZ200)
N2 (UCG, 1, Z-100 Z20, X30 X130)
N3 T1.1 M6
```

2.17.2 Сброс графического дисплея - CLG

Оператор CLG осуществляет сброс текущего профиля с экрана дисплея, оставляя на экране лишь декартовые оси.
Формат:

(CLG) .

2.17.3 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)
       ..........
```

2.18 Управление коррекциями инструмента - RQU

Эта команда позволяет управлять коррекциями инструмента из управляющей программы. Команда RQU переатестирует (изменяет и модифицирует) определенную коррекцию инструмента в соответствии с программируемыми значениями.
Формат:

(RQU, N.инстр., N.корр.,Z.., X..) ,

где

N.инстр - номер инструмента является цифровой константой или параметром E типа IN;

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 длины, который прибавляется к корректору оси Z;

X - опредлеляет инкремент длины, который прибавляется к корректору оси X. При программировании нуля, ничего не прибавляется к текущему значению длины корректора оси X или Z.

Пример
(RQU, 10, 1, ZE40; XE41) - модифицирует инструмент 10, корректор 1. Инкремент для оси Z содержится в E40. Инкремент для оси X содержится в E41.

Если файл корректоров создан для управления текущим (С) и максимальным значениями корректора, команда RQU изменяет текущее значение. Если же вы хотите изменить текущее значение, программируйте RQP с тем же форматом.

Если вы изменяете коррекцию для диаметральной оси, то управление разделит вами определяемое значение на два, перед прибавлением их к корректору.

Если инкрементальное значение коррекции не объявлено в цикле измерения, команда RQU требует индикатор синхронизации (#).

2.19 Определение параметров измерения - DPT

Команда **DPT** позволяет определить параметры измерения с пульта или из программы. Параметры, которые необходимо определить, следующие:

- размер подхода **Qa** (мм);
- размер безопасности **Qs** (мм);
- скорость измерения **Vm** (мм/мин).

Формат:

(DPT, Qa, Qs, Vm) - из программы;
DPT, Qa, Qs, Vm - с пульта.

Пример
DPT, 10, 12, 1000 - с пульта

Когда управление выполняет цикл измерения, тогда оно выполняет следующую последовательность движений (рисунок А.105):

- движение на быстром ходу к точке подхода (PS);
- движение на скорости измерения до точки измерения или расстояния безопасности, затем хранение размеров;
- возврат на быстром ходу до точки начала цикла измерения.

Примечание - Если щуп не переключается до достижения точки безопасности, щуп возвращается к начальной точке измерений, и выдаётся следующее сообщение: «Нет касания щупа при измерении».

2.20 Управление стойкостью инструмента - TOF

Команда **TOF** позволяет объявлять инструмент негодным. Команда **TOF** не может быть активизирована при неприсоединённых осях и запомненном поиске.

Формат:

(TOF, n) ,

где:

n - номер инструмента, который объявляется негодным; цифровая константа или параметр E (E10-E19).

Пример
(TOF, 22)
(TOF, E11)

2.21 Примеры циклов измерения

2.21.1 Измерение при черновой обработке

В этом примере разница между теоретическим и фактическим значениями нуля детали может быть + 5 мм.

Пример (рисунок А.106):
N21 T1.1 M6

```
N22 (DPT, 6, 6, 300)
N23 G X0 Z10
N24 G74 Z0 E30
N25 E31=ABS(E30)
N26 (BGT, E31, 5, K0)
N27 E30=NEG(E30)
N28 (UOT, 0, ZE30)
N29 T2.2 M6
-------------------
N44 T3.3 M6
-------------------
"KO" (DIS, "PIECE OUT OF TOLERANCE")
-------------------
-------------------
```

2.21.2 Пример применения цикла измерения при коррекции инструмента

Пример (рисунок А.107):

```
N1 T1.1 M6
N2 G X70 Z-20
N3 G74 X50 E30
N4 (RQU, 4, 4, XE30)
N5--------------------
----------------------
```

Пример проверки отклонения перед коррекцией:

```
N1 T1.1 M6
N2 G X70 Z-20
N3 G74 X50 E30
N4 E31=ABS(E30)
N5 (BLT, E31, 0.04, OK)
N6 (TOF, 4)
N7 (BNC, K0)
"OK"
N9 (RQU, 4, 4, XE30)
--------------------
--------------------
"K0" (DIS, "TOOL OUT OF USE")
--------------------
```

2.22 Координированная ось шпинделя

При программировании функции М, предусмотренной со стороны станочной логики, например, М21, система управляет осью шпинделя как осью, координированной с осями X и Z.

Ось шпинделя обычно идентифицируется функцией С, выраженной в градусах, и может быть программирована в системе абсолютных величин или приращений в диапазоне с + 0.0001 до +9999.9999. Движение осей может осуществляться на рабочей скорости или на быстром ходу при линейной и круговой интерполяциях. Скорость обработки F выражается в градус/минута и должна быть программирована с функцией G94.

Функции S (должны быть программированы с G97), M3, M4 и M5 будут приписаны моторизированному инструменту, оборудованному на револьверной головке.

Для деактивизации М21 необходимо использовать кнопку [«Сброс»] или другую функцию М, предусмотренную со стороны логики станка; например, функцию М20.

Функции разрешения/запрещения шпинделя должны быть программированы в отдельном кадре при присутствии функции G0.

Пример

```
N24 M21
N25 G97 S2000 M3 M8
N26 G0 G90 X100 Z-100
N27 G1 Z-105 G94 F100
N28 C180 F150
N29 G Z
N30 M20
---------------------
```

Комбинируя движения по X, Z и C осям, можно получить спирали и цилиндрические или конические винтовые линии.

Так как скорость обработки программируется в град./мин, необходимо вычислить её значение в соответствии с диаметром и скоростью в мм/мин, обеспечиваемой на детали.

Для выполнения цикла фрезерования вдоль окружности диаметром D необходимо вычислять угловую скорость по следующей формуле:

$$F = \frac{AV}{D} \cdot \frac{360}{\Pi} = AV \cdot \frac{114{,}64}{D}, \qquad (2.5)$$

где

F - угловая скорость, град/мин;
AV - скорость вдоль окружности, мм/мин;
D - диаметр окружности, мм.

Если ось вращения C двигается вместе с продольной осью W, получается цилиндрическая винтовая линия. В этом случае необходимо вычислить угловую скорость по следующей формуле:

$$F = \frac{AV \cdot \sqrt{dZ2 + dC2}}{\sqrt{dZ2 + (\Pi 0/360 \cdot dC)2}}, \qquad (2.6)$$

где:

F - угловая скорость, град/мин;
AV - скорость вдоль винтовой линии, мин/мин;
DZ - действительное перемещение по оси Z, мм;
DC - действительное перемещение вдоль оси C, градусы.

2.23 Виртуальные оси

Для обработки профилей на плоскости или на цилиндре, при помощи оси вращения и линейной оси, вводится понятие виртуальных осей.

Имеются следующие разновидности:

- **способ 1:** при исполнении профиля на плоскости, является возможным преобразование декартовых координат в полярные координаты. Линейная ось перпендикулярна оси вращения;
- **способ 2:** при исполнении профиля на цилиндре, является возможным преобразование декартовых координат в цилиндрические координаты. Линейная ось является параллельной оси вращения.

Если одна из этих разновидностей активизирована, то 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.

2.23.1 Программирование первым способом

Способ 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. 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 следующую формулу:

$$r = \frac{F}{VCmax} * \frac{360}{2П}, \qquad (2.7)$$

где

- **r** - минимальный радиус;
- **F** - скорость подачи, мм/мин;
- **VCmax** - скорость быстрого хода для оси вращения.

2.23.1.1 Пример программирования первым способом с применением GTL

Иллюстрация примера приведена на рисунках А.108-А.109.

```
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
```

2.23.1.2 Пример программирования первым способом с применением ISO

Пример приведён на рисунке А.109:

```
% : V1 ISO
; 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
%
```

Примечание - В примере минимальный радиус был вычислен при помощи формулы 7:

$$r = \frac{F \times 180}{3{,}14 \times VCmax}$$

2.23.2 Программирование вторым способом

Этим способом можно 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 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 (рисунки А.110-А.112).

```
N1   ("DIS", "EXAMPLE")
N2   T1.1 M6
N3   GX120 Z-20
N4   M21
N5   G0 G94 C0 S.. M..
N6   E30=60
N7   (UAV, 2, C, V, E30)
N8   (DPI, Z, V)
N9   p1=Z-20 V0
N10  E31=2x3.1415xE30
N11  p2=Z-20 VE31
N12  l1=p1, p2
N13  c1=I-45 J80 r25
N14  c2=I-35 J140.71 r-25
N15  c3=I-35 J180 r-25
N16  l2=C1, C2
N17  l3=C2, C3
N18  c4=C3, l1, r15
N19  G21 G41 p1 F500
N20  l1
N21  c1
N22  l2
N23  c2
N24  l3
N25  c3
N26  c4
N27  l1
N28  G20 G40 p2
N29  (UAV, 0)
N30  (DPI, Z, X)
N31  G X 130
----------------
----------------
N99  M30
```

2.24 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 имеет вид:

(UAV, 3, имя ведомой, имя ведущей, соответствие ведущая-ведомая, зерк.)

Пример
(UAV, 3, VU, XZ, 12, 21),

где:

3	- режим параллельных осей;
VU	- ведомые оси (1 до 7 символов);
XZ	- ведущие оси (1 до 4 символов);
12	- цифровой ряд, определяющий соответствие между осями. Значение цифры определяет ведущую ось, а его позиция - ведомую. В примере X является ведущей для V, Z для U;
21	- цифровой ряд, который характеризует тип обработки:
1	- нормальная обработка;
2	- зеркальная обработка.

Относительная позиция цифры определяет ведомую ось. В примере ось V обрабатывается 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**).

2.25 Косоугольная система координат

Для реализации данной функции в системе должны быть определены две виртуальные оси. Программирование обрабатываемого профиля выполняется с помощью виртуальных осей, образующих декартовую систему координат.

Угол косоугольной системы координат определяется с глобальной переменная UGF. Если угол фиксированный и неизменный, то установка величины угла может быть выполнена с инструкцией ASS (секция 4 файла PGCFIL, ASS=UGF,30), либо в режиме «УПРАВЛЕНИЕ СТАНКОМ» командой или кадром, например: UGF=30.

Угол косоугольной системы координат (A) задаётся в градусах и долях градуса и откладывается от положительного направления линейной координатной поперечной оси к положительному направлению линейной координатной продольной оси. Пример косоугольных осей для токарного станка представлен на рисунке 25.1.

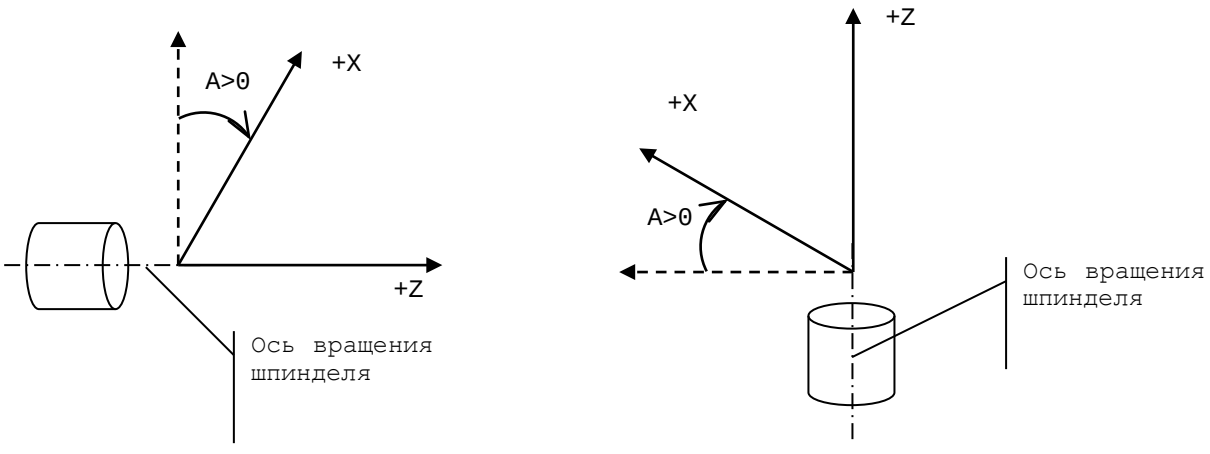


Рисунок 25.1 - Пример косоугольных осей для токарного станка

Для начала программирования в декартовой системе виртуальных координат на станке с косоугольными осями необходимо определить трёхбуквенный код **UAV,5 ,** порядок имён в котором устанавливает соответствие между двумя реальными косоугольными осями (угол определён в переменной **UGF**) и двумя виртуальными осями декартовой системы координат.

Формат

(UAV,5,ось1ось2,ось3ось4,параметр) ,

где:

ось 1 - имя реальной оси 1;
ось 2 - имя реальной оси 2;
ось 3 - имя виртуальной оси 1;
ось 4 - имя виртуальной оси 2;
параметр - незначащий числовой параметр; при задании он игнорируется.

Окончание программирования в декартовой системе координат виртуальных осей:

(UAV,0) - отмена программирования в виртуальных осях.

Перед первым использованием виртуальных осей необходимо определить плоскость интерполяции трехбуквенным кодом DPI, например:

(DPI, U, V) ,

где:

U – абсцисса;
V - ордината.

Программирование между заданием (UAV,5, ZX, UV,) и отменой (UAV,0) должно выполняться только в виртуальных осях.

Пример

```
GXZ
(DPI,U,V)
(UAV,5, ZX, UV,)
(UCG,2,U-50U30,V-20V20)
G0 U20 V
G1 G41 U10 F500
G3 U-10 V0 I0 J0
G1 G40 U-20
GV20
U20
(UAV,0)
```

2.26 Коды для программирования станочных циклов

В эту группу входят следующие макрокоманды:

1) **TGL** - цикл нарезания пазов;
2) **FIL** - цикл нарезания резьбы;
3) **DFP** - определение профиля;
4) **SPA** - осепараллельная черновая обработка без чистовой обработки;
5) **SPF** - осепараллельная черновая обработка с предварительной чистовой обработкой;
6) **SPP** - черновая обработка параллельно профилю;
7) **CLP** - чистовая обработка профиля.

2.26.1 Цикл нарезания пазов - TGL

Этот цикл создает или внешние или внутренние пазы, параллельные осям X или Z. Чтобы получить паз, параллельный оси Z, используется формат:

(TGL, Z величина, X величина, K величина) ,

где:

Z - конечный размер паза;

X - внутренний диаметр;
K - ширина инструмента.

Кадру с командой **TGL** должен предшествовать кадр с перемещением типа G0/G1 в начальной точке цикла.
Пример паза, параллельного оси Z, приведен на рисунке А.113.

Чтобы программировать паз, параллельный оси X, необходимо использовать формат:

(TGL, X величина, Z величина, K величина) ,

где
X - конечный размер паза;
Z - внутренний размер паза;
K - ширина инструмента.

Пример паза, параллельного оси X, изображен на рисунке А.114.

Устройство управления автоматически устанавливает остановку в конце паза. Длительность останова определяется кодом TMR. Чтобы заблокировать автоматическую остановку, программируется TMR=0 перед началом цикла обработки паза.

Пример внутреннего паза приведен на рисунке А.115.

В конце паза инструмент возвращается в стартовую точку, определяемую в предыдущем кадре.

2.26.2 Цикл нарезания резьбы - FIL

2.26.2.1 Цикл нарезания резьбы – FIL (первый вариант)

Цикл нарезания резьбы позволяет вам программировать в одном кадре цилиндрическую или коническую многопроходную резьбу. Допустимый формат:

(FIL, Z.., X.., K.., L.., R.., T.., P.., a.., b..),

где:
Z - конечный размер Z;
X.. - конечный размер X;
K.. - шаг;
L.. - число проходов черновой и чистовой обработки, т.е. L11.2;
R.. - расстояние между инструментом и деталью (по умолчанию, R=1);
T.. - 3-х цифровой код, определяющий тип нарезания резьбы (по умолчанию, T000):

Цифра 1:
- 0 = нарезание с конечным пазом;
- 1 = нарезание без конечного паза.

Цифра 2:
- 0 = внешнее нарезание резьбы;
- 1 = внутреннее нарезание резьбы.

Цифра 3:
- **0** = метрическое нарезание резьбы;
- **1** = дюймовая резьба;
- **2** = нестандартное нарезание резьбы с глубиной и углом, определяемыми параметрами a и b;
- **P.** - число принципов (по умолчанию P=1);
- **a.** - угол резьбы, только для нестандартной;
- **b.** - глубина резьбы.

Устройство управления автоматически вычисляет позиции, скользя вдоль края резьбы, так что часть результирующей стружки остается постоянной. Для резьб с несколькими законами вы должны только определить шаг каждого витка. Устройство управления выполняет каждый проход для каждого закона перед выполнением последу-

ющего прохода. Каждый закон выполняется без перемещения начальной точки каждой резьбы, но на расстоянии от углового нуля шпинделя.

Для резьб с конечным пазом вы должны программировать теоретический конечный Z, т.к. фиксированный цикл обеспечивает увеличение хода, равное половине шага.

В резьбах без конечного паза инструмент достигает программируемого размера и затем перемещается обратно с конической резьбой вдоль обратного диаметра. Резьба без конечного паза не может быть получена в кадровом режиме.

Пример цикла нарезания резьбы приводится на рисунке А.10.

2.26.2.2 Цикл нарезания резьбы - FIL (второй вариант)

Данный вариант существует параллельно с первым и введён во все версии ПрО, начиная с версий 1.41.31Р, 2.31Р, 2.31РИВ, 3.31Р и в версиях 4.ХХР.

Цикл нарезания резьбы позволяет программировать в одном кадре цилиндрическую или коническую многопроходную резьбу.

формат:

(FIL, Z.., X.., K+.., L.., R.., T.., P.., a.., b..),

где:

Z.. - конечный размер Z;
X.. - конечный размер X.

Порядок расстановки имён осей определяет ось, вдоль которой выполняется резьба и задан шаг резьбы:

Z..X..- вдоль оси Z;
X..Z..- вдоль оси X.

K+.. - шаг резьбы. Величина шага резьбы имеет знак «+» или «-».

Знак величины шага определяет ось, вдоль которой выполняется резьба:

- «+» - вдоль оси абсцисс;
- «-» - вдоль оси ординат.

В случае конической резьбы знак для шага устанавливается в зависимости от величины перемещения по осям, определяющим конус:

- «+» - перемещение больше вдоль оси абсцисс;
- «-» - перемещение больше вдоль оси ординат.

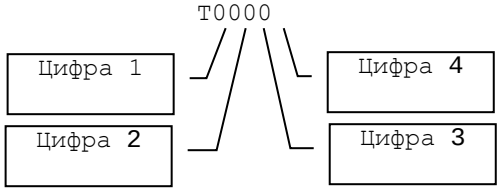
Примечания

1. Угол наклона конической резьбы, откладываемый от оси, вдоль которой задан шаг, не может превышать 45^0, см. рисунок А.9.
2. Ошибочное задание знака для величины шага резьбы будет сопровождаться при выполнении программы ошибкой «НЕКОНГРУЭНТНЫЙ ПРОФИЛЬ».

L.. - число проходов черновой и чистовой обработки, т.е. L11.2;

R.. - расстояние между инструментом и деталью (по умолчанию R=1);

T.. - 4-х цифровой код, определяющий тип нарезания резьбы (по умолчанию - T0000);



Цифры 1 и 2 (цифра 1 может быть опущена или равна 0):

00: - нарезание с конечным пазом,
- врезание под углом,
- без торможения в конце резьбы.

01: - нарезание без конечного паза,
- врезание под углом,
- без торможения в конце резьбы.

10: - нарезание с конечным пазом,
- врезание радиально,
- без торможения в конце резьбы.

11: - нарезание без конечного паза,
- врезание радиально,
- без торможения в конце резьбы.

12: - нарезание с конечным пазом,
- врезание под углом,
- останов в конце резьбы по функции G09.

13: - не рекомендуется:
- останов в конце резьбы по функции G09,
- врезание под углом,
- нарезание без конечного паза.

14: - нарезание с конечным пазом,
- врезание радиально,
- останов в конце резьбы по функции G09.

15: - не рекомендуется,
- останов в конце резьбы по функции G09,
- врезание радиально,
- нарезание без конечного паза.

Цифра 3:
0: - внешнее нарезание резьбы;
1: - внутреннее нарезание резьбы.

Цифра 4:
0: - метрическое нарезание резьбы;
1: - дюймовая резьба;
2: - нестандартное нарезание резьбы с глубиной и углом, определяемыми параметрами **«a»** и **«b».**

P.. - число заходов (по умолчанию P=1);

a.. - угол резьбы (только для нестандартной резьбы);

b.. - глубина резьбы.

Устройство управления автоматически вычисляет позиции, скользя вдоль края резьбы так, что часть результирующей стружки остаётся постоянной. Для резьбы с несколькими заходами вы должны только определить шаг каждого витка. Устройство управления выполняет каждый проход для каждого захода перед выполнением последующего прохода. Каждый заход выполняется без перемещения начальной точки каждой резьбы, но на расстоянии от углового нуля шпинделя.

Для резьбы с конечным пазом вы должны программировать теоретический конечный Z, т.к. фиксированный цикл обеспечивает увеличение хода, равное половине шага.

В резьбе без конечного паза инструмент достигает программируемого размера и затем перемещается обратно с конической резьбой вдоль обратного диаметра.

Резьба без конечного паза не может быть получена в кадровом режиме.

Пример цикла нарезания резьбы с врезанием под углом приводится на рисунке A.10.

Пример цикла нарезания резьбы с радиальным врезанием, конечным пазом и торможением по G09 в конце резьбы приводится на рисунке A.11.

2.26.3 Определение профиля - DFP

Этот код позволяет вам определить и сохранить до 8 профилей. Внутри каждого профиля вы можете определить до 16 элементов по стандартам ISO или GTL.

Запомненные профили могут вызываться из циклов черновой или чистовой обработки.

Примеры определения профиля по стандарту ISO:

```
(DFP, 1)          (DFP, 1)
G1 X40 Z          G21 p1
Z-20              l1
X60 Z-30          r-5
Z-50              c1
X80               l2
(EPF)             r-3
                  l3
                  G20 p2
                  (EPF)
```

При описании профиля следует помнить, что:

- по стандарту ISO все кадры профиля должны содержать контурные коды (G1, G2, G3). Код быстрого хода G0 может появляться только в первом кадре;
- хотя функции F могут программироваться внутри профиля, они только будут активизироваться во время цикла чистовой обработки профиля;
- DFP всегда должен предшествовать соответствующему циклу обработки;
- описываемые ошибки сигнализируются только во время выполнения цикла обработки;
- номер блока в цикле DFP будет воспроизводиться только во время выполнения цикла чистовой обработки (CLP). Во всех других циклах (черновая обработка параллельная оси X или Z и т.д.) ЭЛТ воспроизводит кадр, который содержит макрокоманду обращения к профилю, определенного с помощью DFP;
- для использования компенсации радиуса инструмента программируется G40/G41/G42 внутри цикла DFP. Например:

```
(DFP, 1)
G42 G1 X100 Z0
Z-50
--------------
--------------
G40 X200
(EPF)
```

Устройство управления автоматически использует компенсацию во время циклов черновой и чистовой обработки.

2.26.4 Осепараллельная черновая обработка без предварительной чистовой обработки - SPA

Чтобы запрограммировать черновую обработку параллельно оси X, используется следующий формат:

(SPA, X, n, L.., X.., Z..).

Чтобы запрограммировать черновую обработку параллельно оси Z, используется следующий формат:

(SPA, Z, n, L.., X.., Z..),

где:

n - номер профиля, ранее запомненного с DFP. Он обязателен и может изменяться от 1 до 8;
X - радиальный припуск по оси X;
Z - радиальный припуск по оси Z;
L - число черновых проходов. Может изменяться от 1 до 255.

X и Z можно пропустить. Если они присутствуют, то всегда должны иметь положительную величину.

На основе точки фиксации и направления профиля устройство управления автоматически решает, какая должна быть черновая обработка - внутренней или внешней и присваивает соответствующий знак припуску.

Точка фиксации должна быть внешней относительно поля черновой обработки, по крайней мере, на величину программируемого припуска. Если профиль не монотонный, т.е. если он включает в себя выемки, инструмент автоматически обходит выемки во время черновой обработки.

Пример черновой обработки, параллельной оси X (рисунок А.116):

```
(UCG,2,Z-90Z5,X-5X150)
p1=Z-76 X140
l1=p1, a-90
l2=Z0 X100, a0
l3=Z-60 X100, a-30
l4=Z-15 X30, a-60
p2=Z0 X30
l5=p2, a0
(DFP, 1)
G21 p1
l1
l2
r-18
l3
r-15
l4
l5
G20 p2
(EPF)
T1.1 M6 S… F…
GX143 Z1.5
(SPA, X, 1, L12, X1, Z1)
```

Пример черновой обработки, параллельной оси Z (рисунок А.117):

```
(UCG,2,Z-100Z5,X-5X150)
p1=Z0 X30
l1=p1, a180
l2=Z-15 X30, a120
l3=Z-60 X100, a150
l4=Z-75 X100, a180
p2=Z-75 X140
l5=p2, a90
(DFP, 1)
G21 p1
l1
l2
r15
l3
r18
l4
l5
G20 p2
(EPF)
T1.1 M6 S… F…
GX143 Z1.5
(SPA, Z, 1, L9, X1, Z1)
```

2.26.5 Осепараллельная черновая обработка с предварительной чистовой обработкой - SPF

Для программирования черновой обработки, параллельной оси X с конечной обработкой вдоль профиля, используется формат:

(SPF, X, n, L.., X.., Z..).

Для программирования черновой обработки параллельной оси Z, используется формат:

(SPF, Z, n, L.., X.., Z..).

Эти параметры определены раньше. Профиль должен быть однородным. Иначе будет воспроизводиться сообщение ошибки.

2.26.5.1 Пример использования кода SPF

Пример использования кода SPF приведён на рисунке А.118:

```
(UCG,2,Z-90Z5,X-5X150)
p1=Z0 X30
l1=p1, a180
l2=Z-15 X30, a120
l3=Z-60 X100, a150
l4=Z-75 X100, a180
p2=Z-75 X140
l5=p2, a90
(DFP, 1)
G21 p1
l1
l2
r15
l3
r18
l4
l5
G20 p2
(EPF)
T1.1 M6 S… F…
GX143 Z1.5
(SPF, Z, 1, L9, X1, Z1)
(CLP, 1)
G G79 X Z M30
```

2.26.5.2 Пример внутренней осепараллельной черновой обработки с предварительной чистовой обработкой

Пример внутренней осепараллельной черновой обработки с предварительной чистовой обработкой приведён на рисунке А.119.

```
(UCG,2,Z-60Z5,X-5X80)
p1=Z-47 X20
l1=p1, a90
l2=Z0 X40, a0
l3=Z-20 X68, a45
p2=Z0 X68
l4=p2, a0
(DFP, 1)
G21 p1
l1
l2
l3
l4
G20 p2
(EPF)
T1.1 S… F…
G X15 Z2.5
(SPF, X, 1, L10, X2, Z2)
```

2.26.5.3 Пример немонотонного профиля с черновой обработкой

Пример немонотонного профиля с черновой обработкой приведён на рисунке А.120.

```
(UCG,2,Z-100Z5,X-5X100)
p1=Z0 X62
l1=p1, a135
l2=Z0 X68, a180
l3=Z-15 X0, a90
l4=Z0 X76, a180
l5=Z-38 X76, a198
l6=Z-77 X76, a105
p2=Z-97 X86
l7=p2, a135
(DFP, 1)
G21 p1
l1
l2
```

```
l3
r2
l4
r10
l5
r-8
l6
r4
l4
l7
G20 p2
(EPF)
T1.1 S… F… M3
G X90 Z2
(SPA, Z, 1, L7, X1, Z1)
```

В вышеуказанном примере программируется код SPA. Устройство управления выполняет цикл черновой обработки и пропускает карман. Оставшиеся осколки могут быть удалены стандартными движениями. Затем программируется код CLP для чистовой обработки профиля.

Направление профиля должно совпадать с направлением инструмента в цикле черновой обработки. На рисунке А.121 приведен пример направления профиля и инструмента.

2.26.6 Черновая обработка параллельно профилю - SPP

Этот код позволяет вам выполнять цикл черновой обработки деталей со средним припуском. Допустимый формат следующий:

(SPP, n, L, X1. X2., Z1. Z2.) ,

где:

N	- номер профиля;
L	- число проходов;
X1	- X припуск, оставленный на чисто обработанной детали;
X2	- X припуск на необработанной детали;
Z1	- Z припуск, оставленный на чисто обработанной детали;
Z2	- Z припуск на необработанной детали;
X1 и Z1	- обязательны, даже если их величина равна нулю.

Точка зажима описана в предыдущем разделе (**SPA - SPF**).

Пример использования SPP (рисунок А.122):

```
(UCG,2,Z-100Z5,X-5X150)
p1=Z0 X30
l1=p1, a180
l2=Z-15 X30, a120
l3=Z-60 X100, a150
l4=Z X100, a180
p2=Z-75 X140
l5=p2,a90
(DFP, 1)
G21 p1
l1
l2
r15
l3
r18
l4
l5
G20 p2
(EPF)
T1.1 S… F… M3
GX143 Z1.5
(SPP, 1, L4, Z1 Z10, X1 X10)
```

Пример (рисунок А.123):

```
(UCG,2,Z-100Z5,X0X150)
p1=Z0 X60
l1=p1, a180
l2=Z-30 X60, Z-50 X40
l3=Z0 X40, a180
l4=Z-70 X40, Z-90 X60
p2=Z-110 X60
l5=p2, a180
(DFP, 1)
G21 p1
l1
r10
l2
r-10
l3
r-10
l4
r10
l5
G20 p2
(EPF)
T1.1 S… F…
G X84 Z1
(SPP, 1, L4, X X10)
```

2.26.7 Чистовая обработка профиля - CLP

Для программирования чистовой обработки профиля используется формат:

(CLP, n) ,

где:

N - имя профиля, ранее определенного с DFP;

CLP - это единственный цикл обработки, во время которого могут активизироваться функции F, программируемые внутри DFP.

Пример осепараллельной обработки с предварительной чистовой обработкой и черновой обработкой (рисунок А.124):

```
(UCG,2,Z-90Z5,X-5X155)
p1=Z0 X30
l1=p1, a180
l2=Z-15 X30, a120
l3=Z-60 X100, a150
l4=Z-75 X100, a180
p2=Z-75 X140
l5=p2, a90
(DFP, 1)
G21 p1
l1
l2
r15
l3
r18
l4
l5
G20 p2
(EPF)
T1.1 M6 S2000 F500
GX143 Z1.5
(SPF, Z, 1, L9, X1, Z1)
X300 Z200
T2.2 M6 S… F…
GX30 Z2
(CLP, 1)
GX300 Z200
```

2.27 Синхронные кадры

Синхронными кадрами являются те кадры, которые системой выполняются только в определенные моменты или при определенных условиях. Для включения/выключения синхронизации используют следующие символы:

1) # - включает синхронизацию;
2) & - выключает синхронизацию.

Эти символы программируются после номера кадра и перед инструкцией.

Примечание - По умолчанию система выполняет команды без синхронизации. Однако синхронизация присутствует по умолчанию для кадров, содержащих переменные SA и SK.

2.27.1 Включение синхронизации

При программировании символа # перед кадром определенные оси перемещаются после выполнения вычислений (1 кадр в режиме из точки в точку, n кадров в непрерывном режиме). Синхронизация применяется для программирования команд, которые зависят от результатов вычислений или которые присваивают значение переменной в конце программированного перемещения.

Пример

```
N9  GZ100 X80
N10 #TIM1=TIM0    -    принимает время часов системы при окончании
--------------       движения осей, запрограммированного в кадре N9.
--------------
--------------

N29 GZX
N30 #(USG,2,Z-50 Z100,X-20,X80)  -  определяет графическое поле при
--------------                      окончании движения GZX.
--------------
--------------

N50 GZ200
N51 #(BEQ,SA126,1,LAB)   -   переходит к метке LAB, если бит 126 из
--------------               буфера SA установлен в 1, после
--------------               перемещения оси X на 200.
--------------
N59 GZ50
N60 #(DLY,10)   -   осуществляет остановку в 10 сек. в конце движения
--------------      оси GZ50.
--------------
--------------

N87 E30=0.2
N88 #(RQU,1,1,ZE30)  -  модификация инструмента, когда E30 достигает
--------------          значения 0.2 в кадре N88 и 0.3 в кадре N95.
--------------
N94 E30=0.3
N95 #(RQU,1,1,ZE30)
```

2.27.2 Выключение синхронизации

Необходимо использовать символ **&** для выключения синхронизации между вычислениями и движением осей.

3 ТРЁХБУКВЕННЫЕ КОДЫ, ИСПОЛЬЗУЕМЫЕ ПРИ ПРОГРАММИРОВАНИИ СИСТЕМ NC-110, NC-200, NC-201, NC-201М, NC-202, NC-210, NC-220, NC-230 (ТОКАРНЫЙ ВАРИАНТ)

Трёхбуквенные коды, используемые при программировании систем, в зависимости от их функций могут быть разделены на пять групп:

1) коды, используемые в режиме «**КОМАНДА**» при управлении файлами (перечень кодов приведён в таблице 3.1);
2) коды периферийных устройств (перечень кодов приведён в таблице 3.2);
3) коды, используемые при управлении УП (перечень кодов приведён в таблице 3.3);
4) коды, используемые при управлении инструментом (перечень кодов приведён в таблице 3.4);
5) коды, используемые в кадрах управляющей программы (перечень кодов приведён в таблице 3.5).

Таблица 3.1 - Коды, используемые в режиме «**КОМАНДА**»

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 УП

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 илипеременных	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

Продолжение таблицы 3.3

Код	Формат	Функция
TMR	TMR=значение	Определяет время выдержки в конце движения при G04 или в фиксированных циклах (выражается в секундах или в количестве оборотов шпинделя)
SSL	SSL = величина	Определяет предельную скорость шпинделя
RTR	RTR=1 RTR=0	Дробление стружки разрешено Дробление стружки запрещено
SRT	SRT = значение	Определяет шаг дробления стружки
VRT	VRT = значение	Определяет скорость дробления стружки
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
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 вышеназванного режима
URL	URL=1 URL=0	Разрешение работы корректора рабочей подачи Отмена вышеназванного режима
USO	USO=1 USO=0	Подтверждение M01 Отмена M01
UCV	UCV=N	Определяет тип вывода на экран осевых значений для видеостраницы #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 вышеназванного режима
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
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, то 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 не указан операнд n, то команда удаляет все записи файла
VOL	VOL=1 VOL=0	Активизация штурвала Отключение штурвала
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, выраженная в 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 УП

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 введён 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.
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..,X..)	Переквалификация инструмента. NUT : номер инструмента; NCOR: номер корректора. Изменяет текущие корректоры и файл корректоров.
RQP	(RQP,NUT, NCOR,Z...,X.)	Изменяет корректоры Z и/или X, определенных в объявлении; файл корректоров не изменяется.
DPI	(DPI,ось S1, ось S2)	Определяет плоскость интерполяции; ось 1, ось 2: оси, имена которых определяет 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 защищённой 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

Продолжение таблицы 3.5

Код	Формат	Функция
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 (max=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,5,ZX,UV,)	Определяет виртуальные оси U и V
	(UAV,0)	Запрещает виртуальные оси.
DFP	(DFP,n)	Определяет номер профиля (1-8), который вызывается во время циклов черновой и чистовой обработки
EPF	(EPF)	Закрывает определение профиля
SPA	(SPA,a,n,l, x,z)	Цикл черновой обработки, параллельной к оси «a»: a: ось x или z; n: номер профиля; l: число проходов; x: припуск по x; z: припуск по z. SPA не может быть применена к немонотонным профилям.
SPF	(SPF,a,n,l,x,z)	Цикл черновой обработки, параллельной к оси «a» с предварительной чистовой обработкой: a: ось x или z; n: номер профиля; l: число проходов; x: припуск по x; z: припуск по z. SPF не может быть применена к немонотонным профилям.
SPP	(SPP,n,l,z1,z2,x1,x2)	Цикл черновой обработки, параллельной к профилю: z1: припуск по z; z2: первоначальный припуск по Z; x1: припуск по x; x2: первоначальный припуск по x.
CLP	(CLP,n)	Вызов цикла числовой обработки, n: номер профиля
TGL	(TGL,z..,x..,k..)	Цикл обработки паза параллельно к оси x или z: z: конечный размер паза; x: внутренний диаметр паза; k: ширина инструмента.
FIL	(FIL,z..,x..,k..,l..,r.., t..,p..,a..,b..,)	Цикл резьбонарезания.
USS	(USS,s+i)	Управляет моторизированным инструментом: s: ось, конфигурируемая в качестве шпинделя; i: число оборотов; знак указывает направление вращения.
EXE	(EXE,N,Имя УП/MPx)	N – номер процесса Имя выполняемой УП и устройство памяти MP
REL	(REL)	Деактивизация УП
WAI	(WAI,n) (WAI,Var=значение)	Ожидание синхронизирующего кода SND из процесса «n» Ожидание значения переменной
SND	(SND,n)	Синхронизирующий сигнал в процесс «n»

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 **#**.

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.

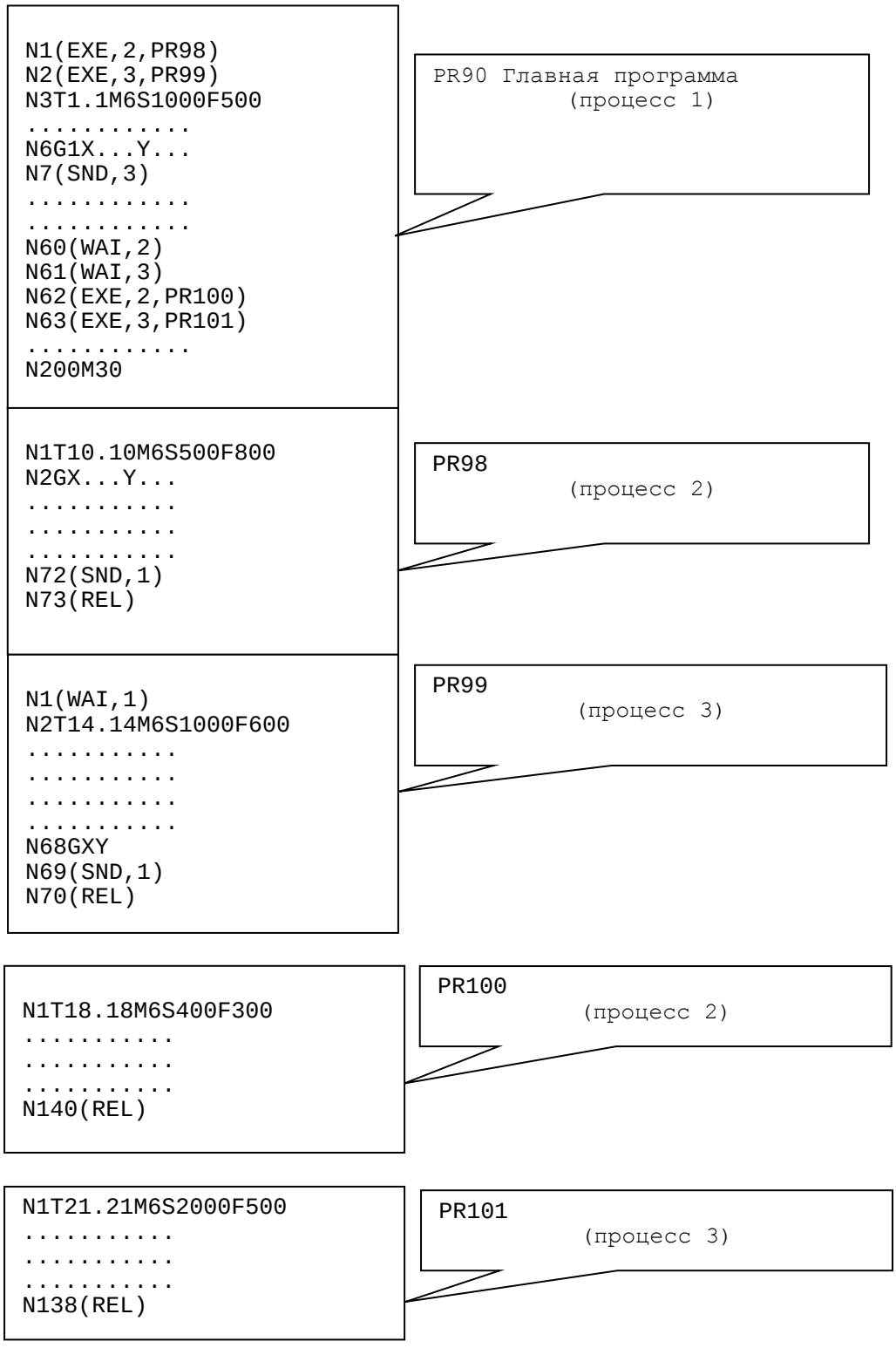
На кадрах N68-N86-N81 три процесса синхронизируются на повторном старте.

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.

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-ю 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, откры-
того на канале 1.
```

Содержание полей записи записывается в переменные, описанные в структуре два.

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 третье поле второго 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 уста-
новлено 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 определен, 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); создается 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 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 «А»:

код с «А» = код цвета фона + код цвета символа.

Таблица 5.4 - Порядок расположения цвета и код цвета по палитре **RGB**

Номер цвета	**R** крас-ный	**G** зелё-ный	**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 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 строкой 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:

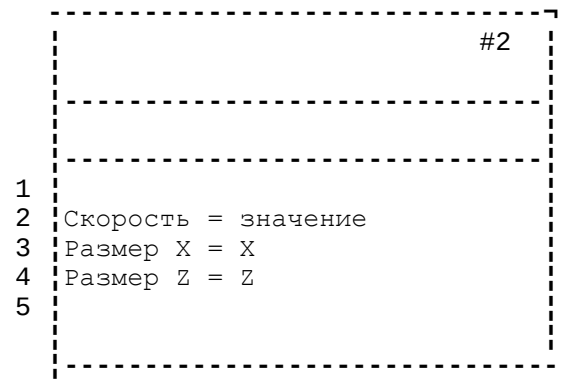


Рисунок 5.1

(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.

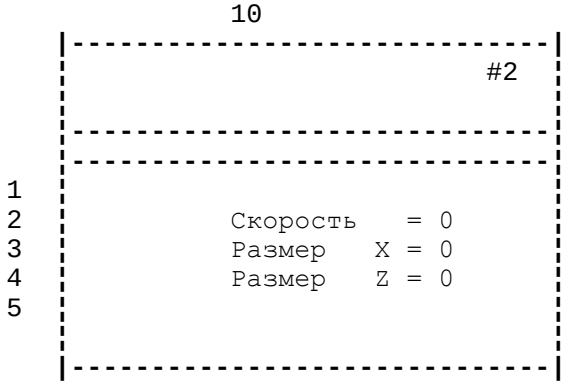


Рисунок 5.2

(INP, 1,E40, 2); ввод значения для поля 2 (X), связанный 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;
УП - управляющая программа;
УЧПУ - устройство числового программного управления;

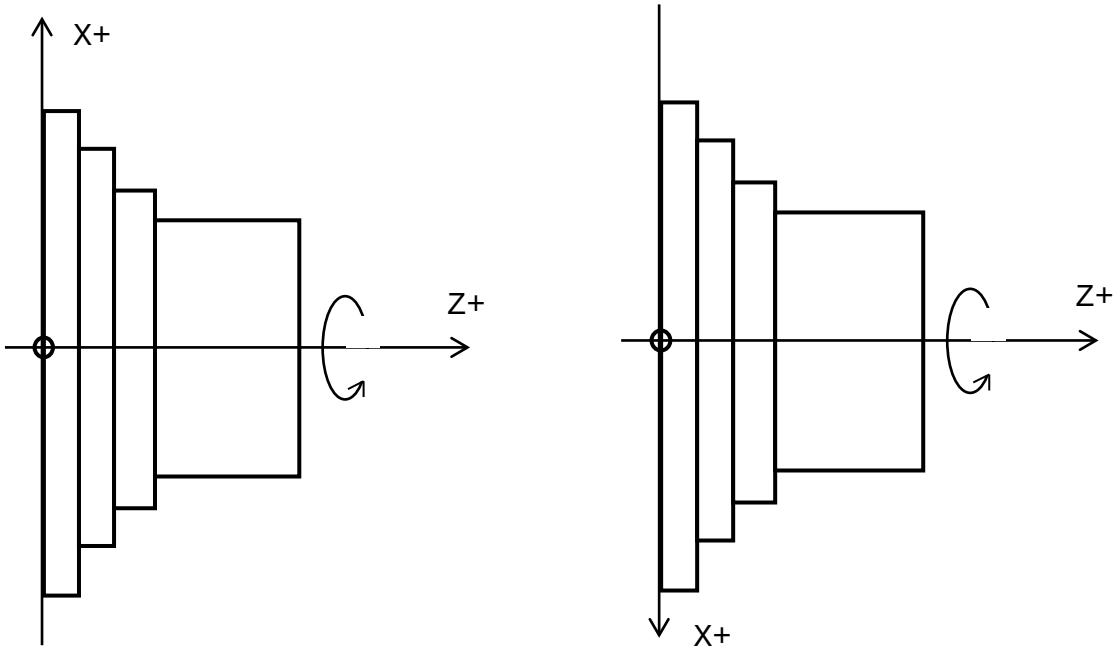
PLC – программируемый логический контроллер;

ПЕРЕЧЕНЬ ТАБЛИЦ

ПРИЛОЖЕНИЕ А

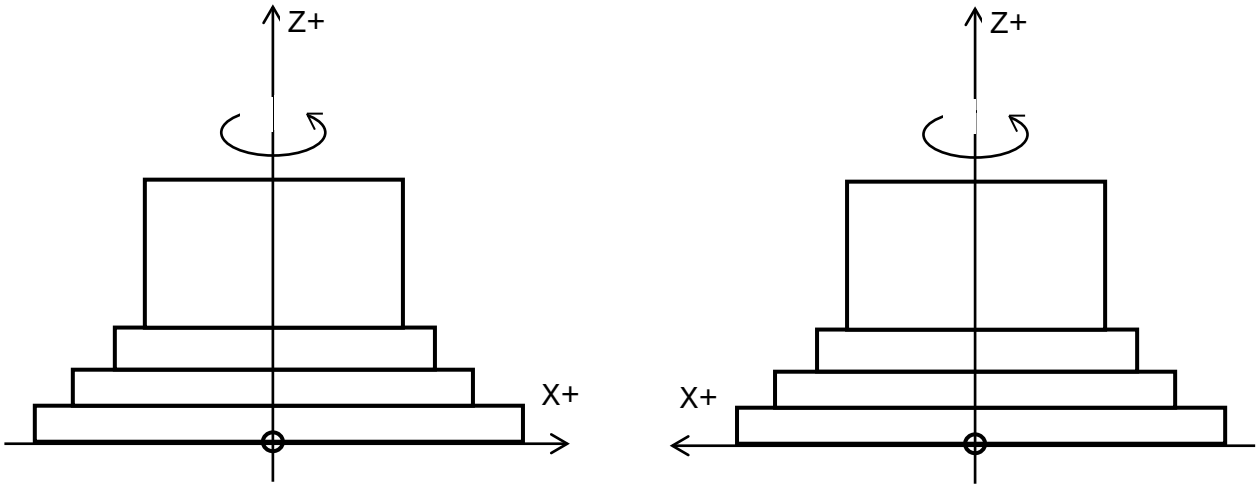
(обязательное)

ИЛЛЮСТРАЦИОННЫЙ МАТЕРИАЛ



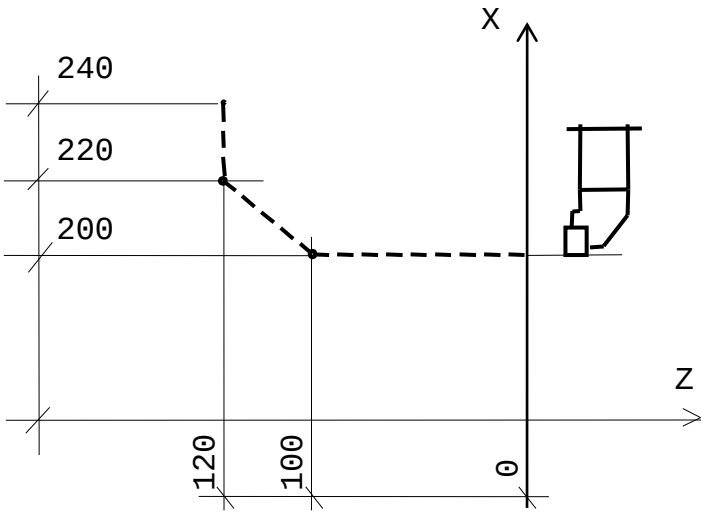
а) б)

Рисунок А.1 - Горизонтальное расположение осей станка



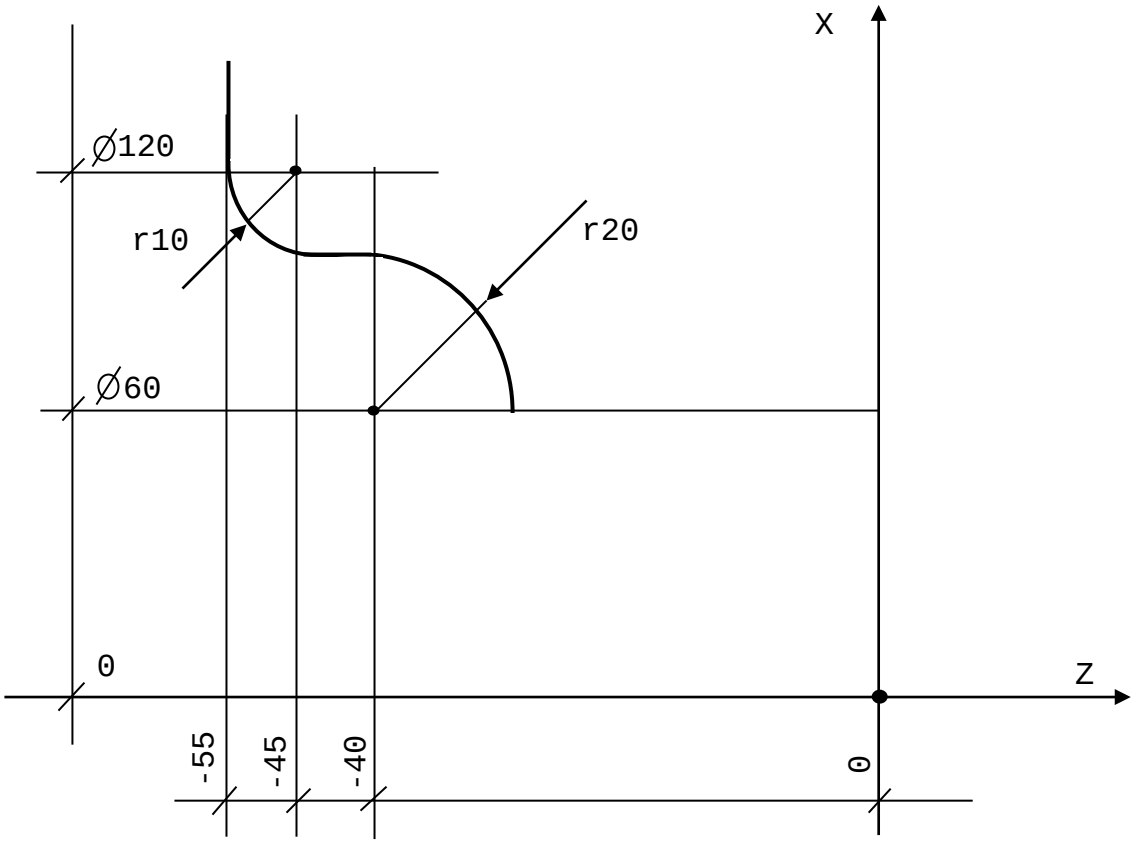
а) б)

Рисунок А.2 - Вертикальное расположение осей станка




```
N10 G1 Z-100 F0.3
      N20 X220 Z-120
      N30 X240
```

Рисунок А.3 - Пример линейной интерполяции G01




```
N10 G1 X60 Z-20
N20 G3 X100 Z-40 I-40 J60
N30 G1 Z-45
N40 G2 X120 Z-55I-45J120
N50 G1 X...
```

Рисунок А.4 - Пример круговой интерполяции (G02, G03)

```
1     G02 X20 Y20 R+20 F100
2     G02 X20 Y20 R-20 F100
```

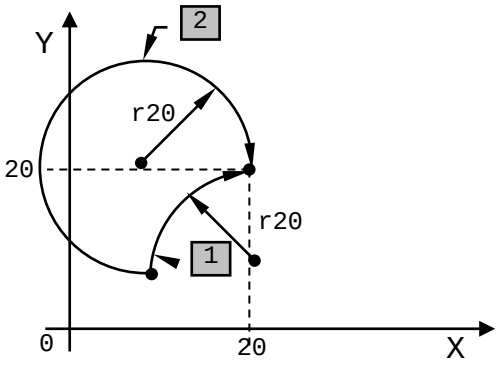
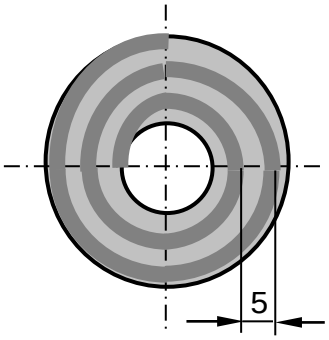


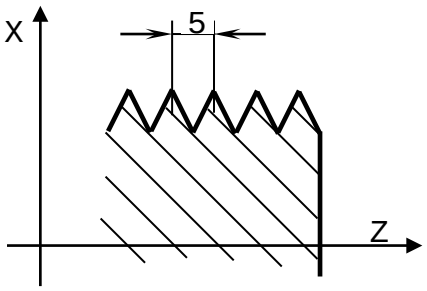
Рисунок А.5 – Пример программирования дуги менее 360 градусов

```
N15 G33 X0 K5
```



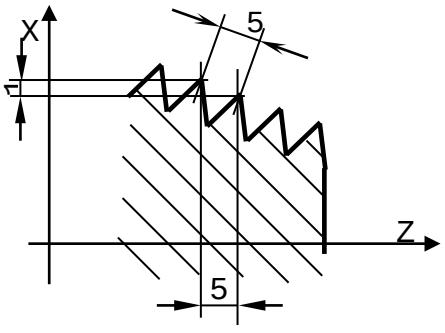
а) фронтальная резьба

```
N22 G33 Z-50 K5
```



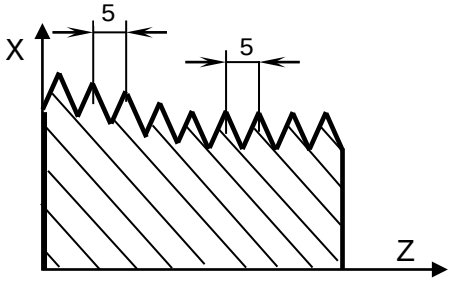
б) цилиндрическая резьба

```
N22 G33 X50 Z-50 K5.1
```



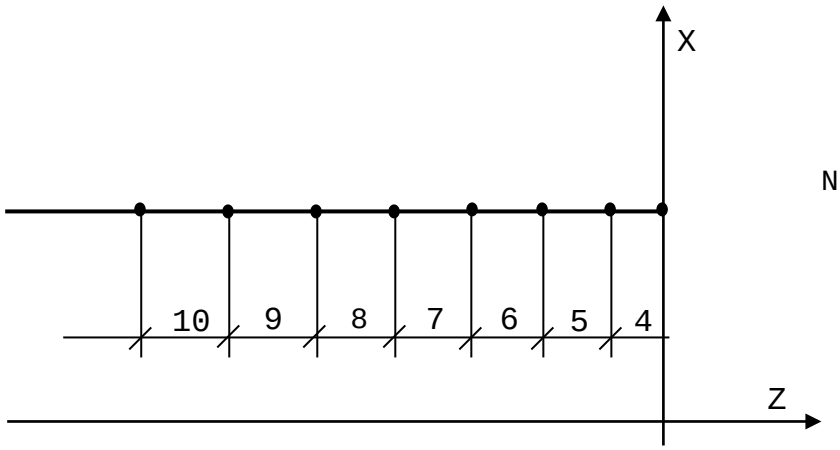
в) коническая резьба

```
N22 G34 Z-50 K5
N22 G34 X30 Z-50 K5
```



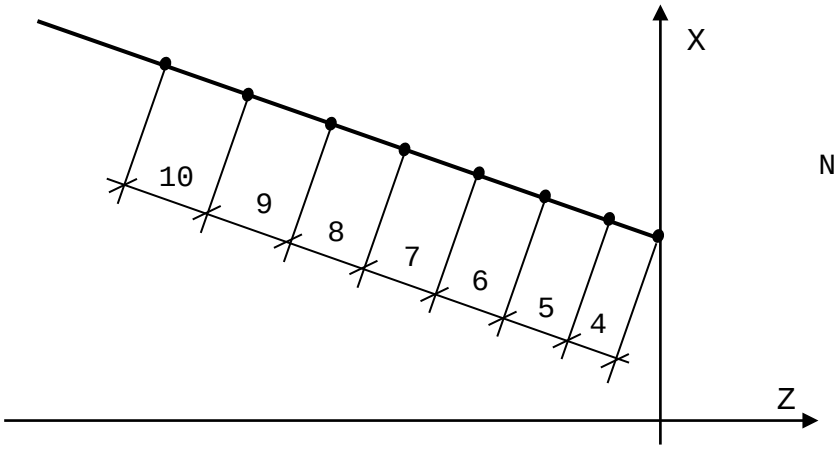
г) цилиндрическо-коническая резьба

Рисунок А.6 – Примеры нарезания резьбы с постоянным шагом (G33)



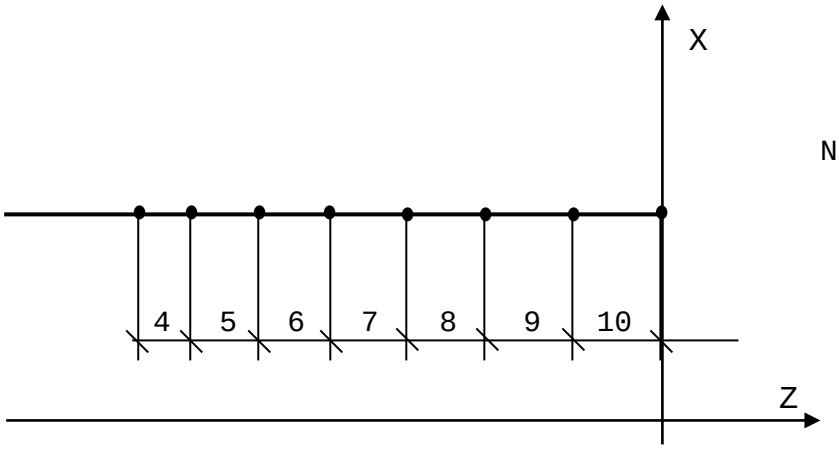
N24 G33 Z-50 K4 I1

а) цилиндрическая резьба с возрастающим шагом



N24 G33 X50 Z-40 K4 I1

б) коническая резьба с возрастающим шагом

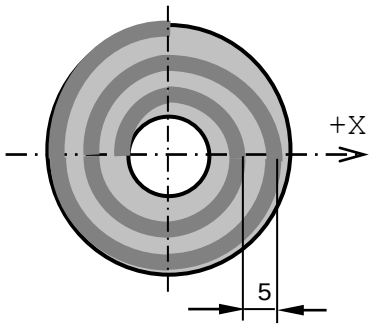


N24 G33 Z-50 K10 I-1

в) цилиндрическая резьба с уменьшающимся шагом

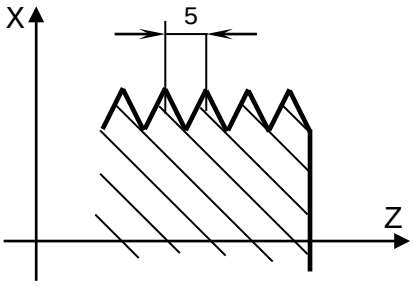
Рисунок А.7 – Примеры нарезания резьбы с переменным шагом

N15 G34 X0 K-5



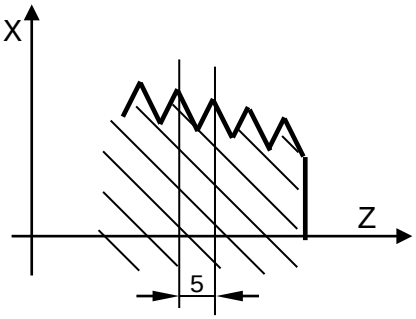
а) фронтальная резьба

N22 G34 Z-50 K5



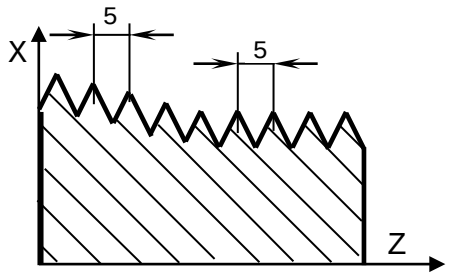
б) цилиндрическая резьба

N21 G00 X20 Z0
N22 G34 X30 Z-50 K5



в) коническая резьба

N22 G34 Z-50 K5
N22 G34 X30 Z-50 K5



г) цилиндрическо-коническая резьба

Рисунок А.8 - Примеры нарезания резьбы с постоянным шагом (G34)

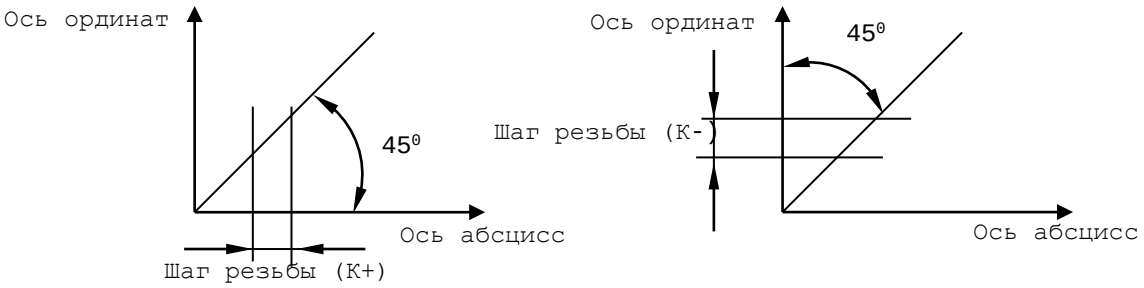


Рисунок А.9 – Угол наклона конической резьбы

```
N35 T5.5 M6
N36 G0 G97 X24 Z37 S250 M3 M8
N37 (FIL,Z4,K2,L5.1,R2)
N38 G0 X250 Z215
```

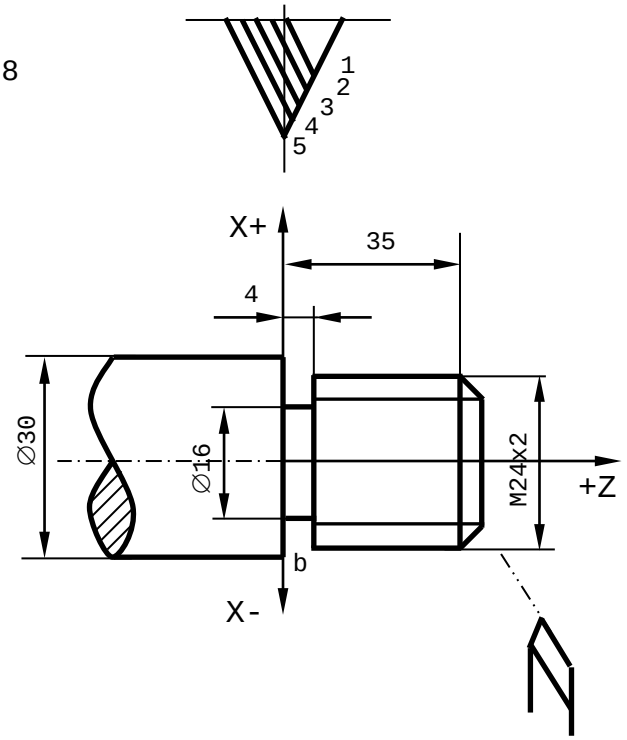


Рисунок А.10 – Цикл нарезания резьбы с врезанием под углом

```
N35 T5.5 M6
N36 G0 G97 X24 Z37 S250 M3 M8
N37 (FIL,Z4,K2,L5.1,T1400,R2)
N38 G0 X250 Z215
```

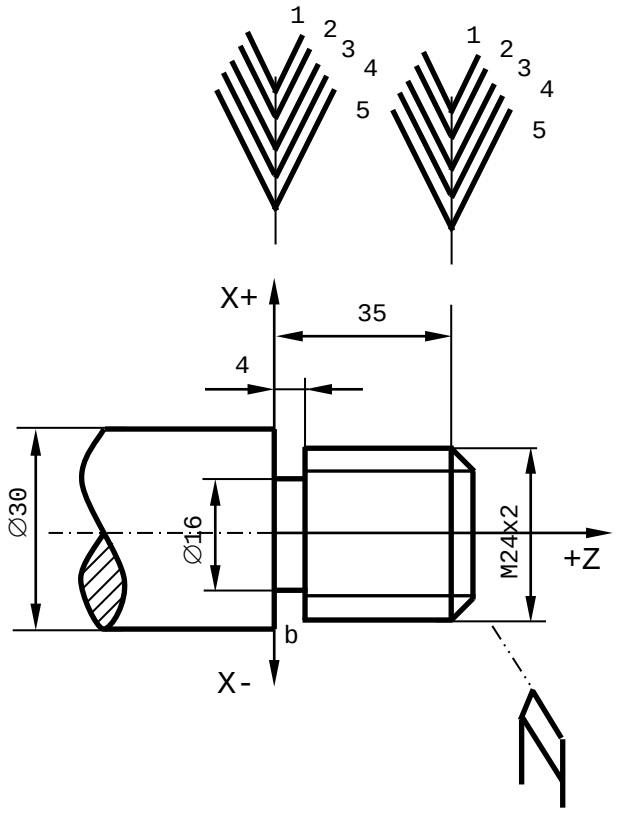
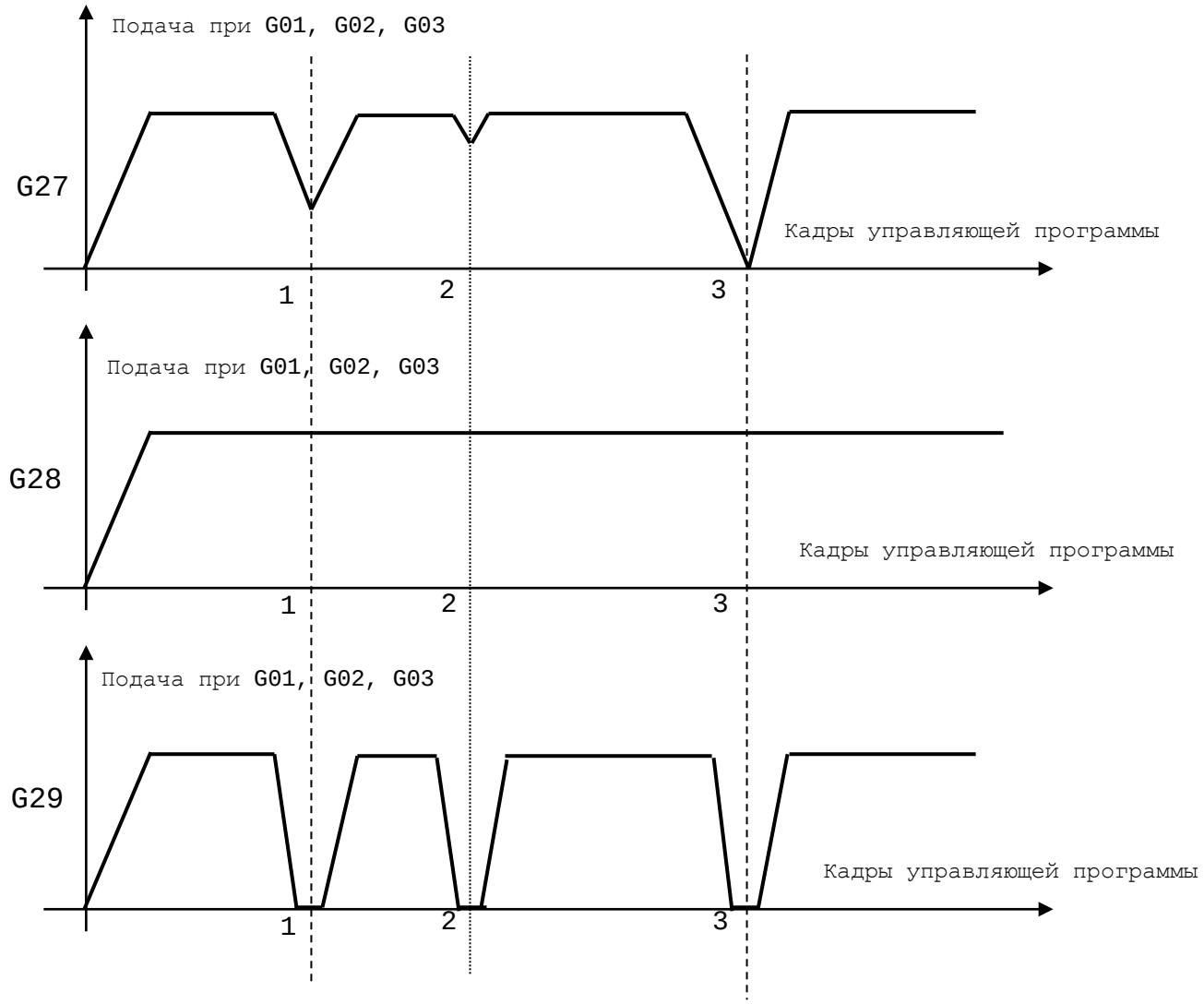
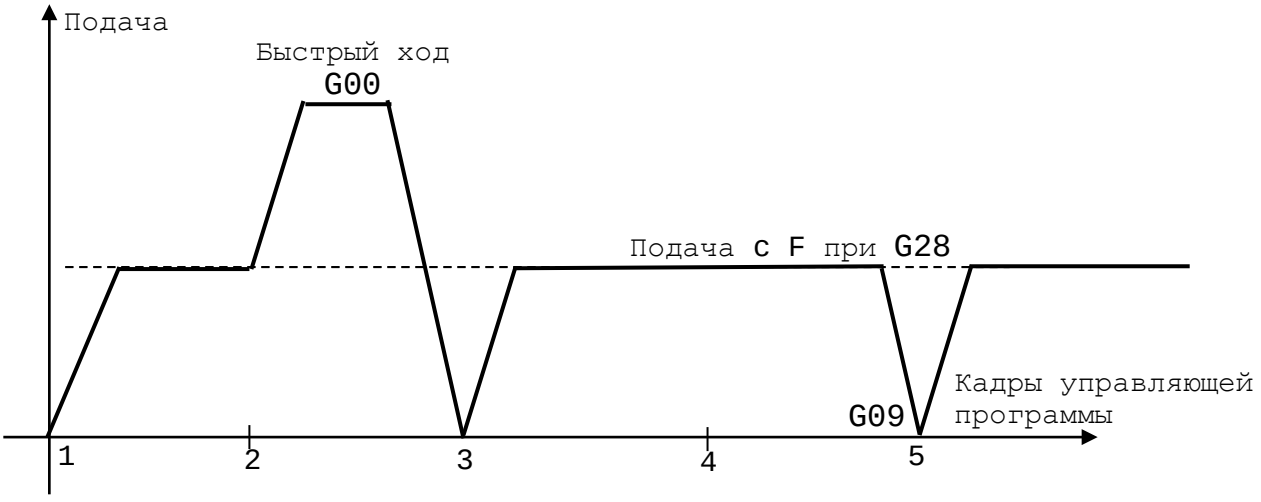


Рисунок А.11 - Цикл нарезания резьбы с радиальным врезанием, конечным пазом и торможением по G09



а) от кадра к кадру по функциям G27, G28, G29



б) программирование контура при непрерывном режиме и в режиме G00

Рисунок А.12 - Графическое изображение режимов динамики движения

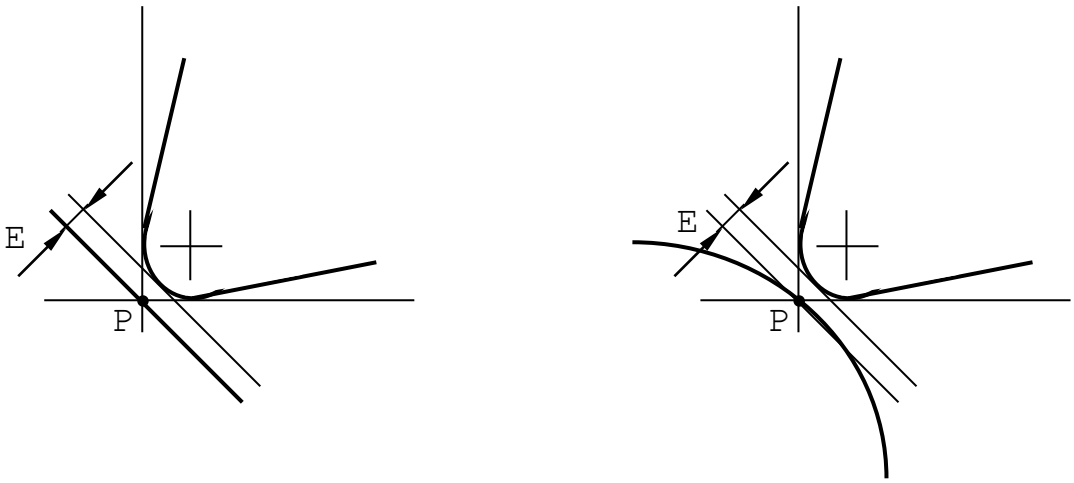


Рисунок А.13 - Теоретический профиль детали

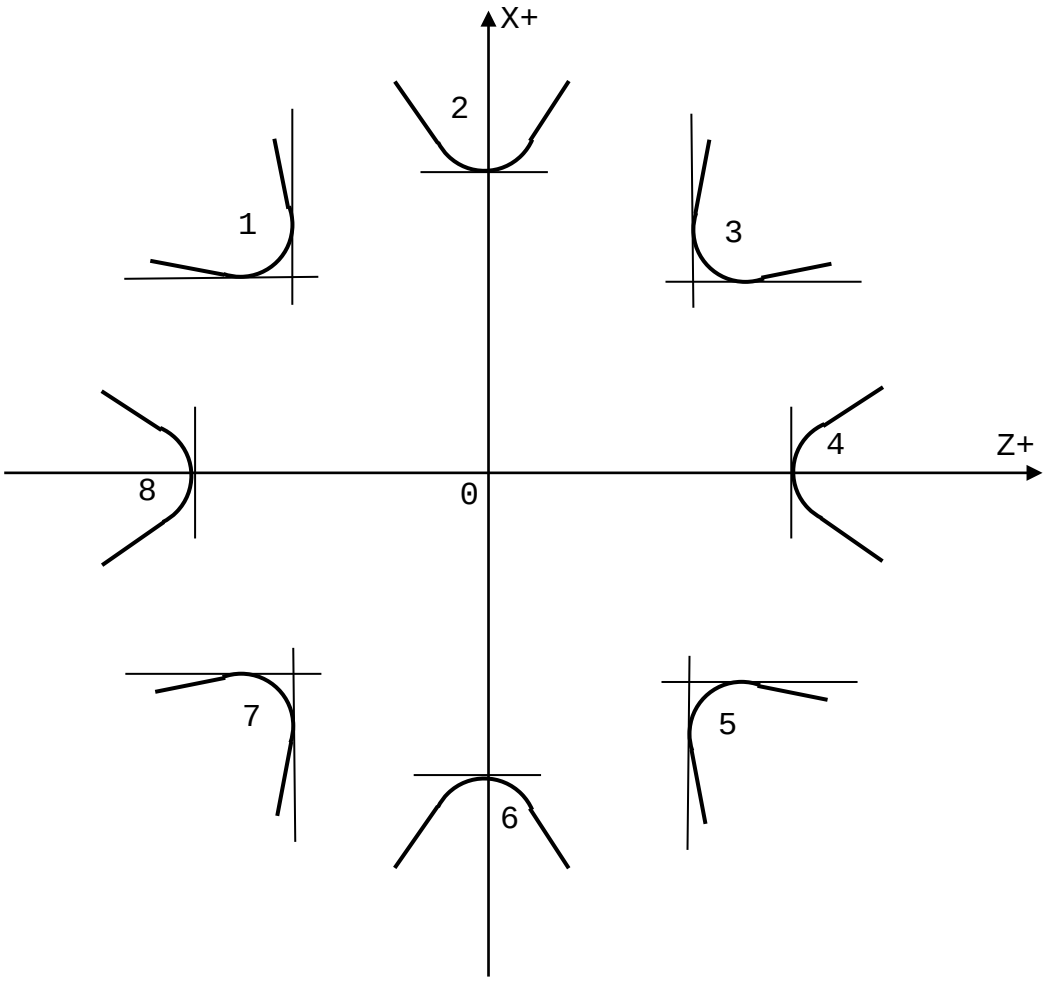


Рисунок А.14 - Коды ориентации

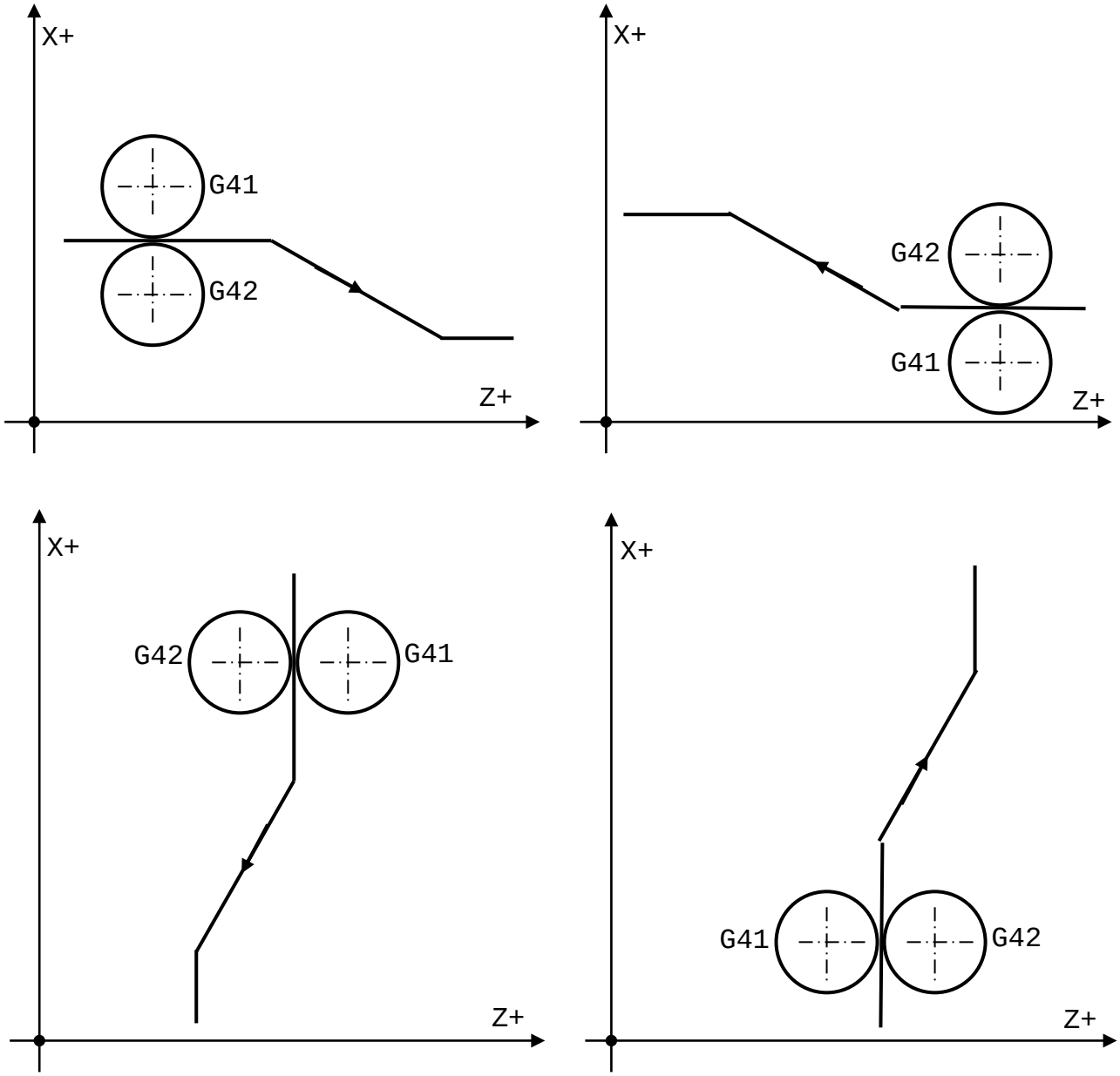
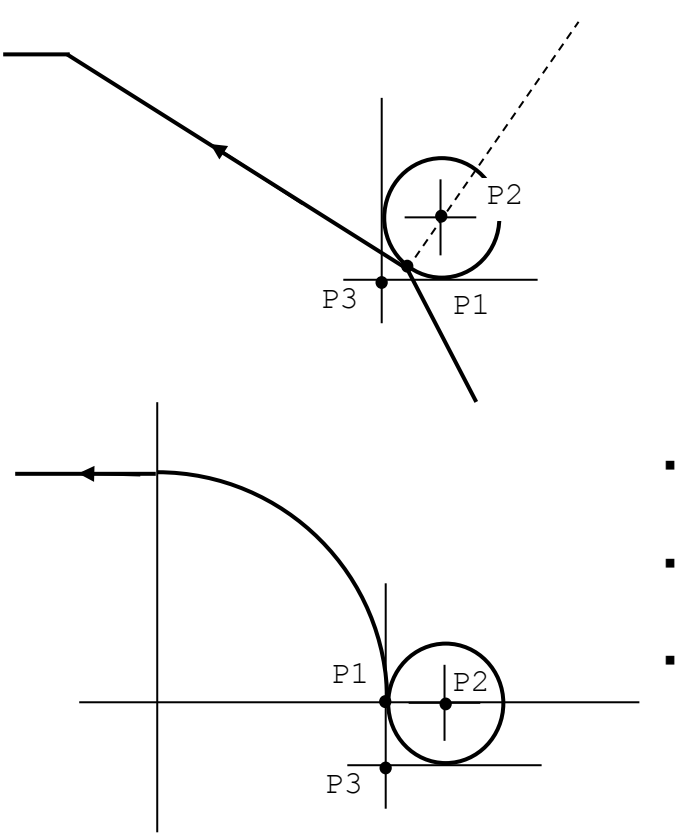
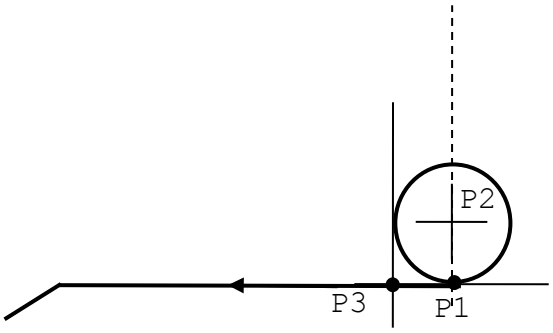


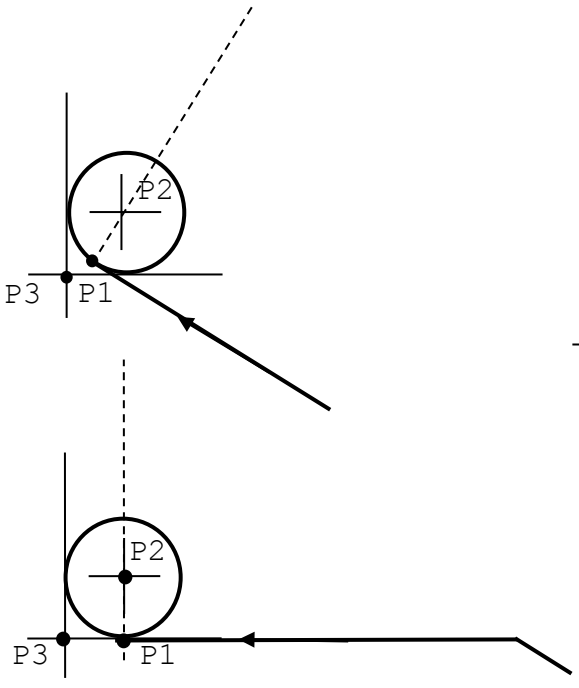
Рисунок А.15 - Применение функций G41, G42

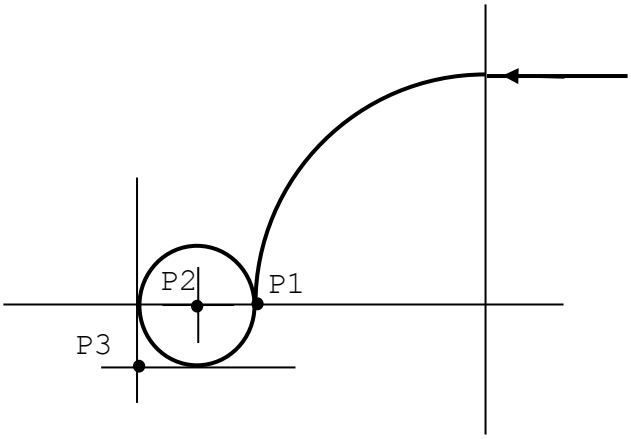




- P1 -запрограммированная точка
- P2 -точка центра инструмента
- P3 -теоретическая позиция точки (с **G41**)

а) начало профиля





б) окончание профиля

Рисунок А.16 - Начало и конец профиля с компенсацией конца инструмента

```
Радиус инструмента = 2мм
Код ориентации = 3
```

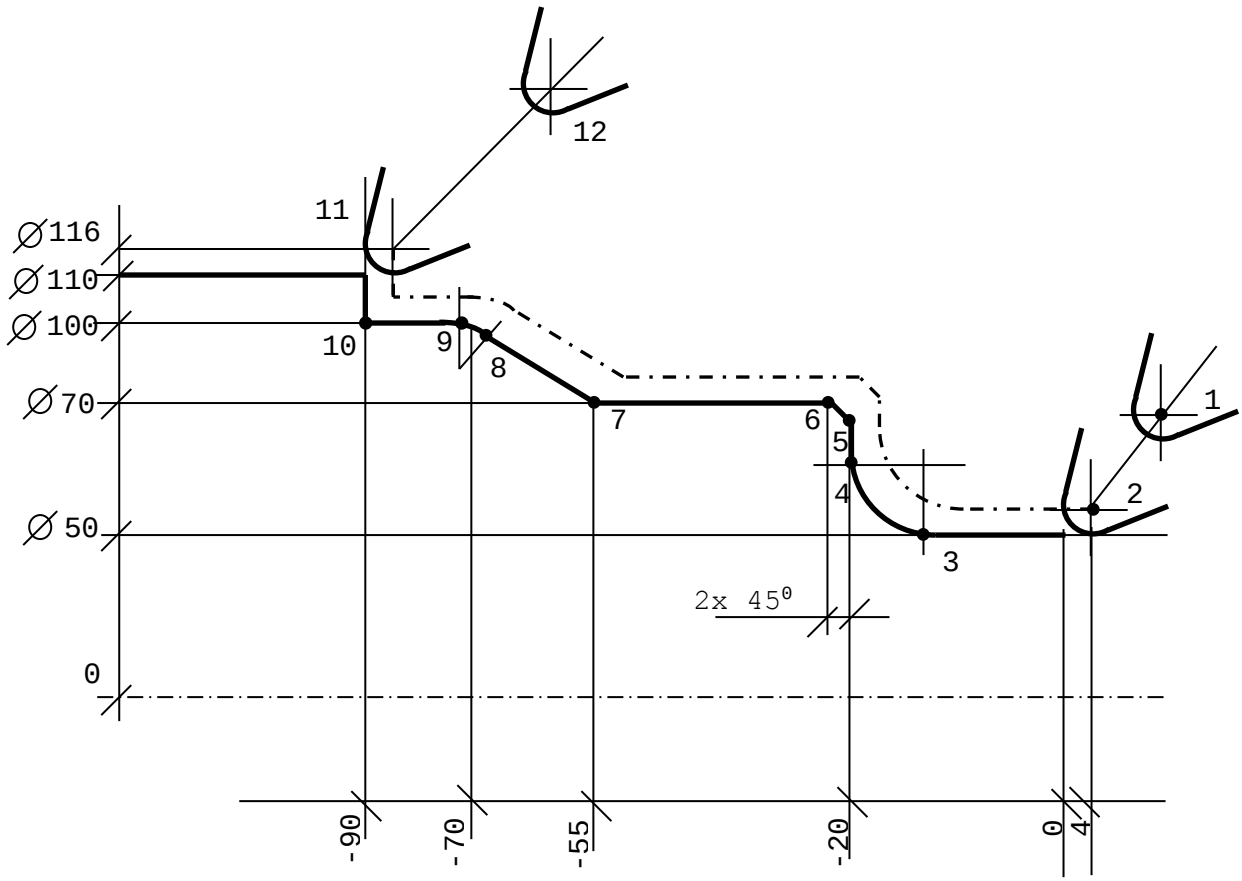
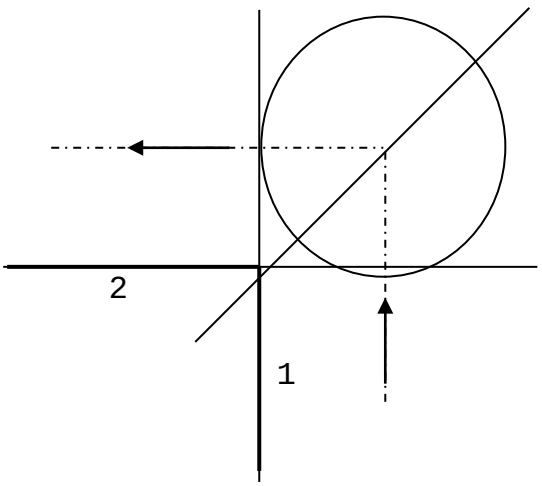


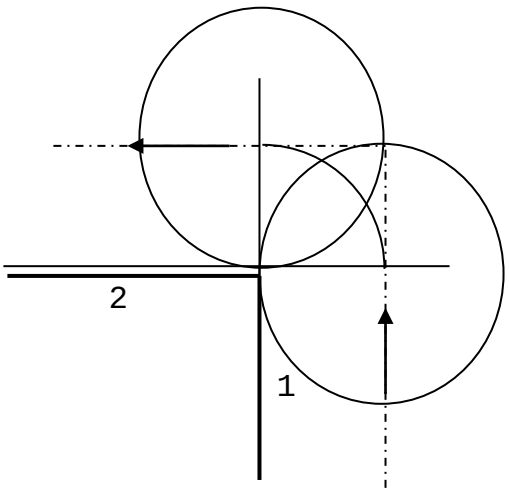
Рисунок А.17 - Пример программирования функций G41/G42/G40




```
1  N20 G1 X100

2  N21 Z-100
```

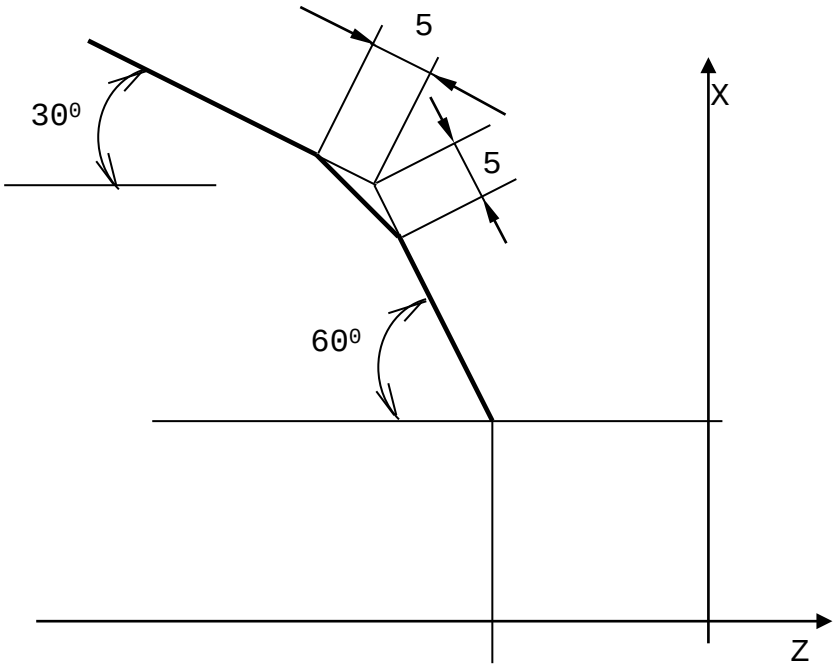
а) без радиуса скругления




```
1  N20 G1 X100
   N21 P0
2  N22 Z-100
```

б) с радиусом скругления

Рисунок А.18 - Примеры программирования обхода угла




```
N10 G1 X50 Z
N11 X100 Z-30
N12 b5
N13 Z-60
```

Рисунок А.19 - Пример программирования скоса (фаски)

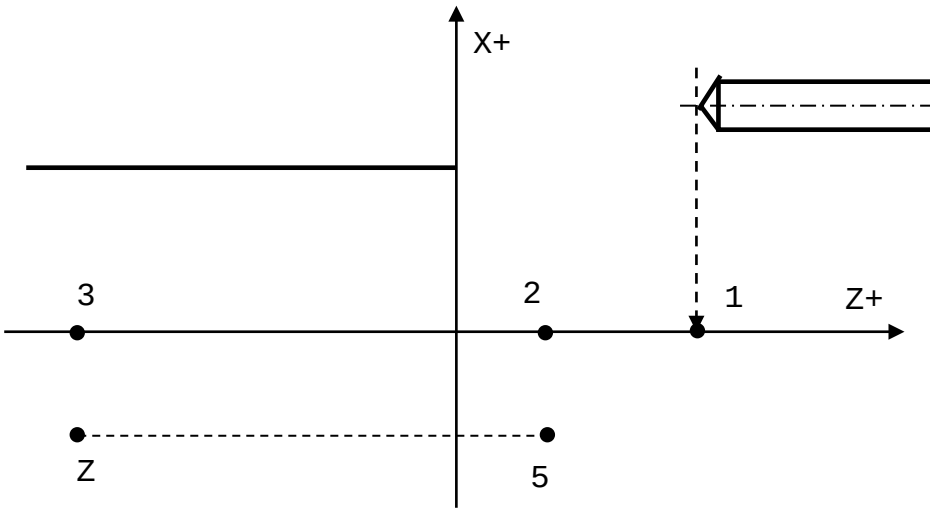
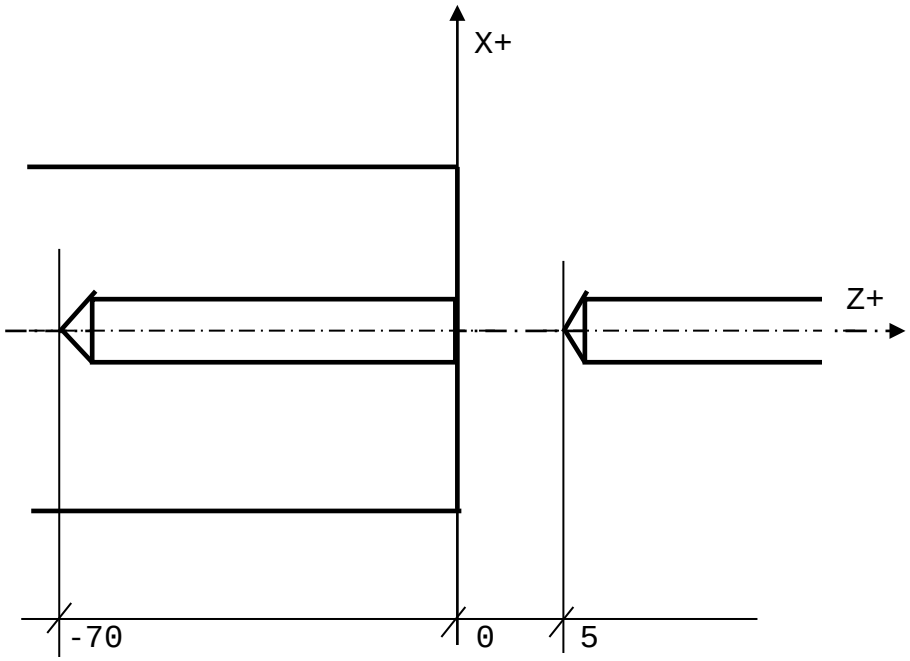
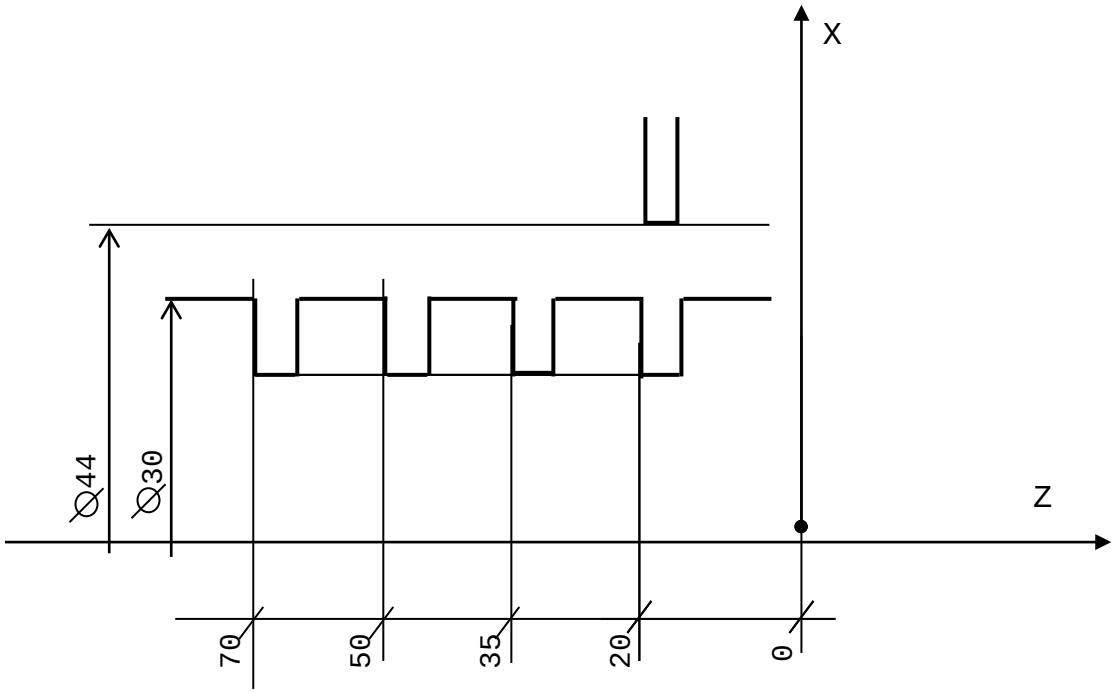


Рисунок А.20 - Постоянный цикл с R2=R1 и R2 не равно R1



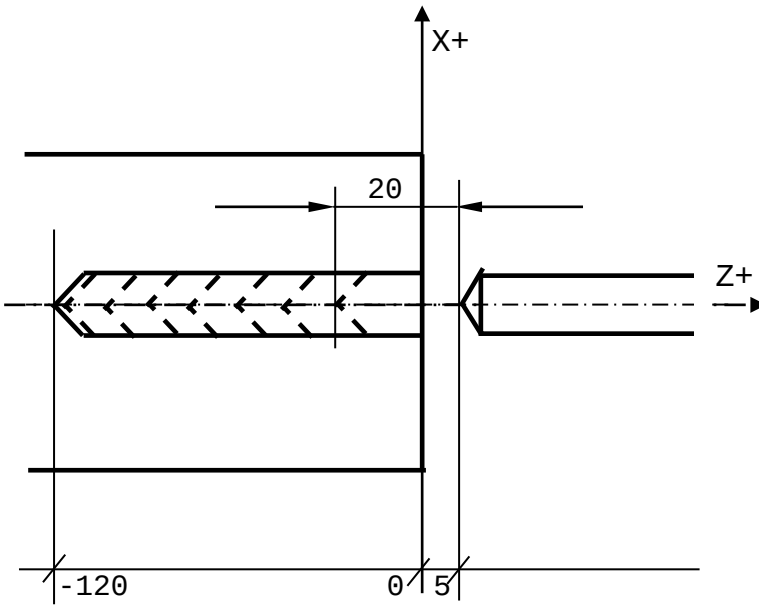

```
N31 (DIS,”TWIST DRILL D=6.5)
N32 G97 S1000 T4.4 M06 M3 M7
N33 G81R5 Z-70 F0.2
N34 X0
N35 G80
N36 G.. X.. Z..
```

Рисунок А.21 - Пример программирования постоянного цикла G81 (сверление)




```
N10 T4.4 M6 S100 M3 M8
N11 TMR = 2
N12 G82 R44 X30
N13 Z-20
N14 Z-35
N15 Z-50
N16 Z-70
N17 G80 X.. Z..
```

Рисунок А.22 - Пример программирования постоянного цикла G82 (растачивание)




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

Рисунок А.23 - Пример программирования постоянного цикла G83 (глубокое сверление с разгрузкой стружки)

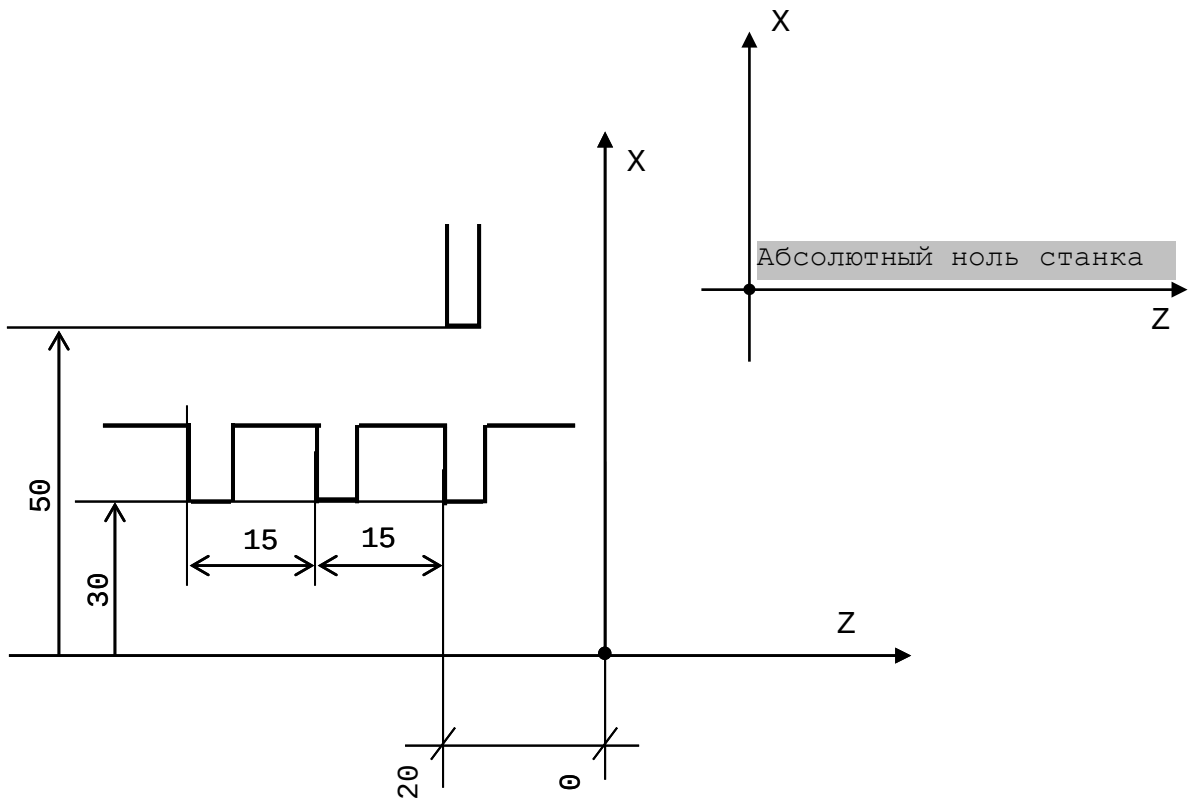


Рисунок А.24 - Пример программирования в приращениях (G91)

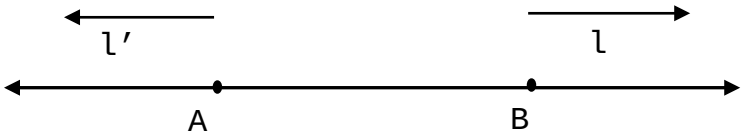
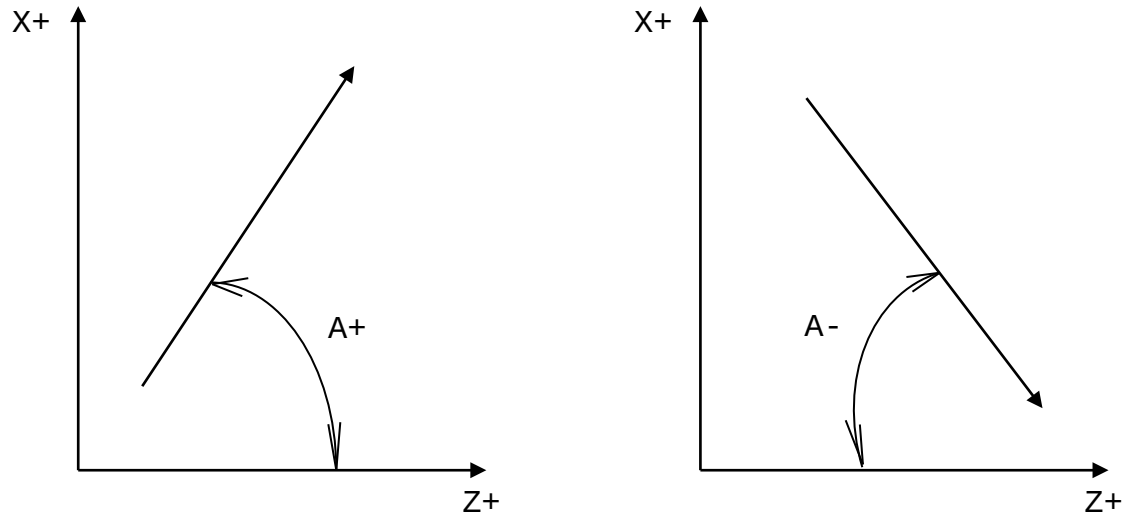
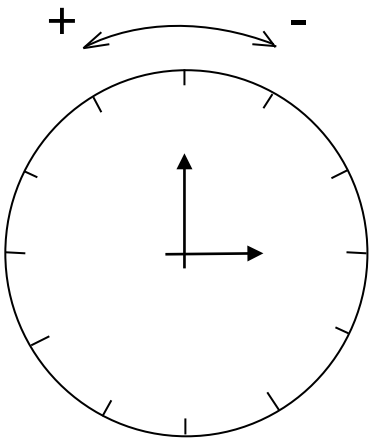


Рисунок А.25 – Представление прямой линии в векторной геометрии



а) для прямой линии



б) для окружности

Рисунок А.26 – Определение направления движения в геометрическом программировании

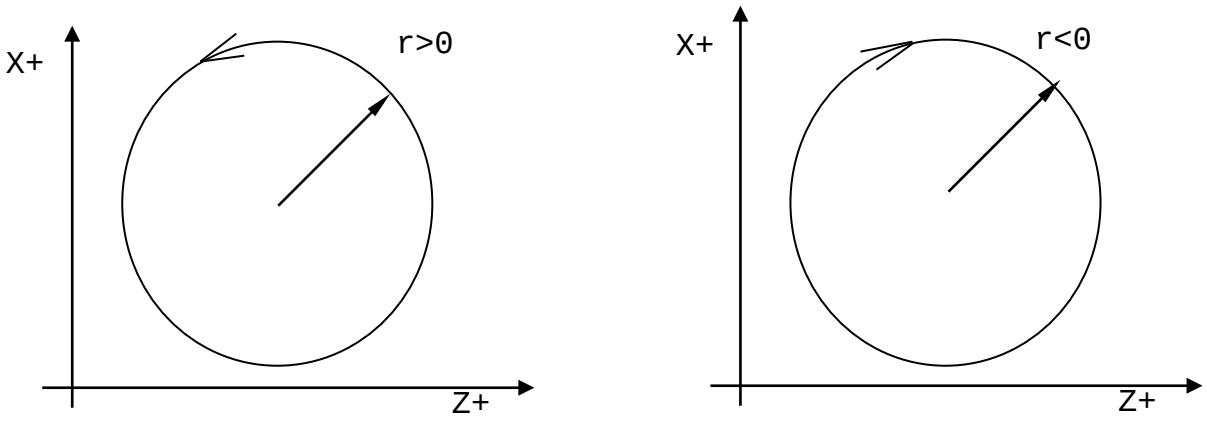


Рисунок А.27

p1=c1,l3

p2=c1,l3,s2

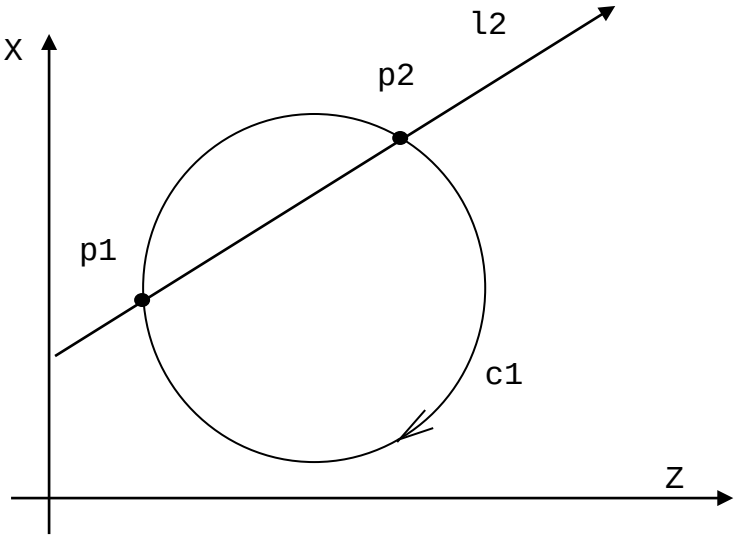


Рисунок А.28

o4=Z20X30a-45

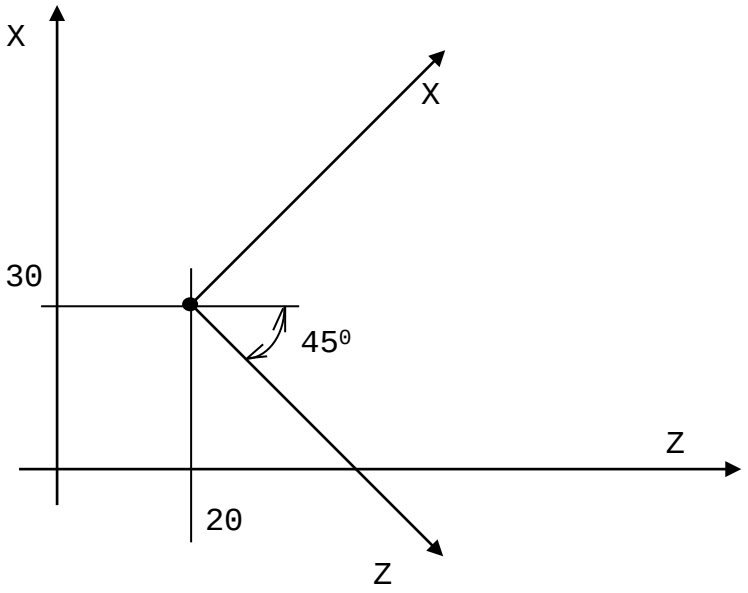


Рисунок А.29

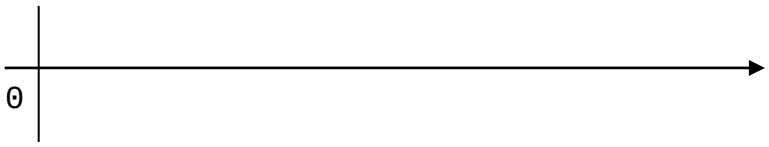


Рисунок А.30 - Полярные оси

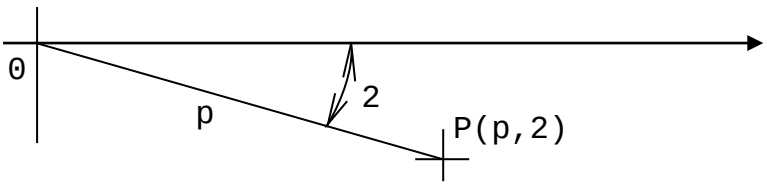


Рисунок А.31 - Полярные координаты

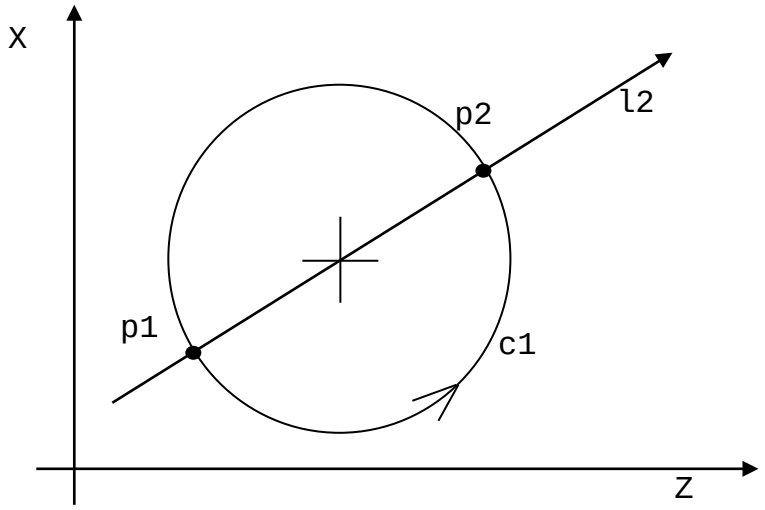


Рисунок А.32

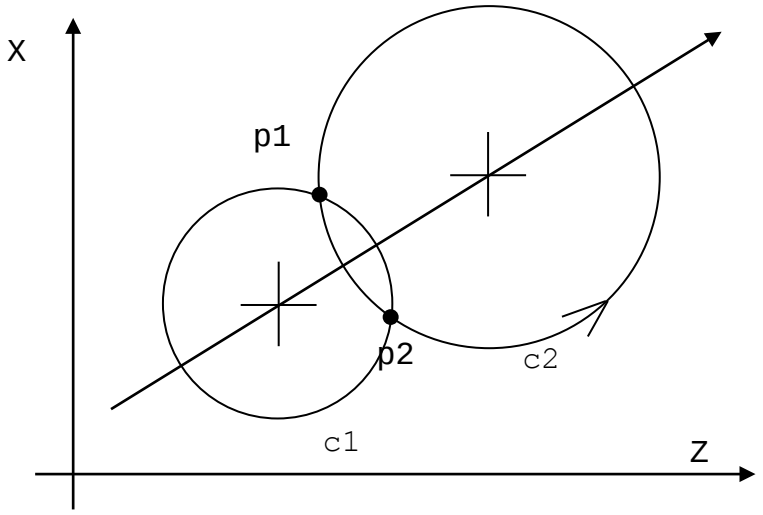


Рисунок А.33

p1=Z30X160

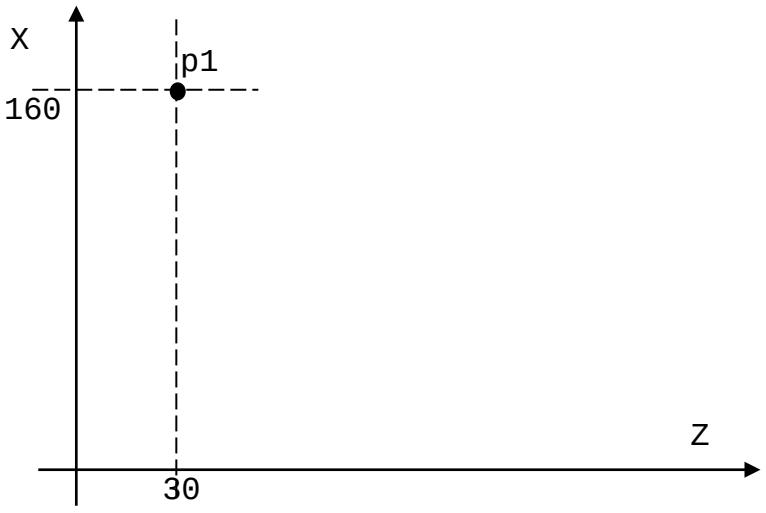


Рисунок А.34

```
o1=Z30X30a-20
p5=o1Z20X10
```

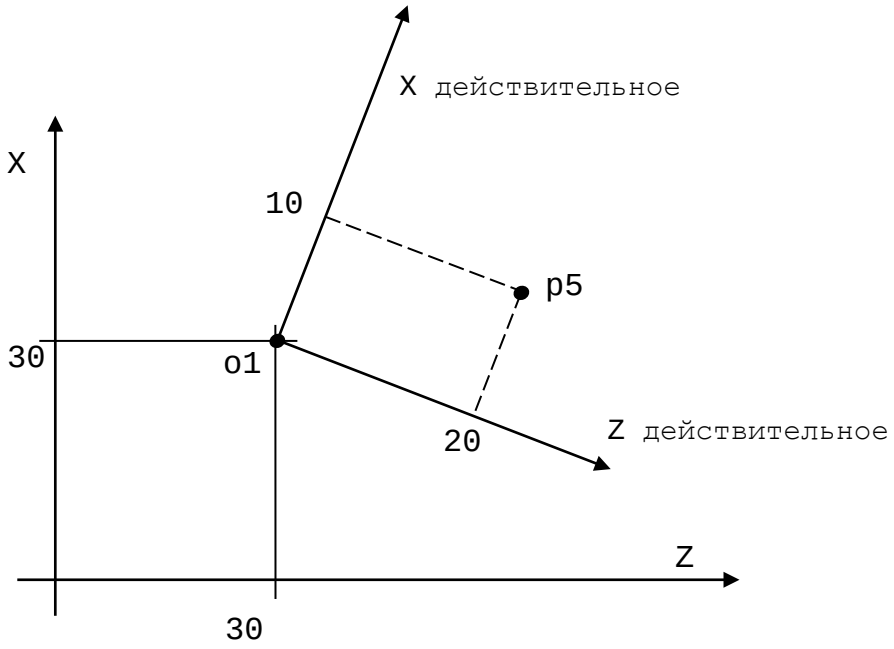


Рисунок А.35

```
p2=m55a60
```

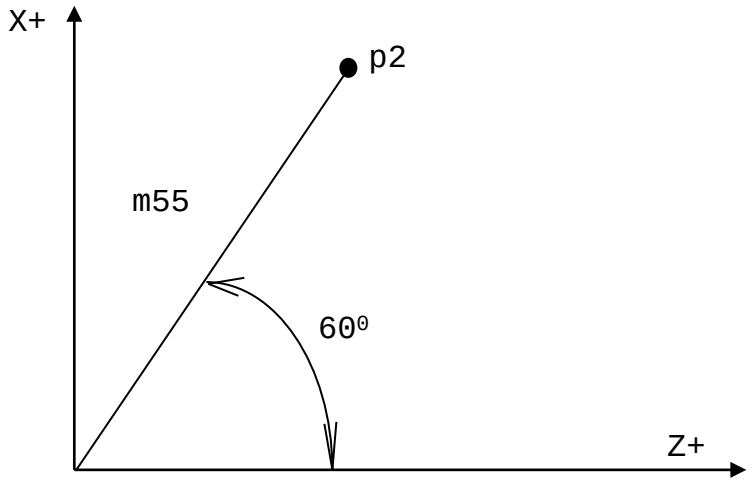


Рисунок А.36

```
p1=l1,l2
```

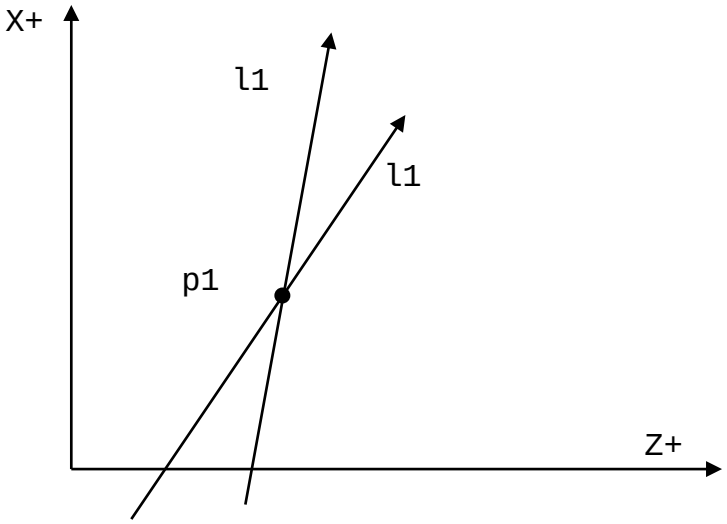


Рисунок А.37

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

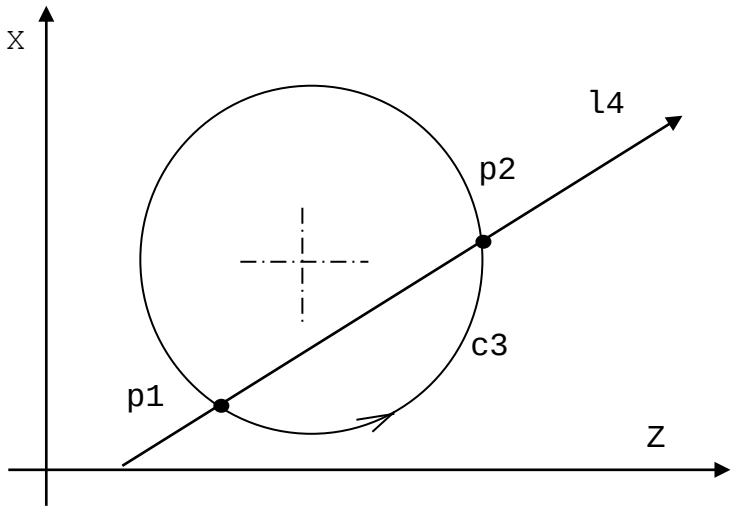


Рисунок А.38

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

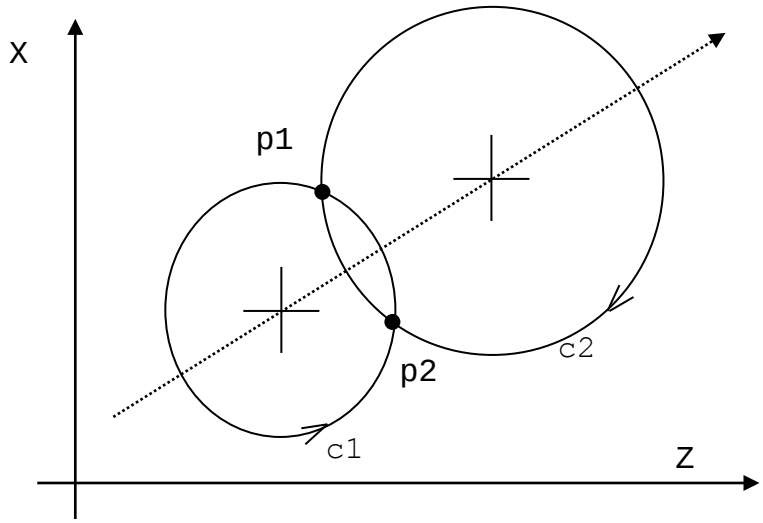


Рисунок А.39

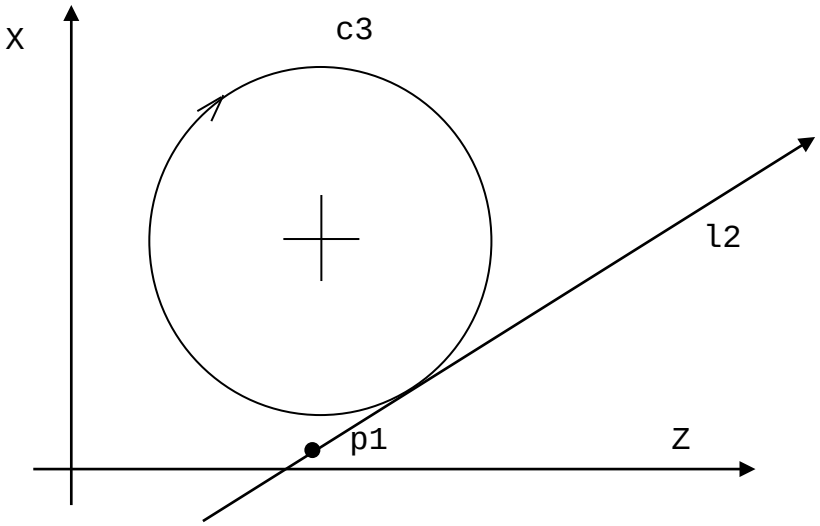


Рисунок А.40

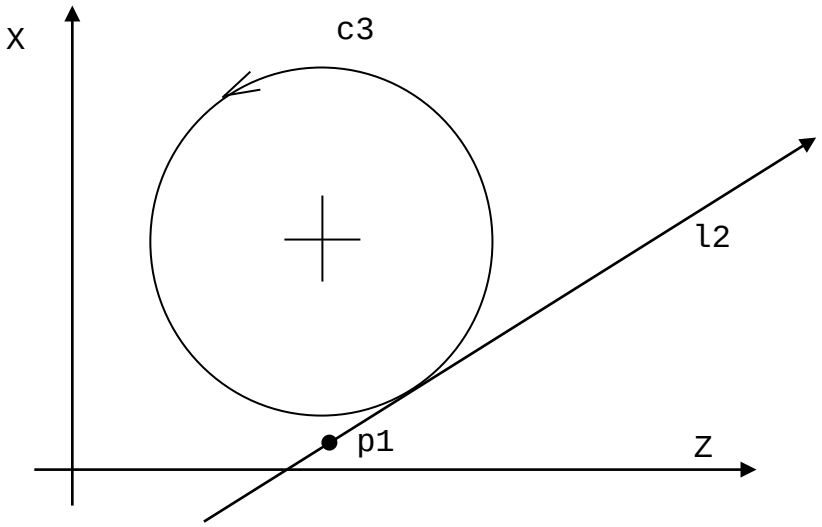


Рисунок А.41

```
l1=Z40X20,Z60X70
```

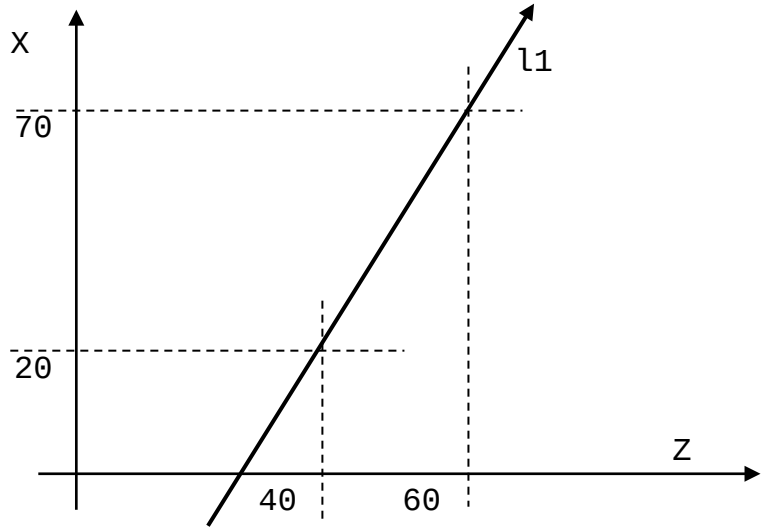


Рисунок А.42

```
l1=Z10X15,I45J30r-15
l2=Z10X15,I45J30r15
```

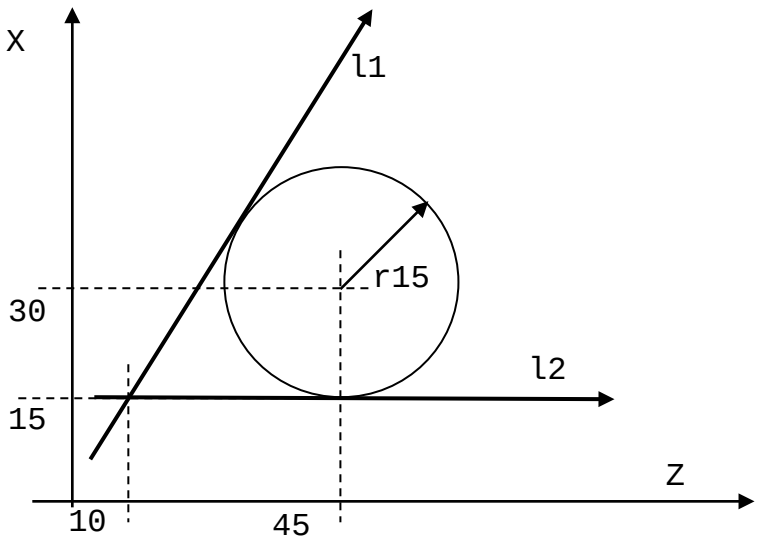


Рисунок А.43

```
l2=Z20X20,a-20
```

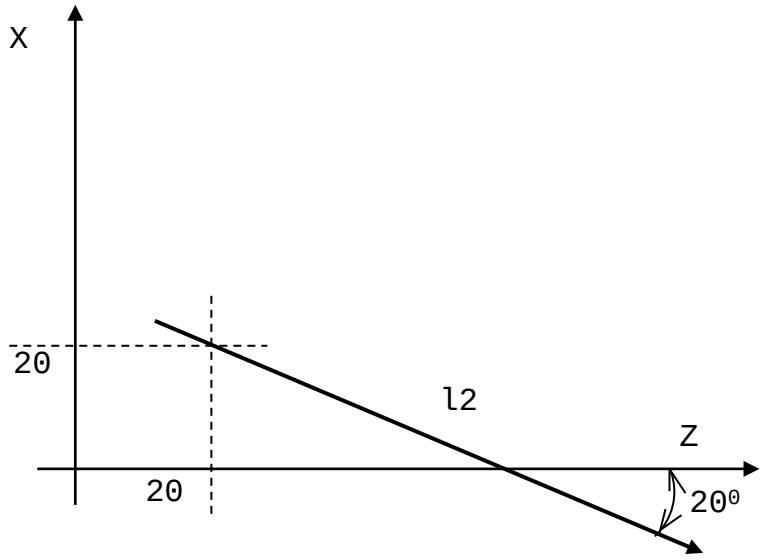


Рисунок А.44

```
o1=Z30X30a-40
l5=o1Z25X30,a60
```

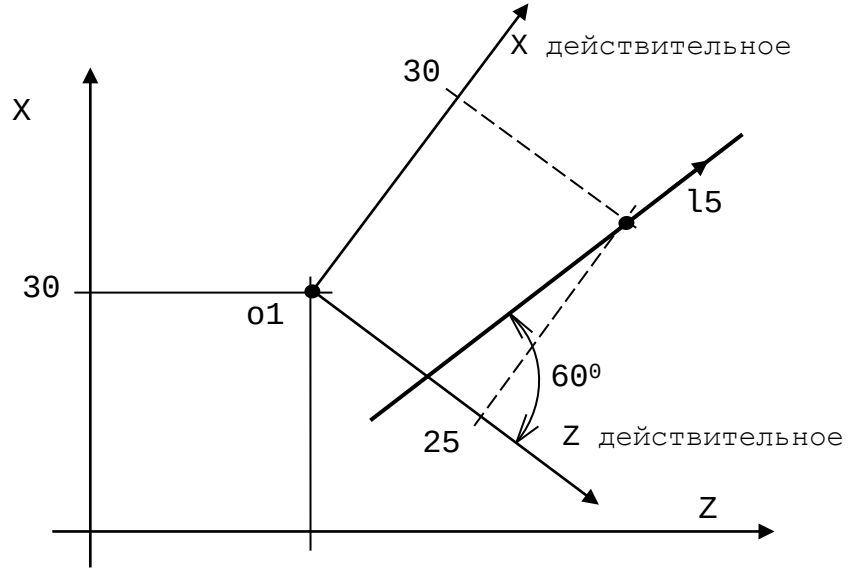


Рисунок А.45

```
l1=I60J80r20,a45
```

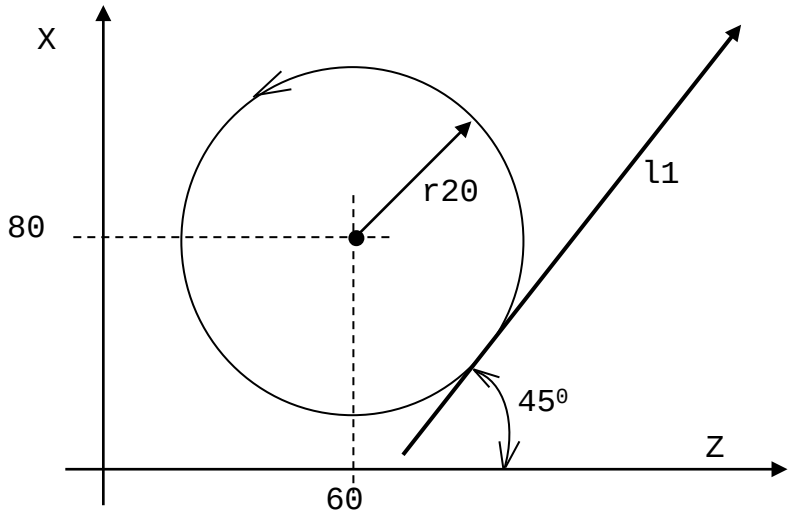


Рисунок А.46

```
o3=Z25X10a20
l4=o3I25J15r10,a115
```

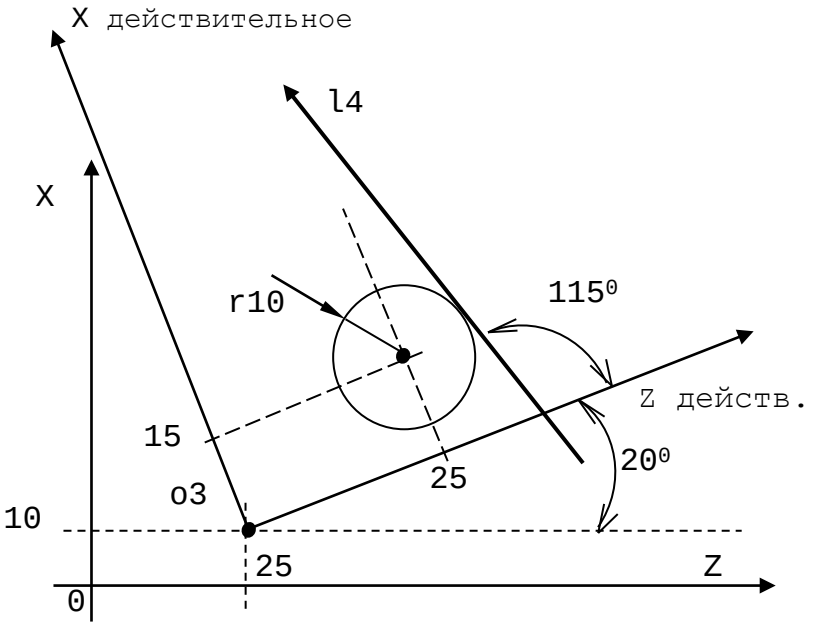


Рисунок А.47

```
l3=I25J35r-17,I70J20r13
```

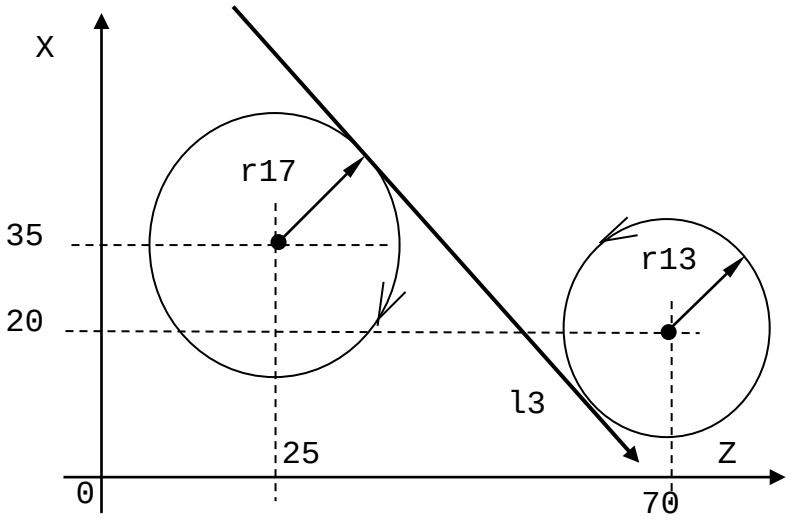


Рисунок А.48

```
l3=I25J35r17,I70J20r13
```

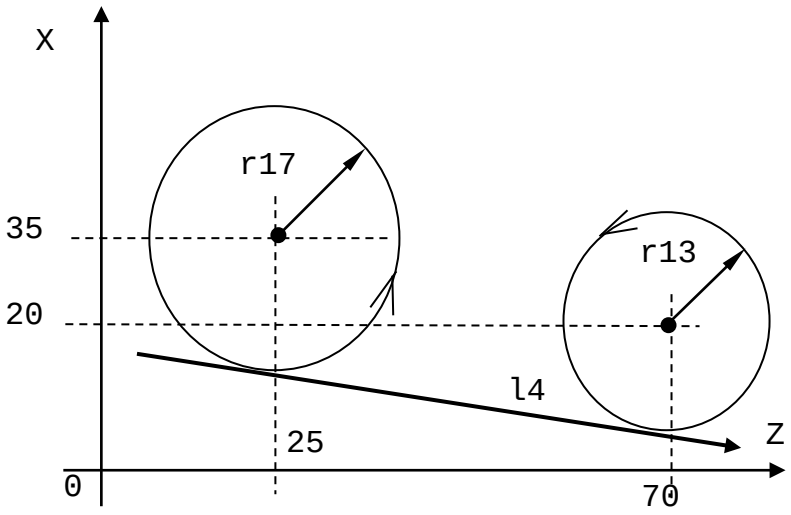


Рисунок А.49

l9=p7,p8

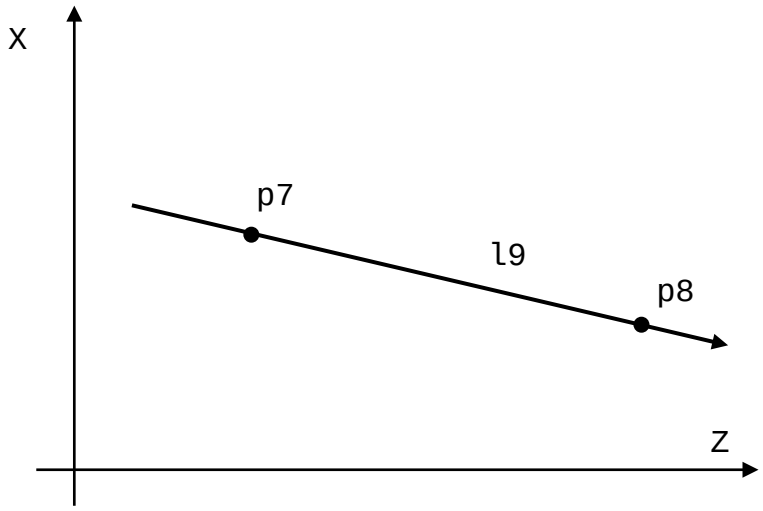


Рисунок А.50

l1=p1,c1
l2=p1,-c1

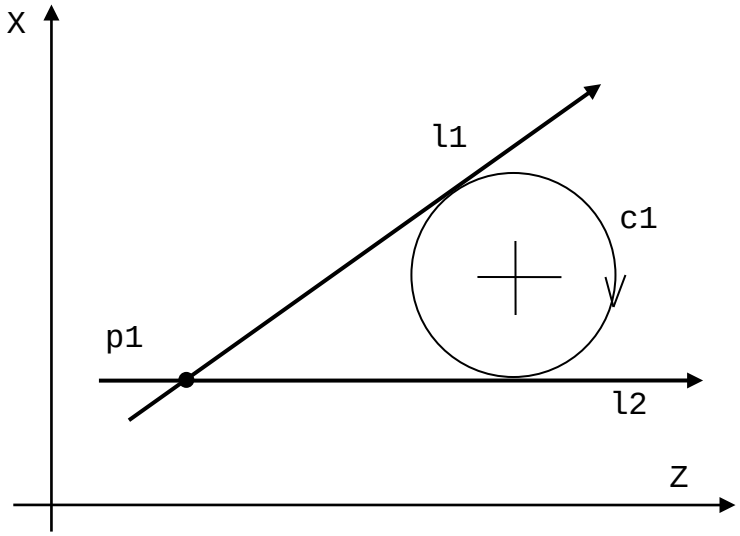


Рисунок А.51

l3=c1,c2

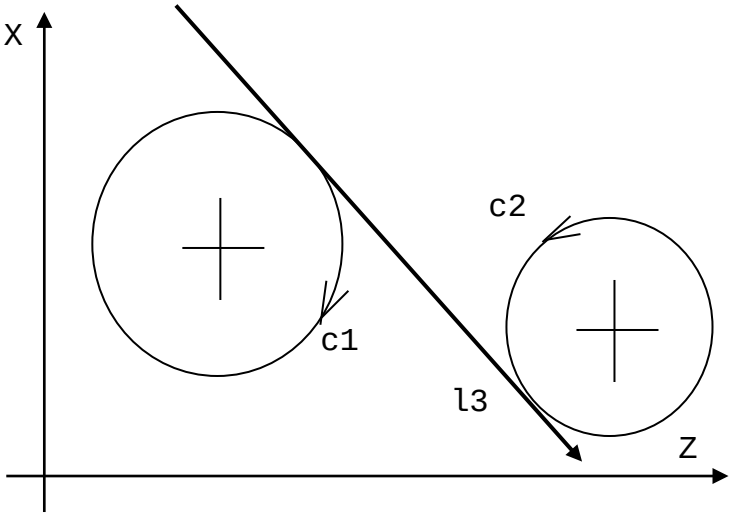


Рисунок А.52

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

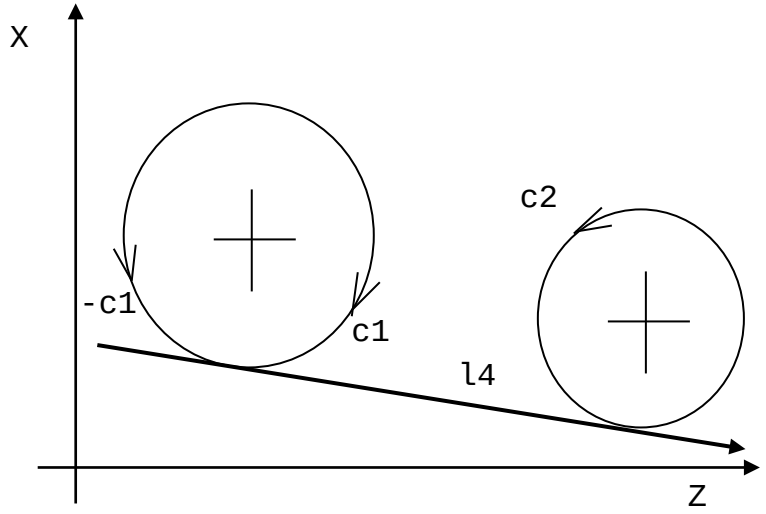


Рисунок А.53

```
l3=p1,a45
```

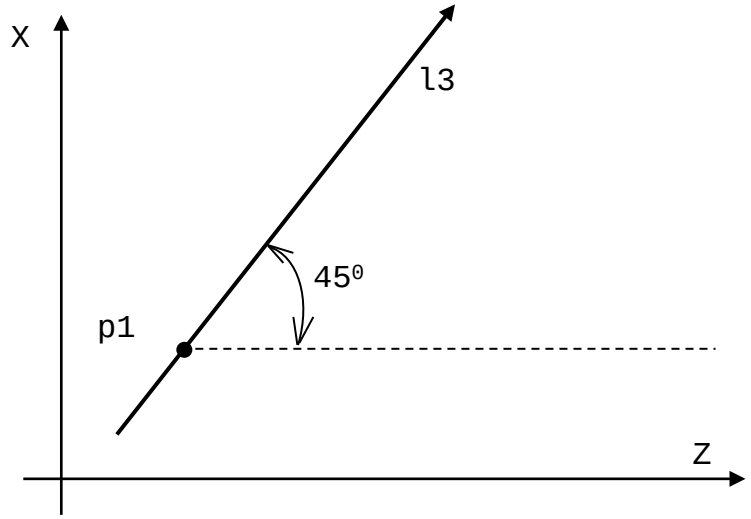


Рисунок А.54

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

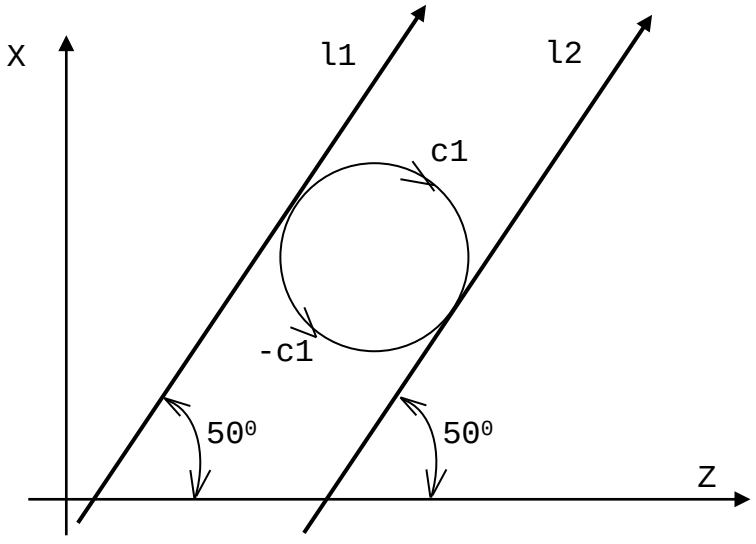


Рисунок А.55

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

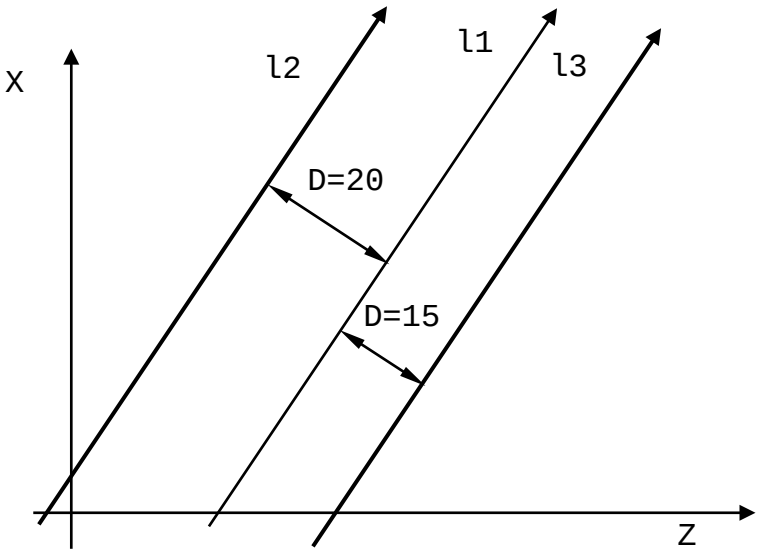


Рисунок А.56

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

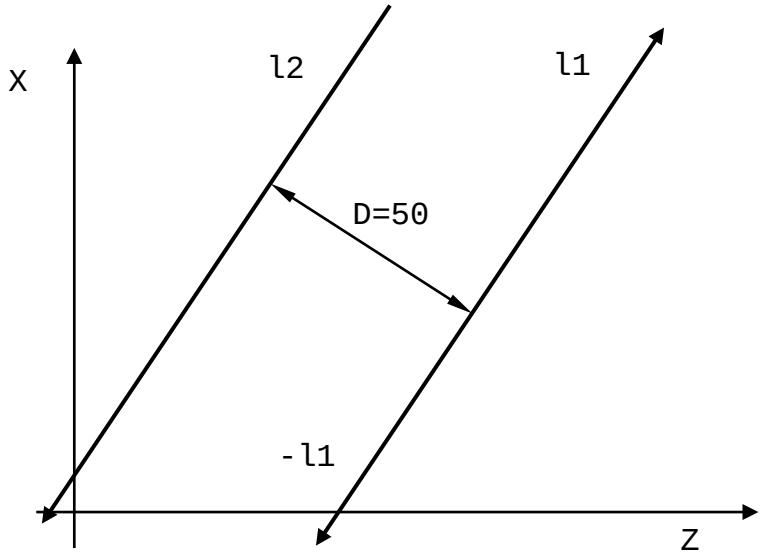


Рисунок А.57

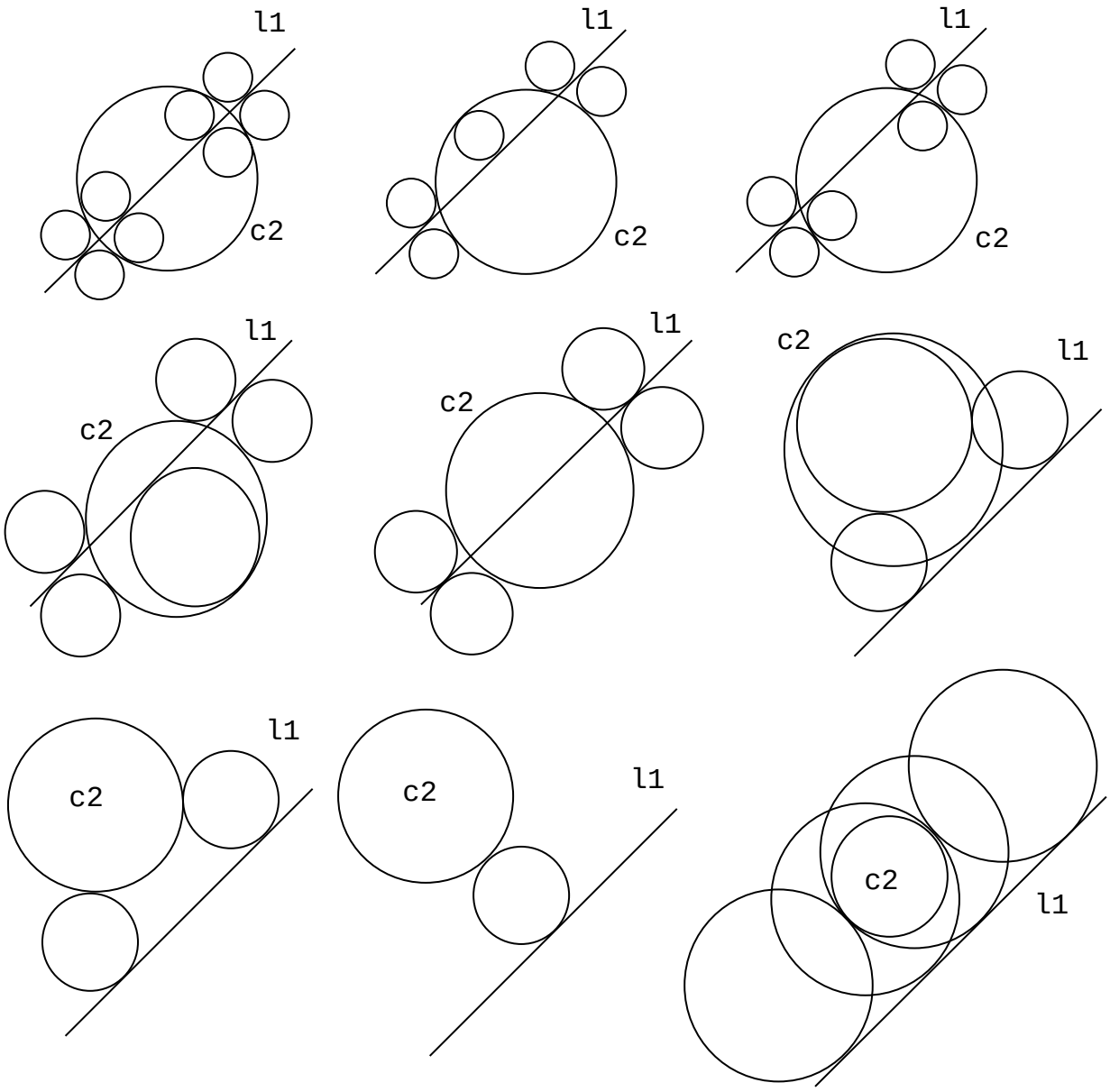
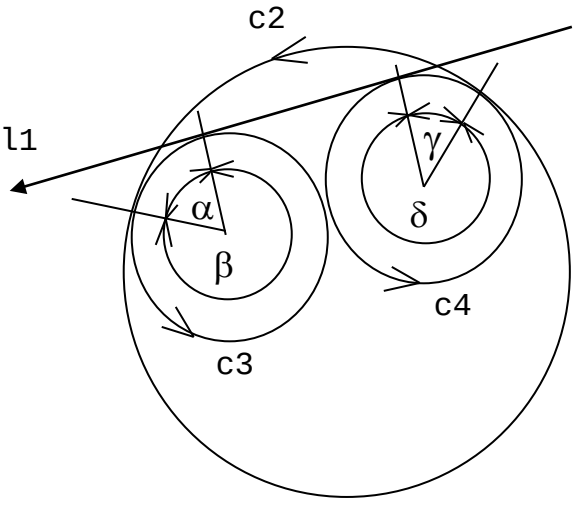


Рисунок А.58



Рискнок А.59

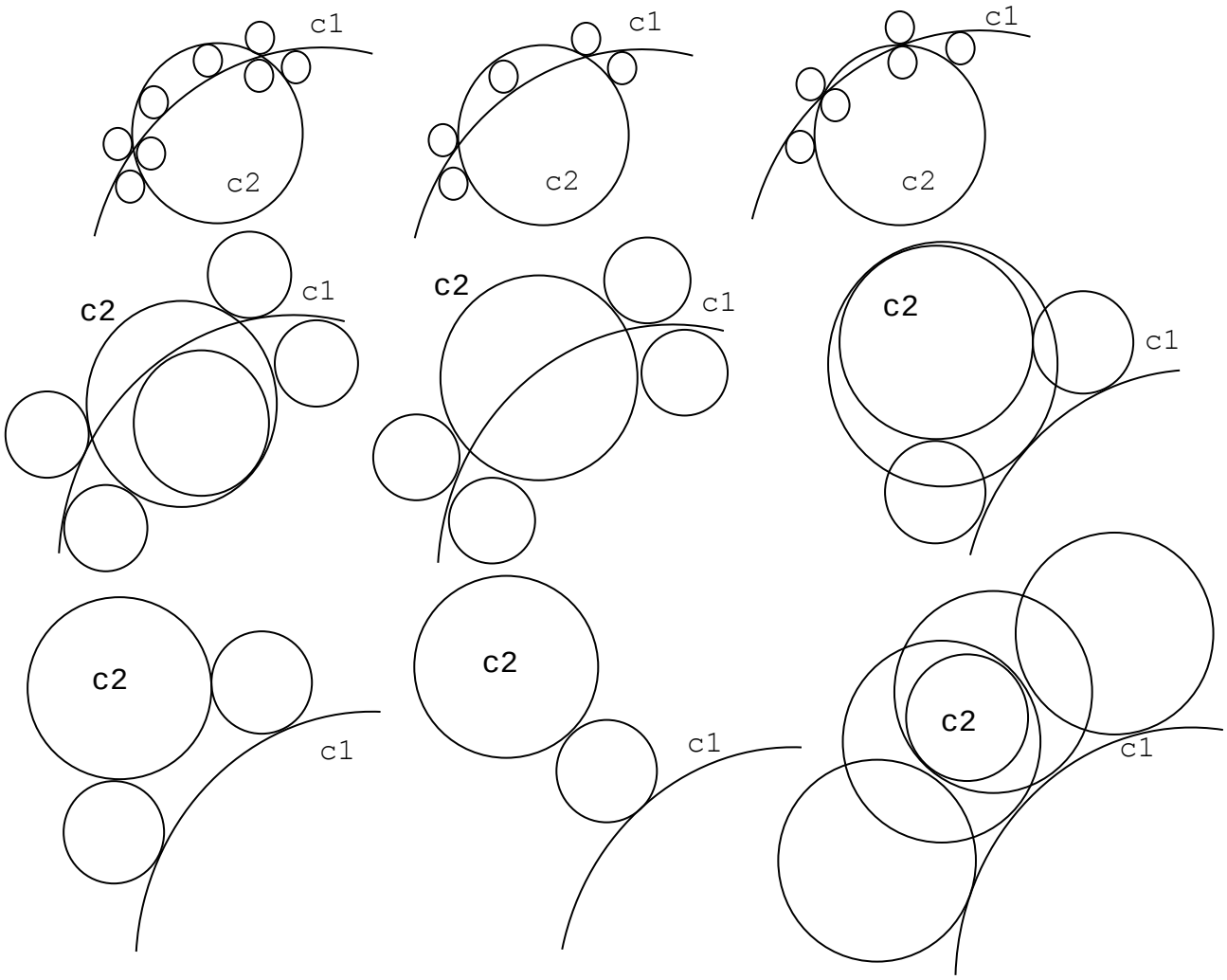


Рисунок А.60

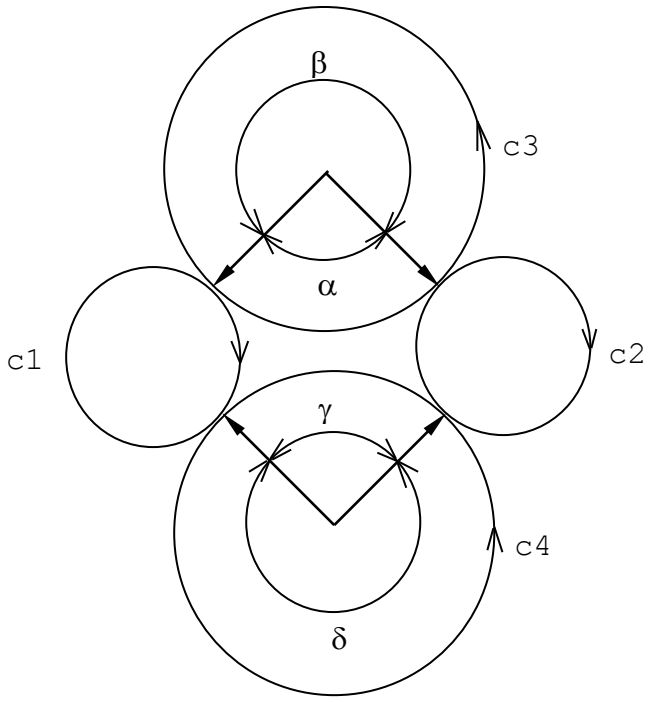


Рисунок А.61

c2=I50J100r-40

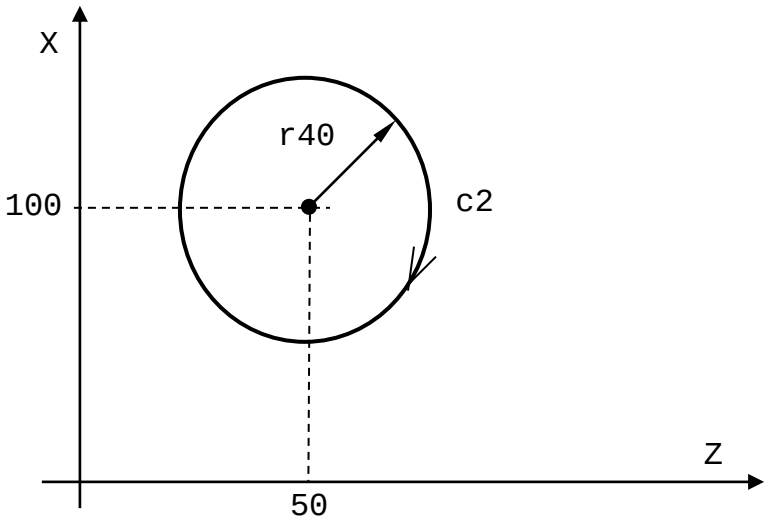


Рисунок А.62

c1=o1I20J20r-15

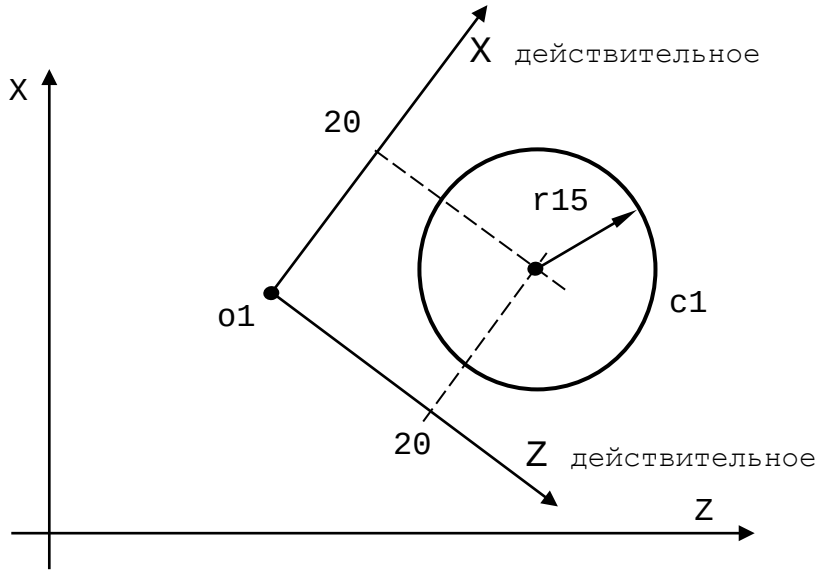


Рисунок А.63

c2=m70a30r15

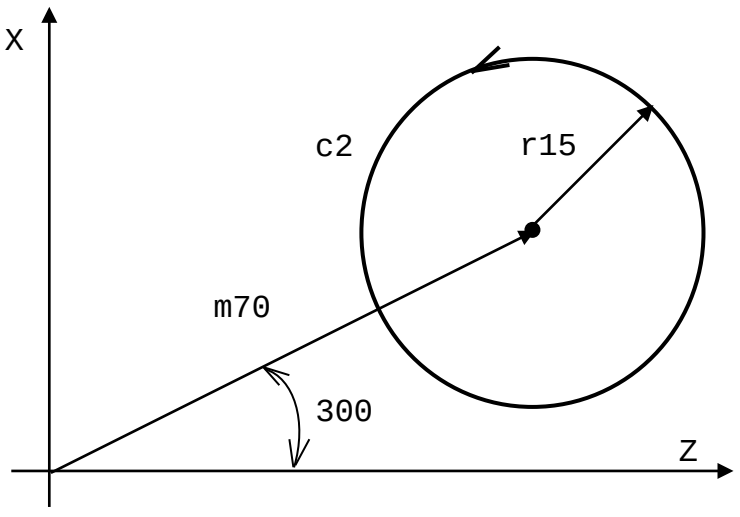


Рисунок А.64

```
c3=l1,l2,r-15
```

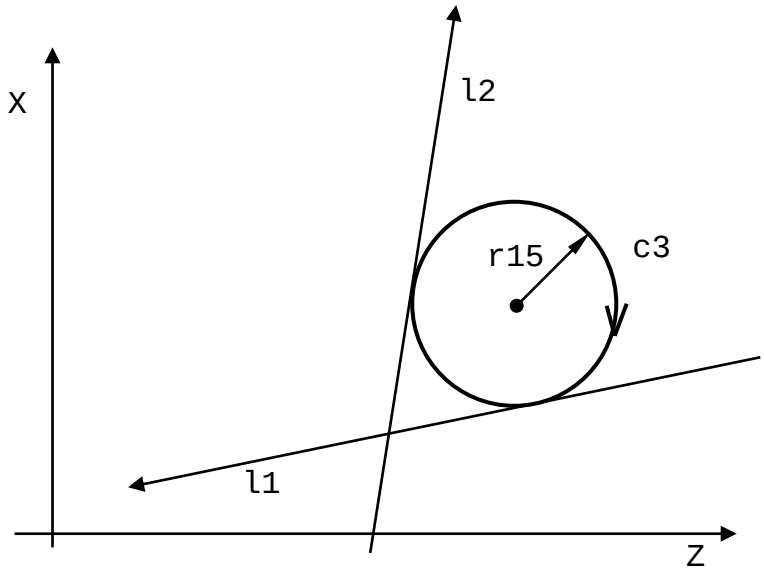


Рисунок А.65

```
c3=l1,-c2,r8
c4=-c2,l1,r8
```

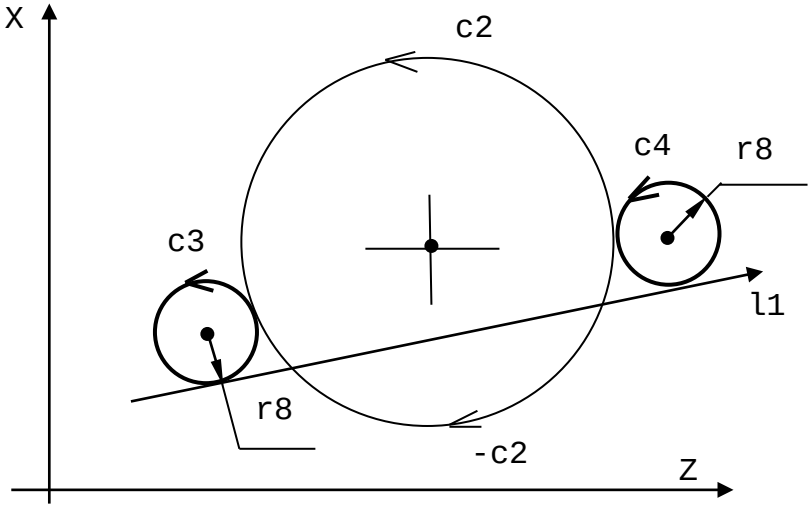


Рисунок А.66

```
c9=-c2,l1,r-8
c10=l1,-c2,r-8
```

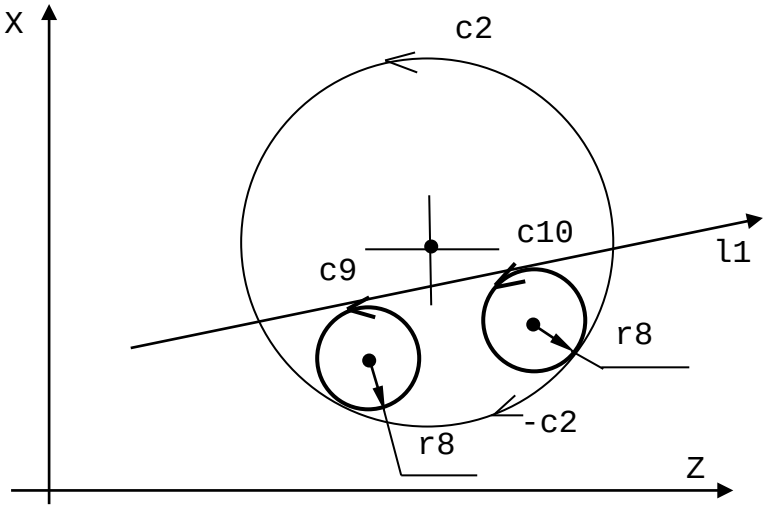


Рисунок А.67

```
c4=-l2,c1,r-40
c5=c1,-l2,r-40
```

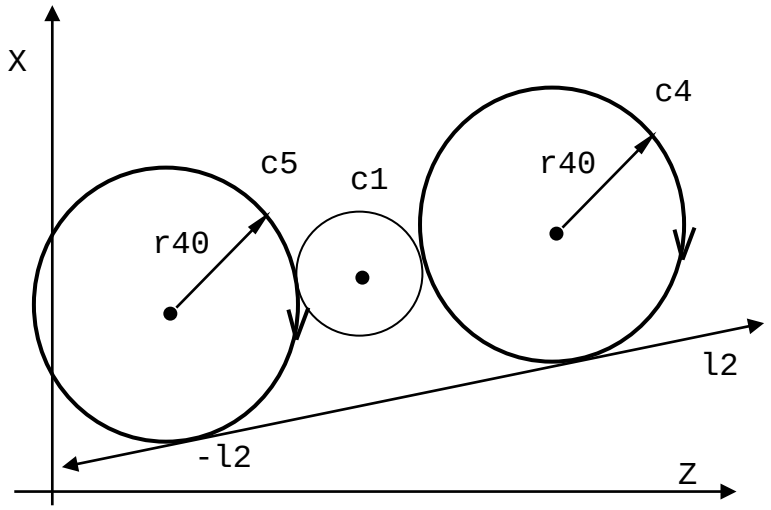


Рисунок А.68

```
c3=p1,-l1,r25
c4=-l1,p1,r25
```

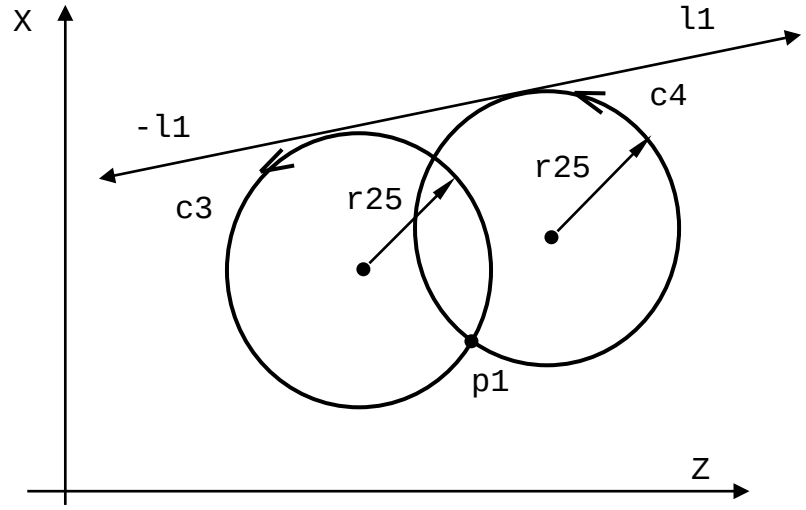


Рисунок А.69

```
c5=c1,c2,r-8
c6=c2,c1,r-8
```

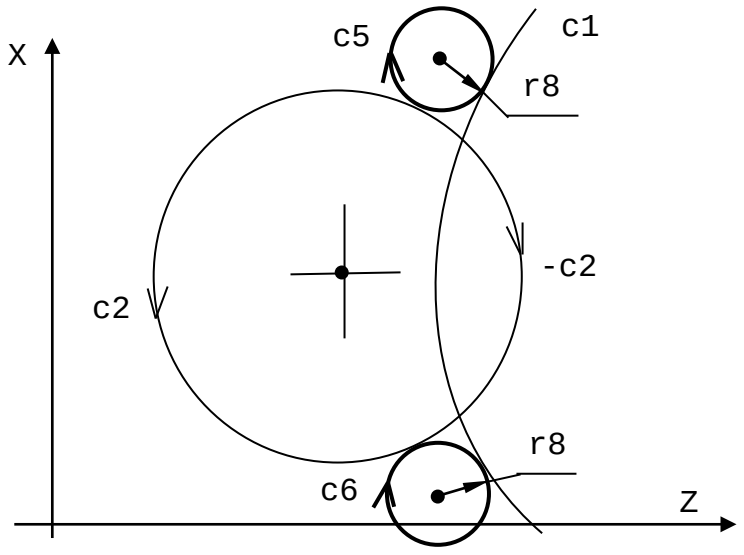


Рисунок А.70

```
c9=-c2,c1,r-8
c10=c1,-c2,r-8
```

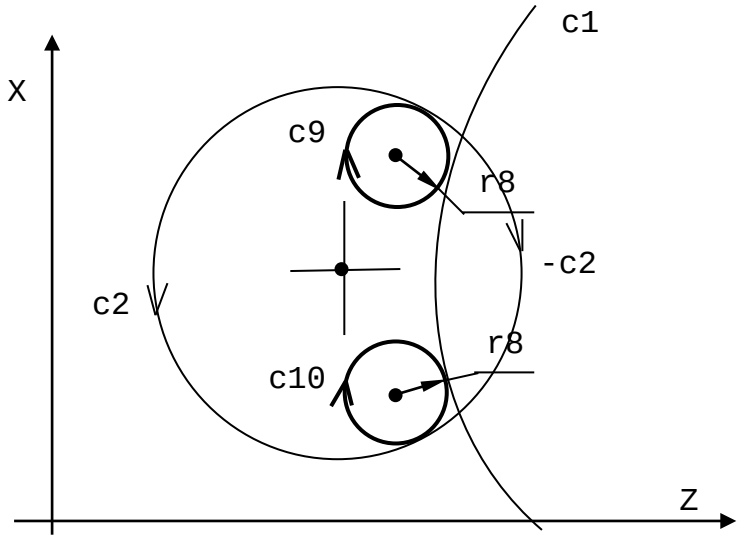


Рисунок А.71

```
c2=c1,p1,r60
c3=p1,c1,r60
```

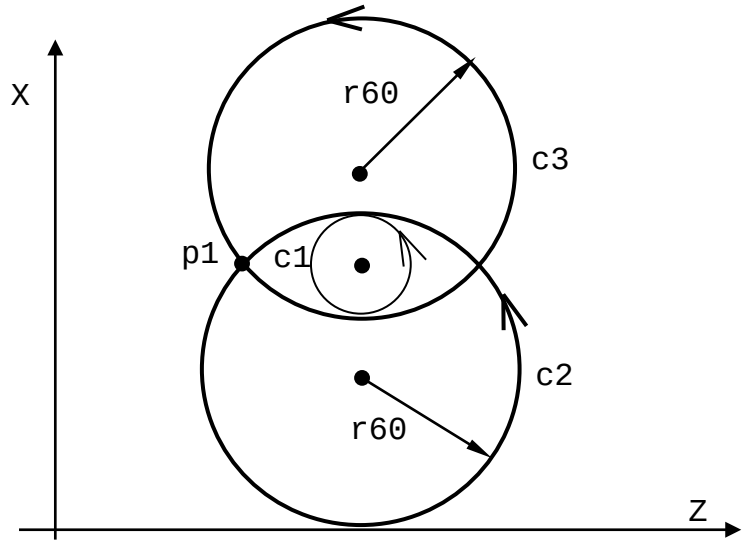


Рисунок А.72

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

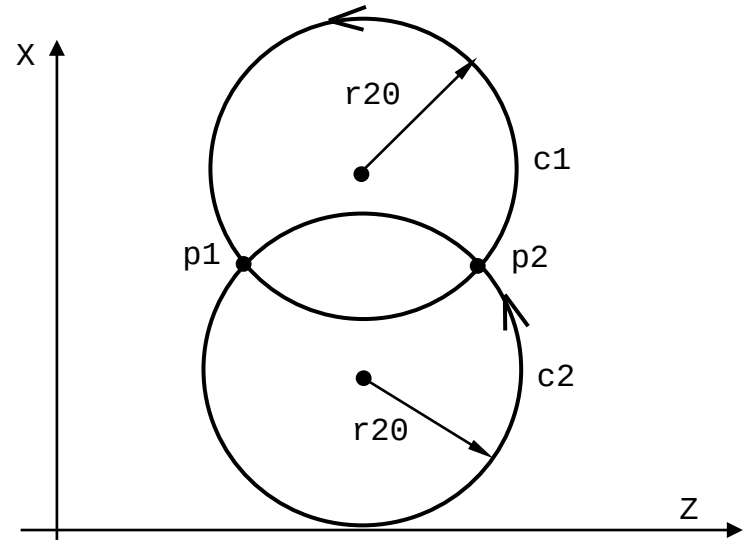


Рисунок А.73

```
c3=p1,l2
```

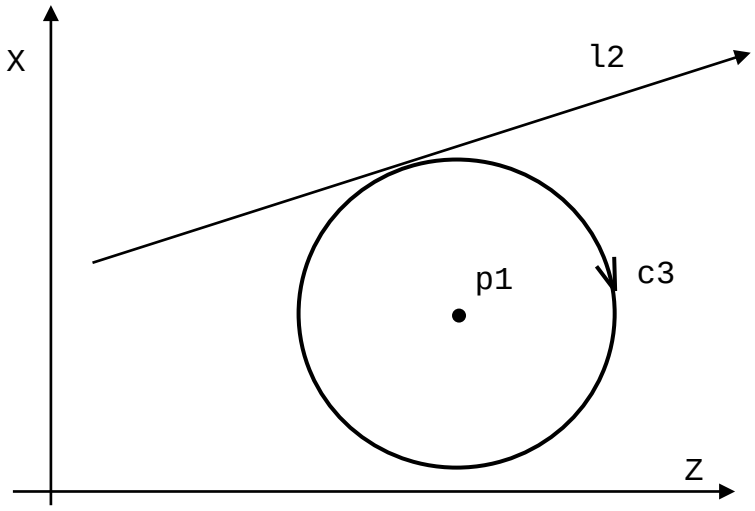


Рисунок 74

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

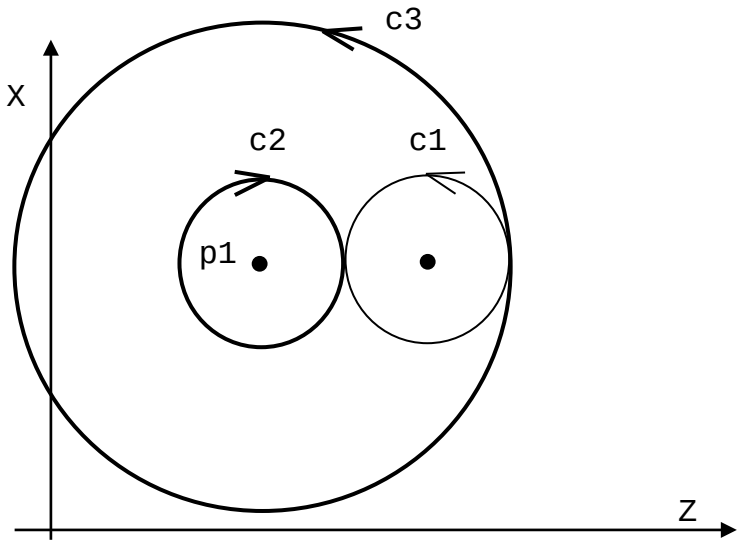


Рисунок А.75

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

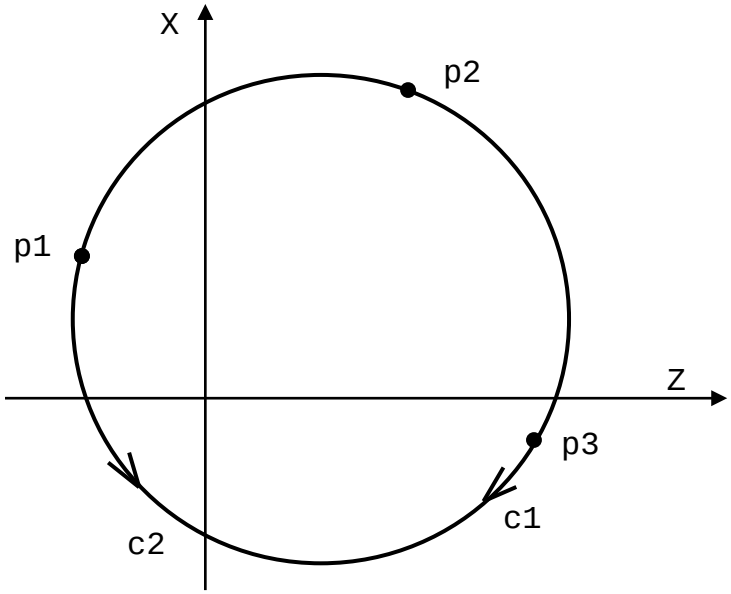


Рисунок А.76

c1=p1,r-40

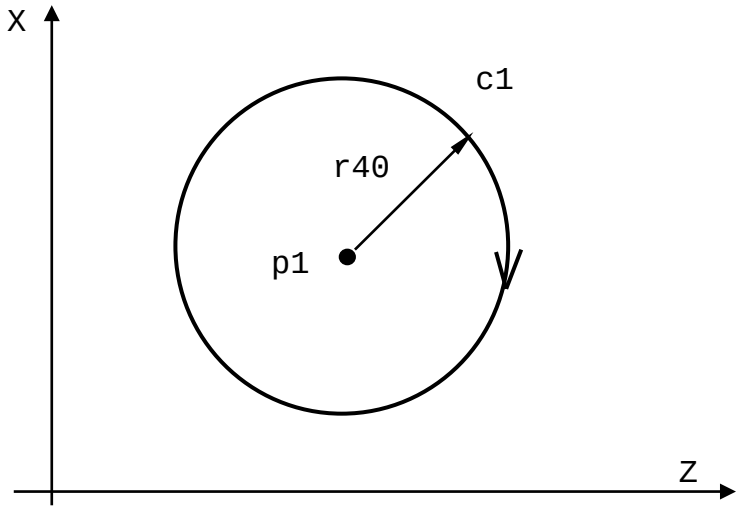


Рисунок А.77

c6=c5,d-10

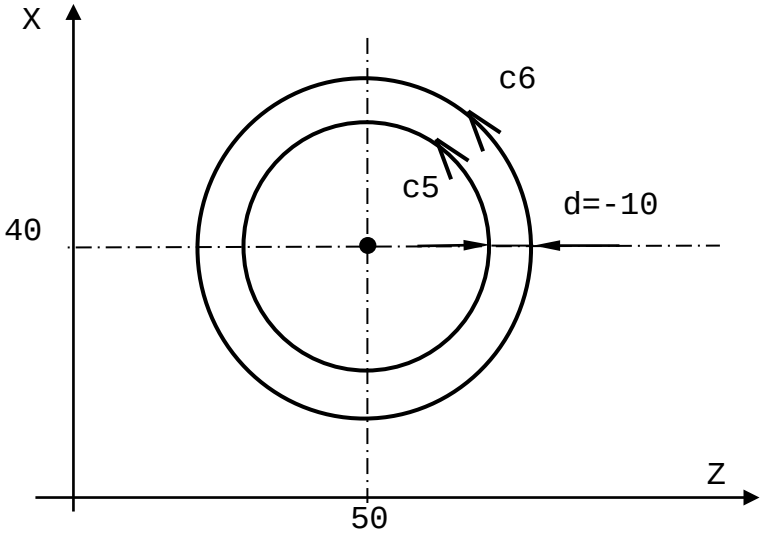


Рисунок А.78

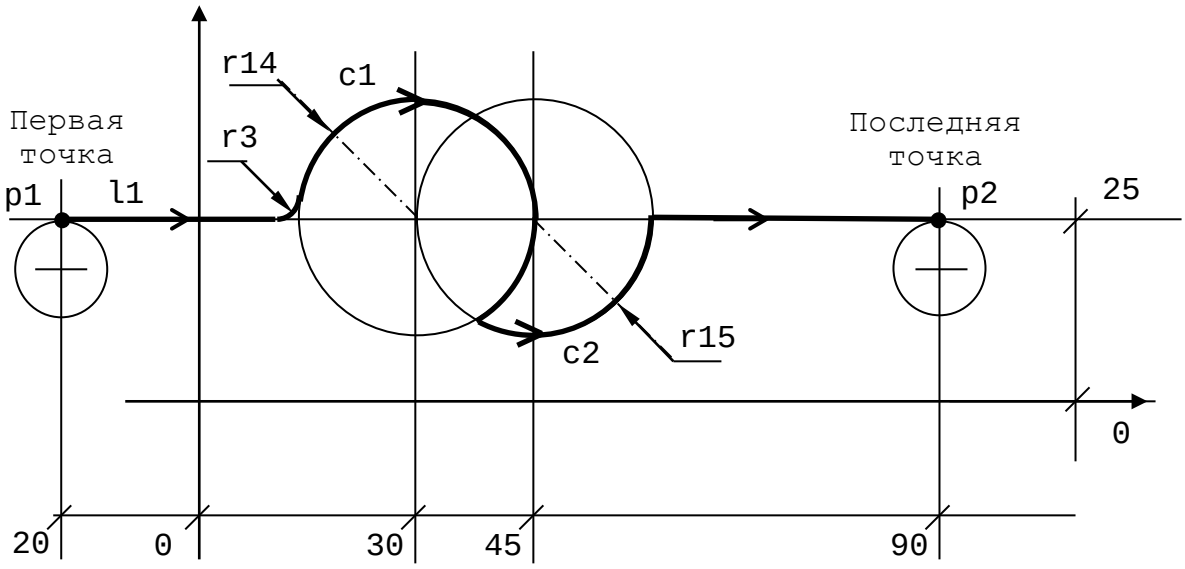
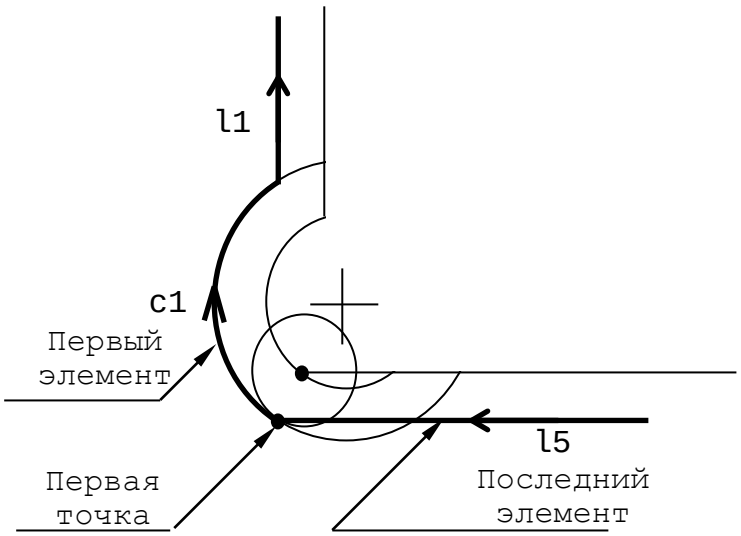


Рисунок А.79




```
..............
l5=ZX-15,a180
..............
l1=Z-30X-15,a135
..............
          G21G42l5
l1
..............
l5
G20G40l1
```

Рисунок А.80

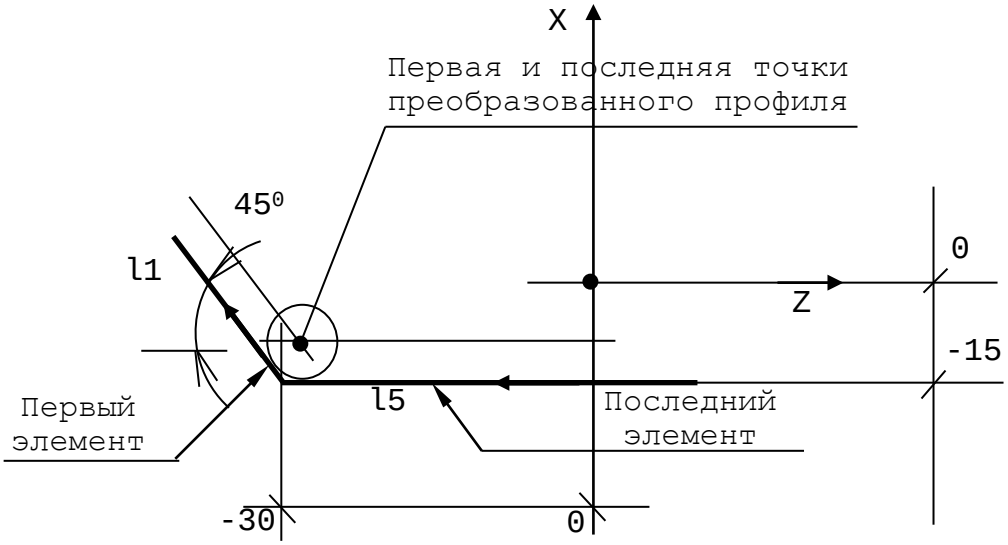


Рисунок А.81

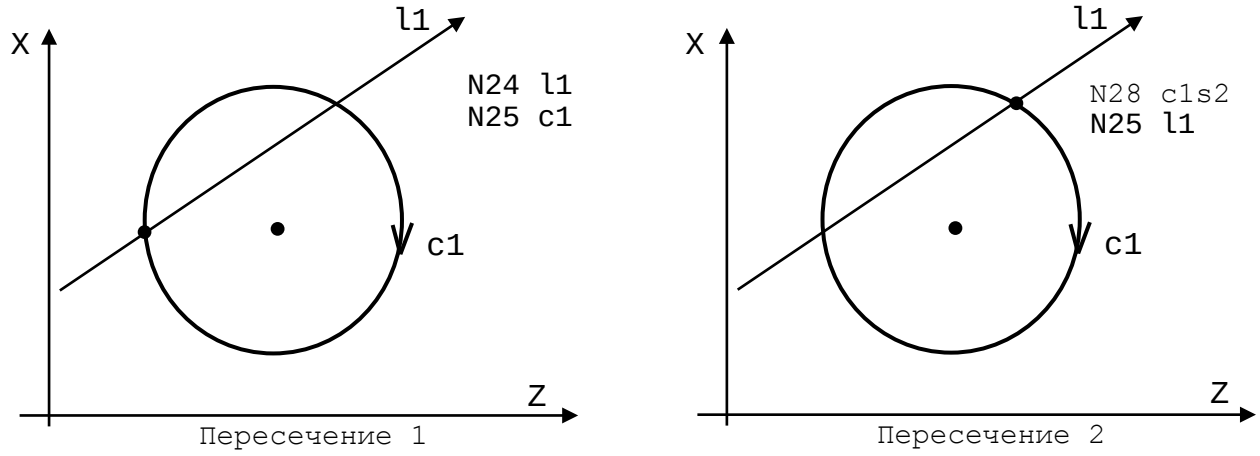


Рисунок А.82

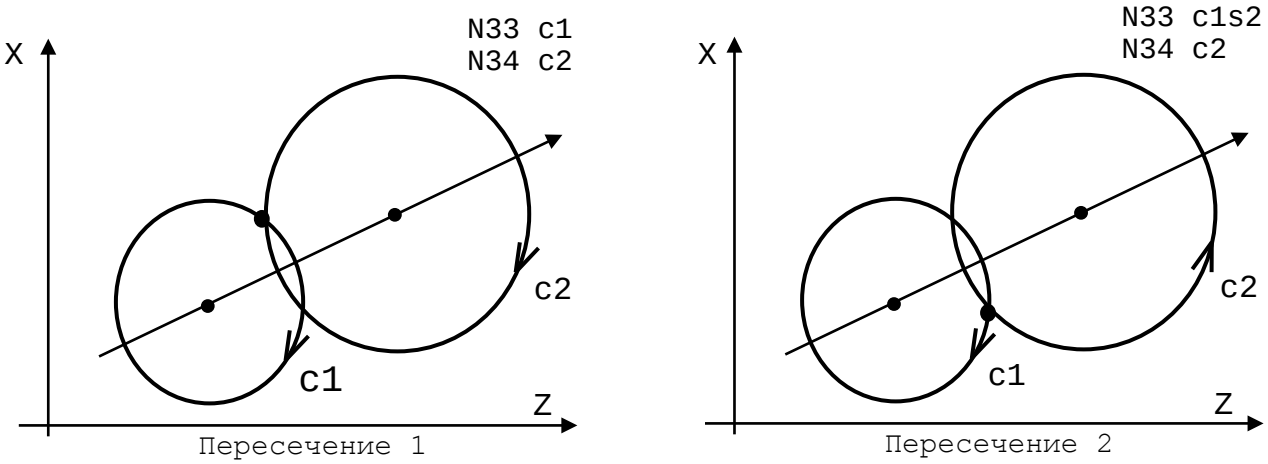


Рисунок А.83

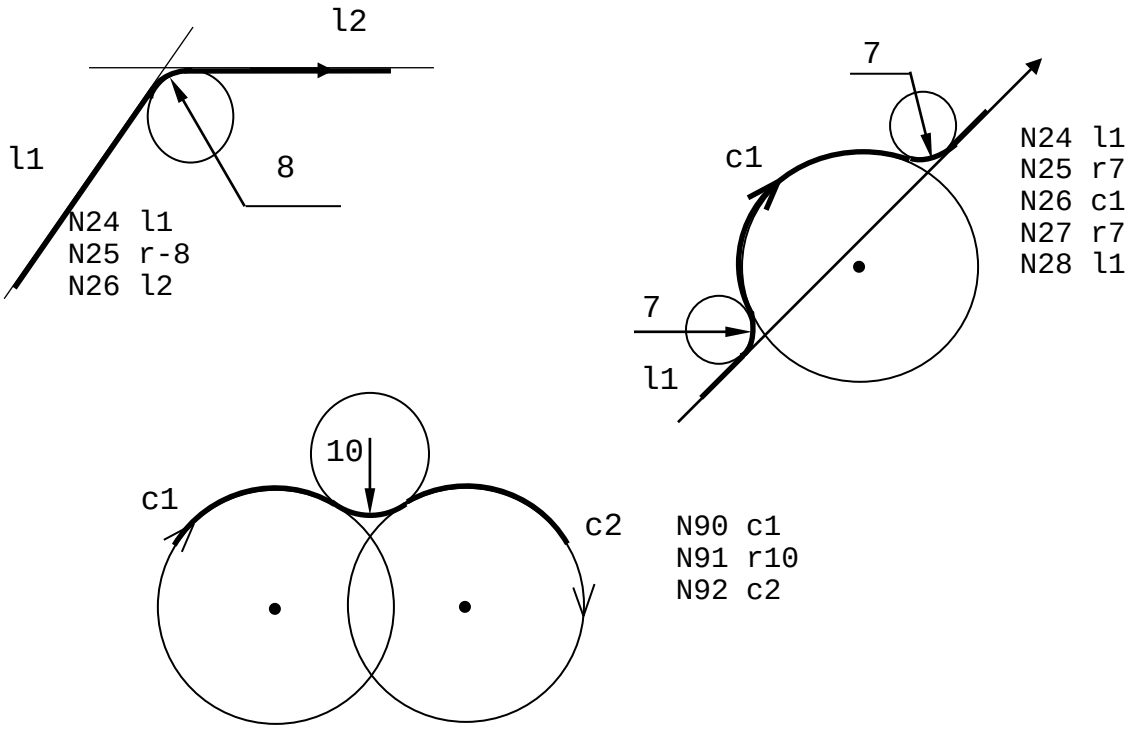


Рисунок А.84

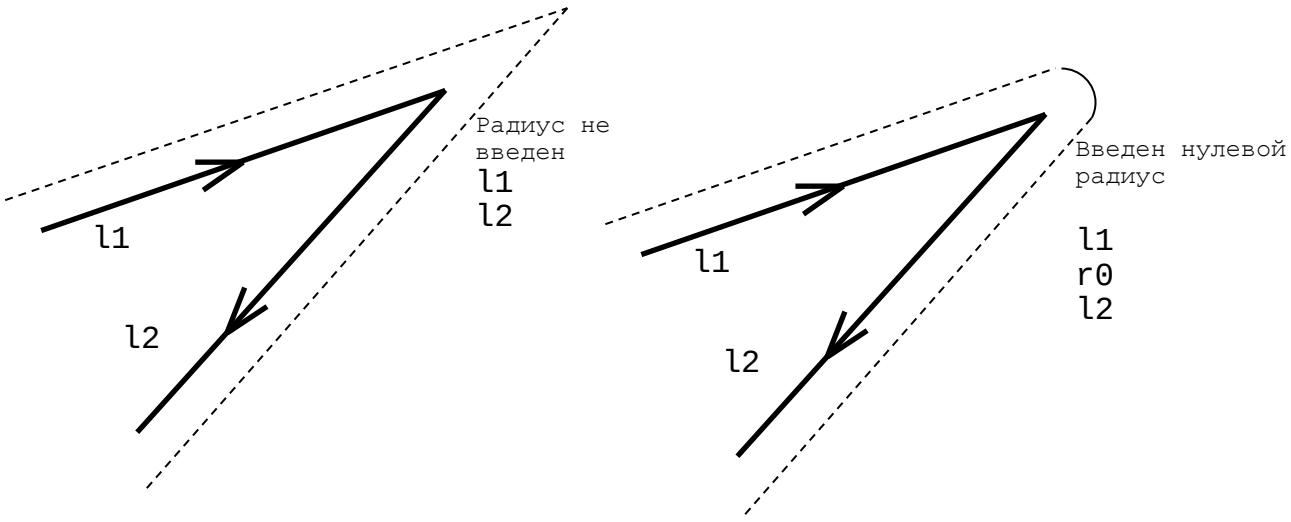


Рисунок А.85

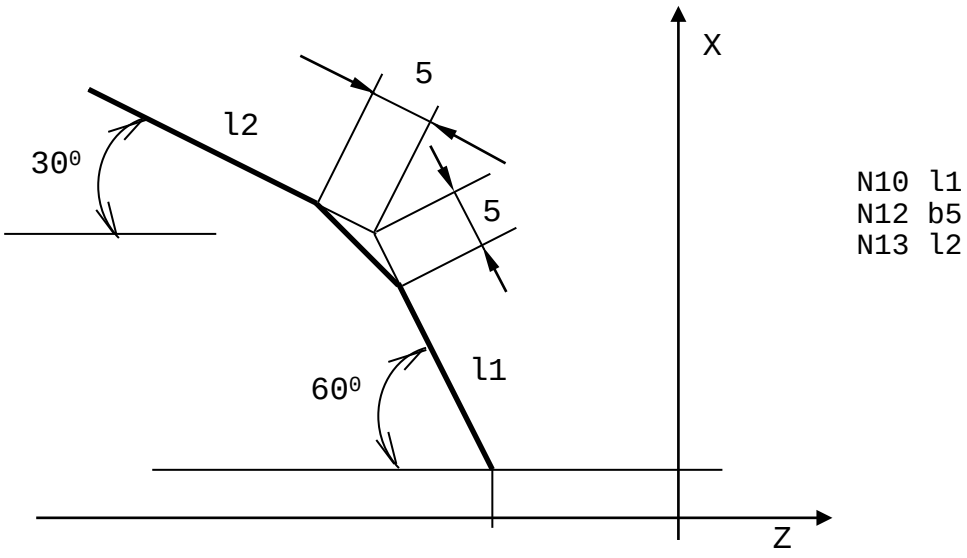


Рисунок А.86

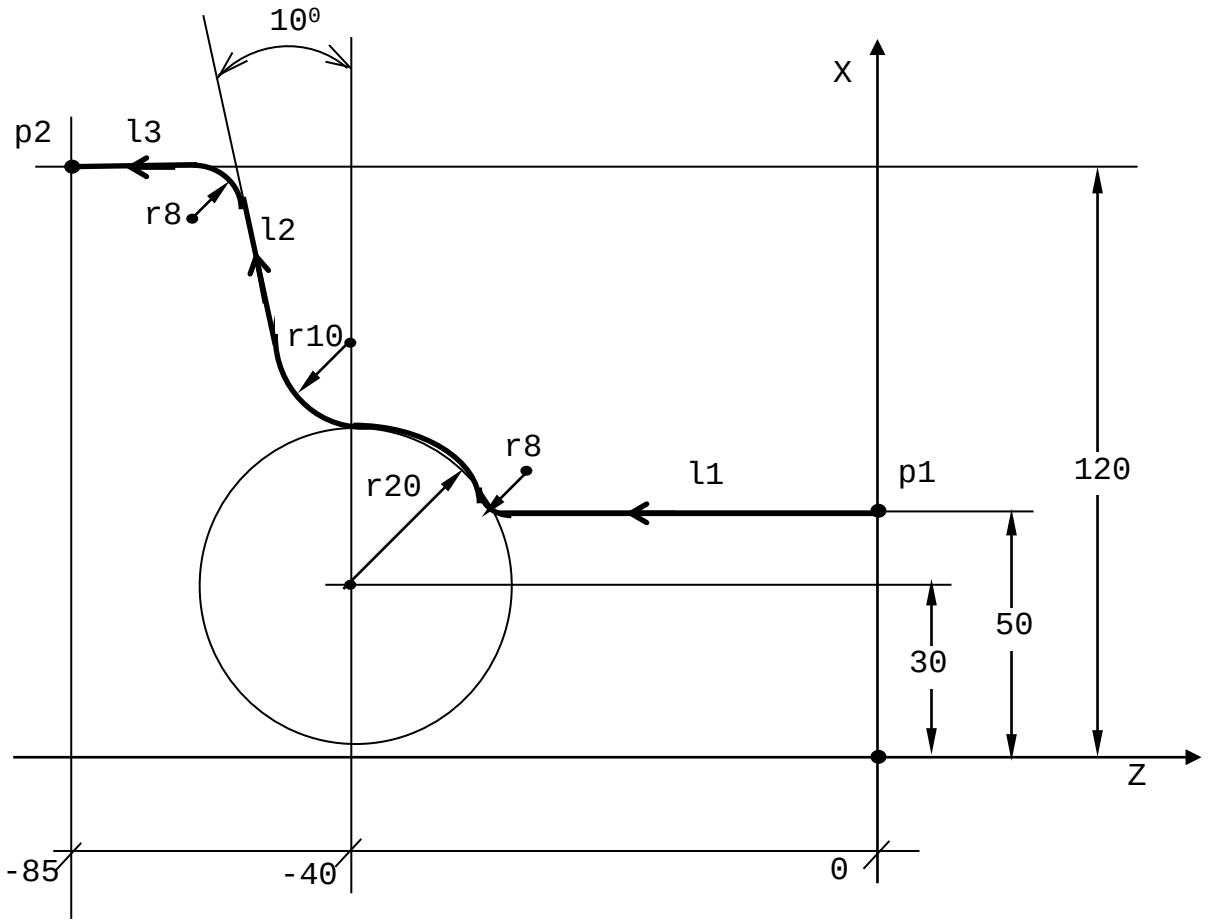


Рисунок А.87

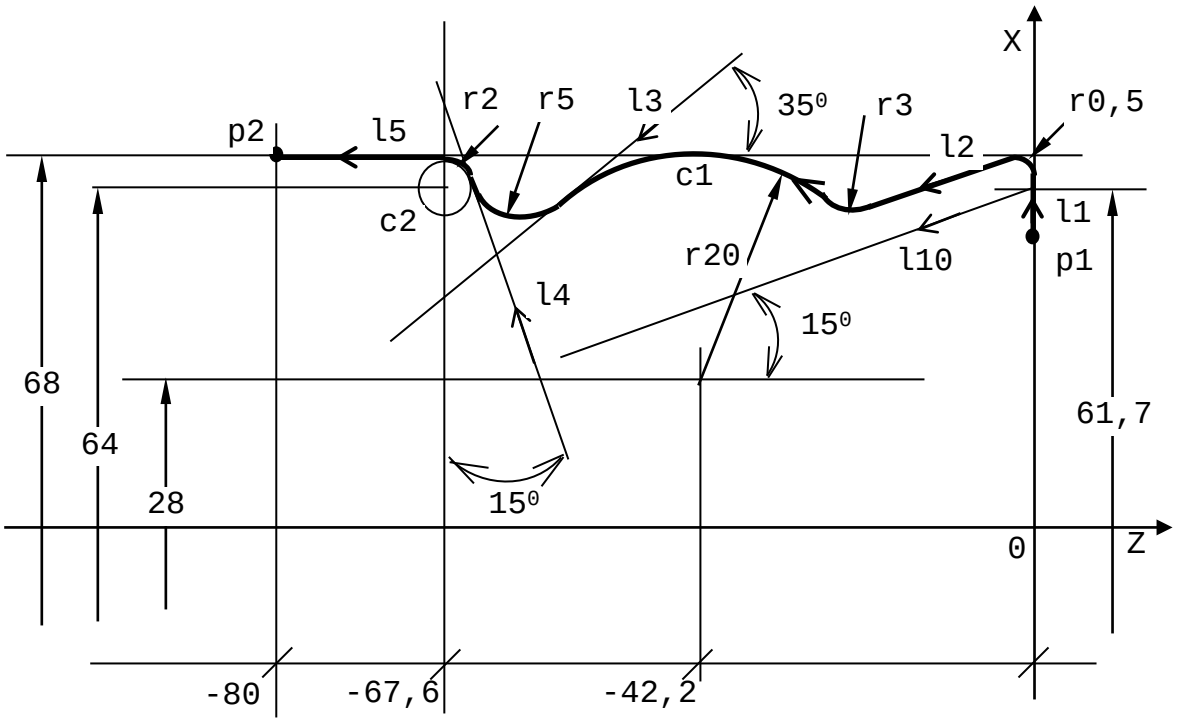


Рисунок А.88

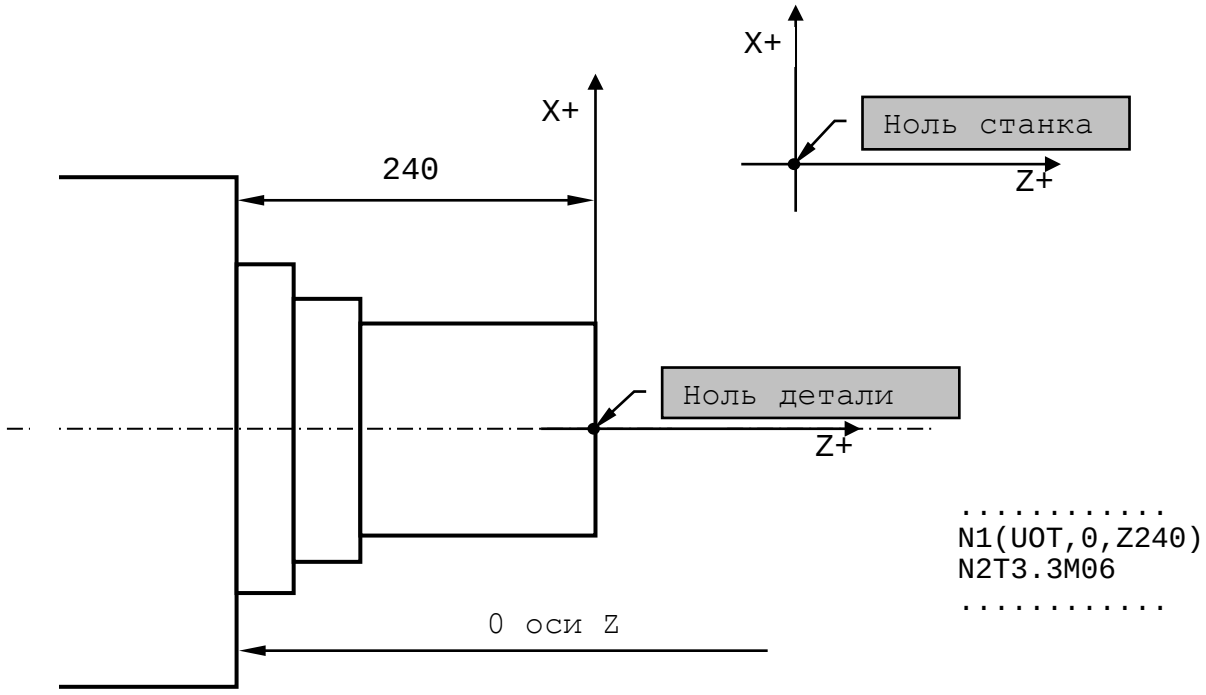


Рисунок А.89

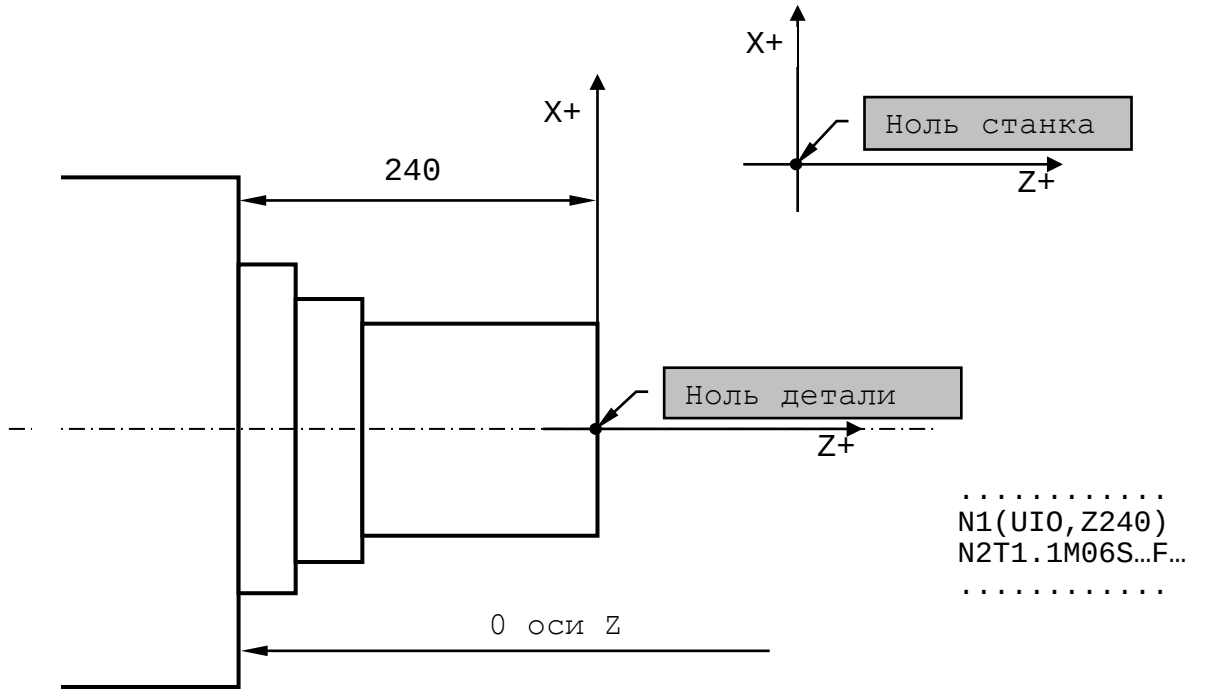


Рисунок А.90

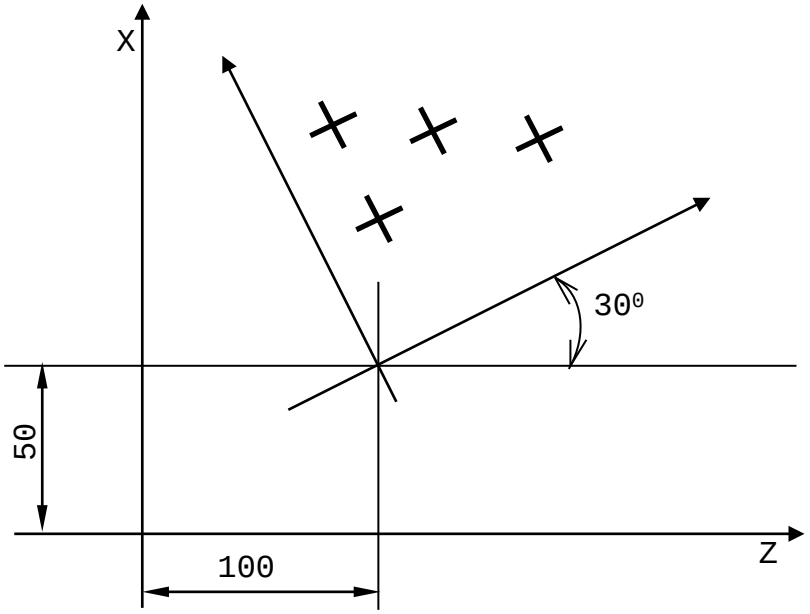


Рисунок А.91

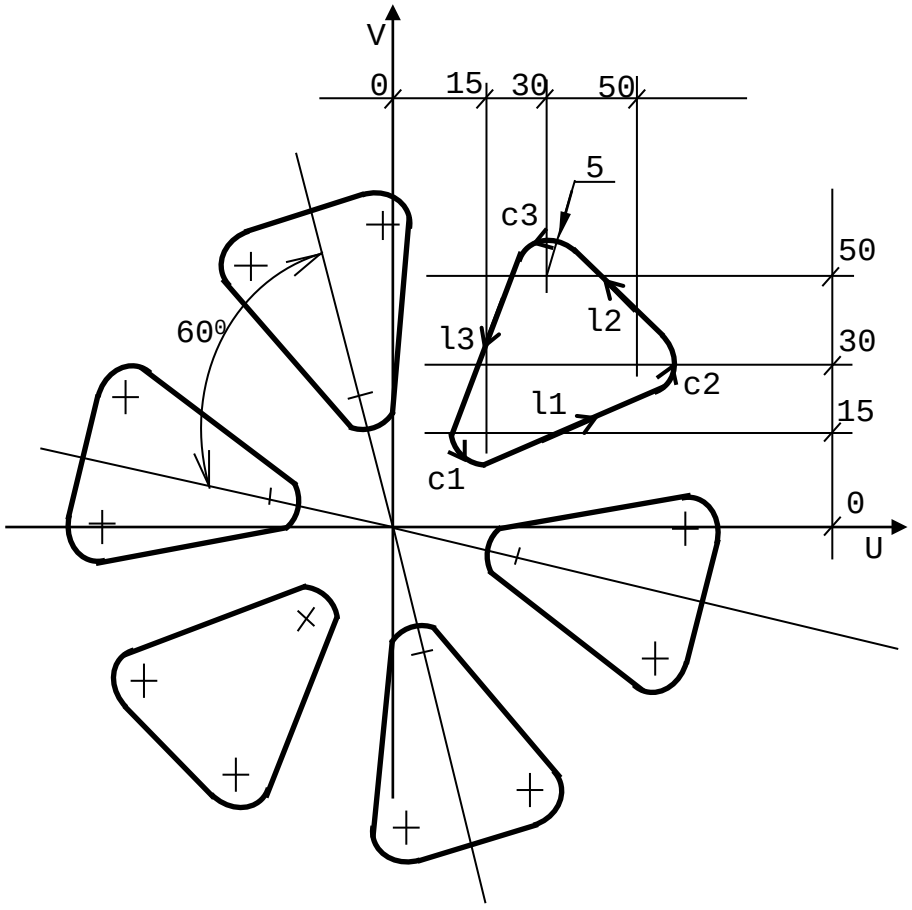


Рисунок А.92

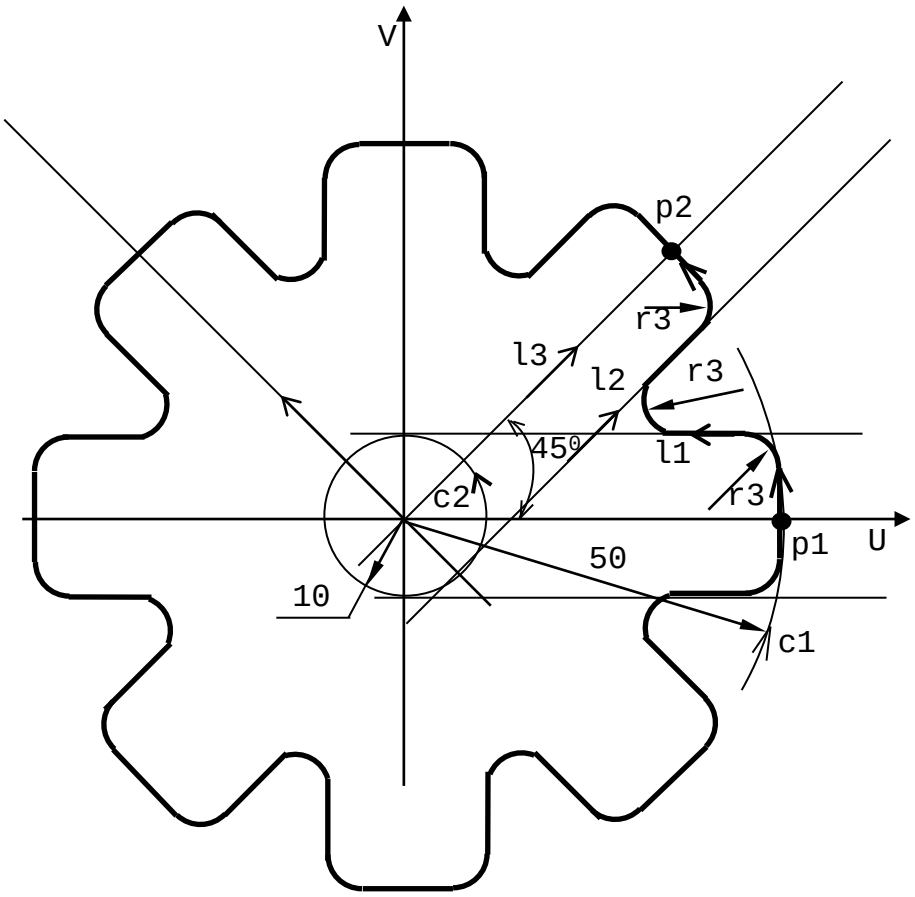
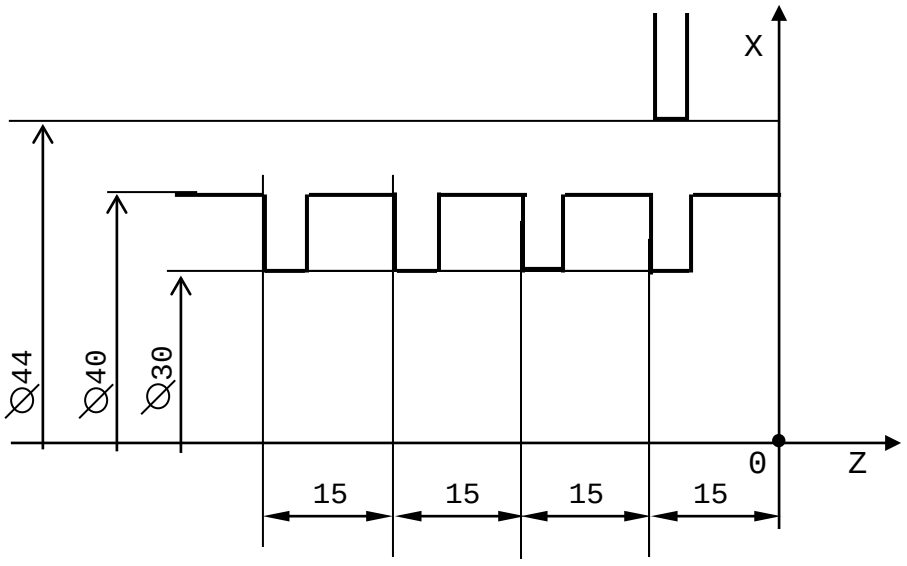
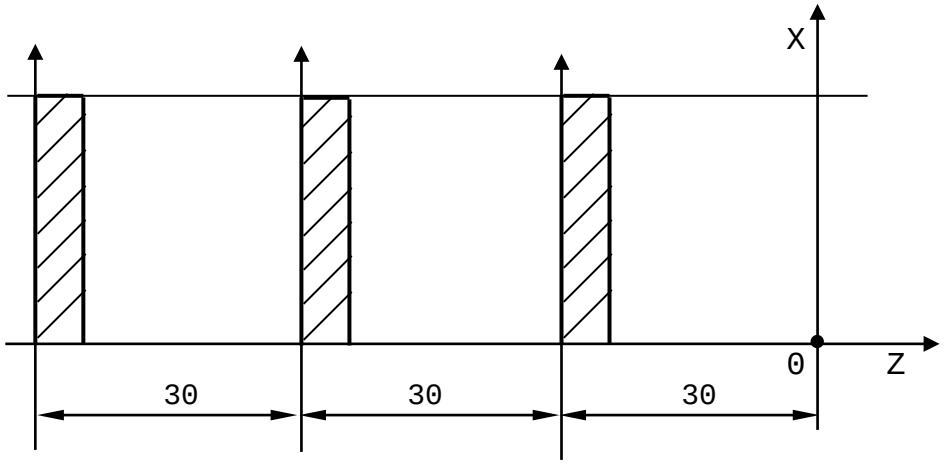


Рисунок А.93




```
N14 G00 X44 Z0
N15 (RPT,3)
N16 G91 Z-15
N17  G90  G1  G4
X30 F0.2
N18 G00 X44
N19 (ERP)
```

Рисунок А.94




```
.....
(RPT,3)
.....
.....
(UIO,Z-30)
(ERP)
M30
```

Рисунок А.95

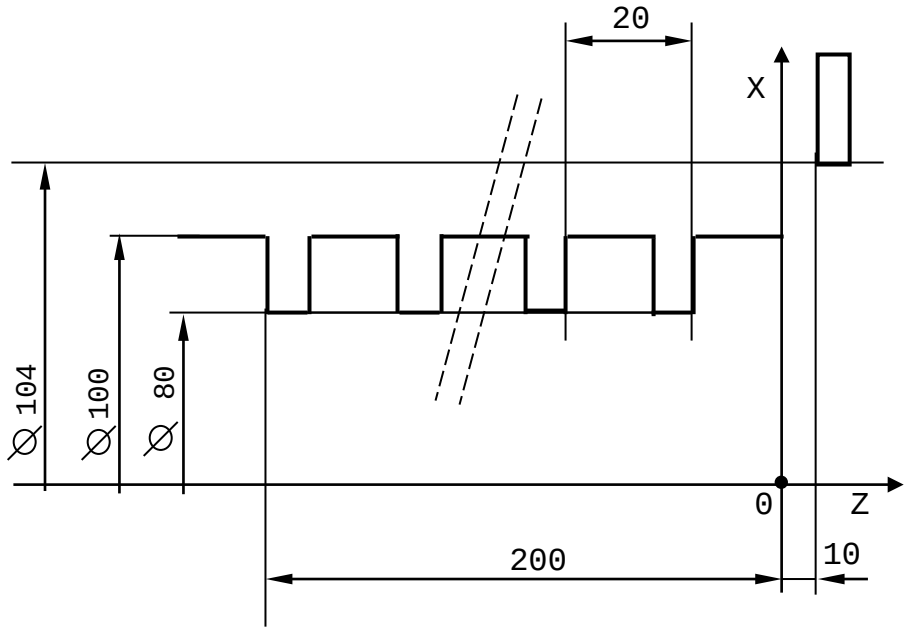
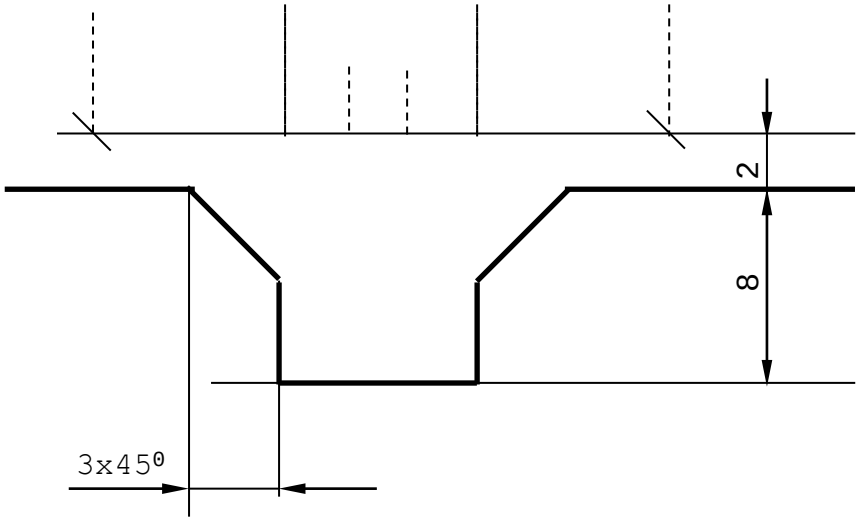
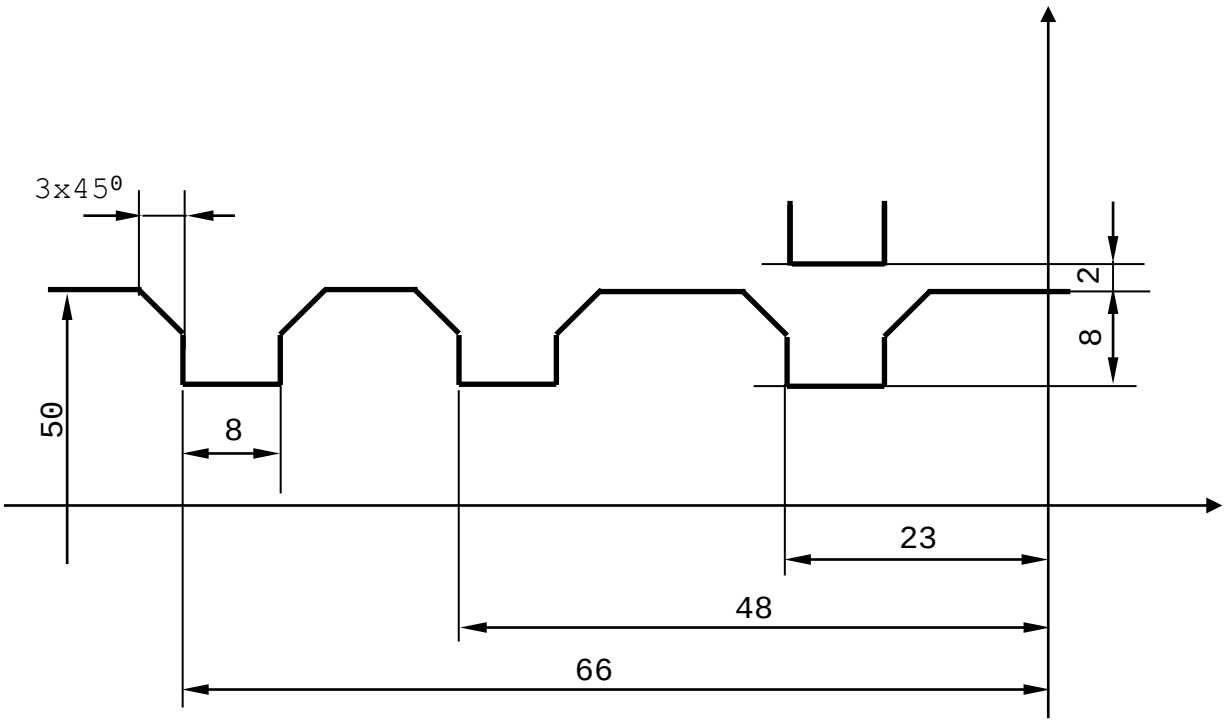


Рисунок А.96




```
N90 G91 G1 G4 X-
20 F0.1
N91 G0 X20
N92 Z-5
N93 G1 X-10 Z-5
N94 G0 X10
N95 G1 X-10 Z-5
N96 G0 X10
N97 G90
```

Рисунок А.97 - Определение цикла обработки паза




```
N17 TMR=2
N18 G0 X54 Z-23
N19 (CLS,P90)
N20 Z-48
N21 (CLS,P90)
```

Рисунок А.98 - Вызов цикла

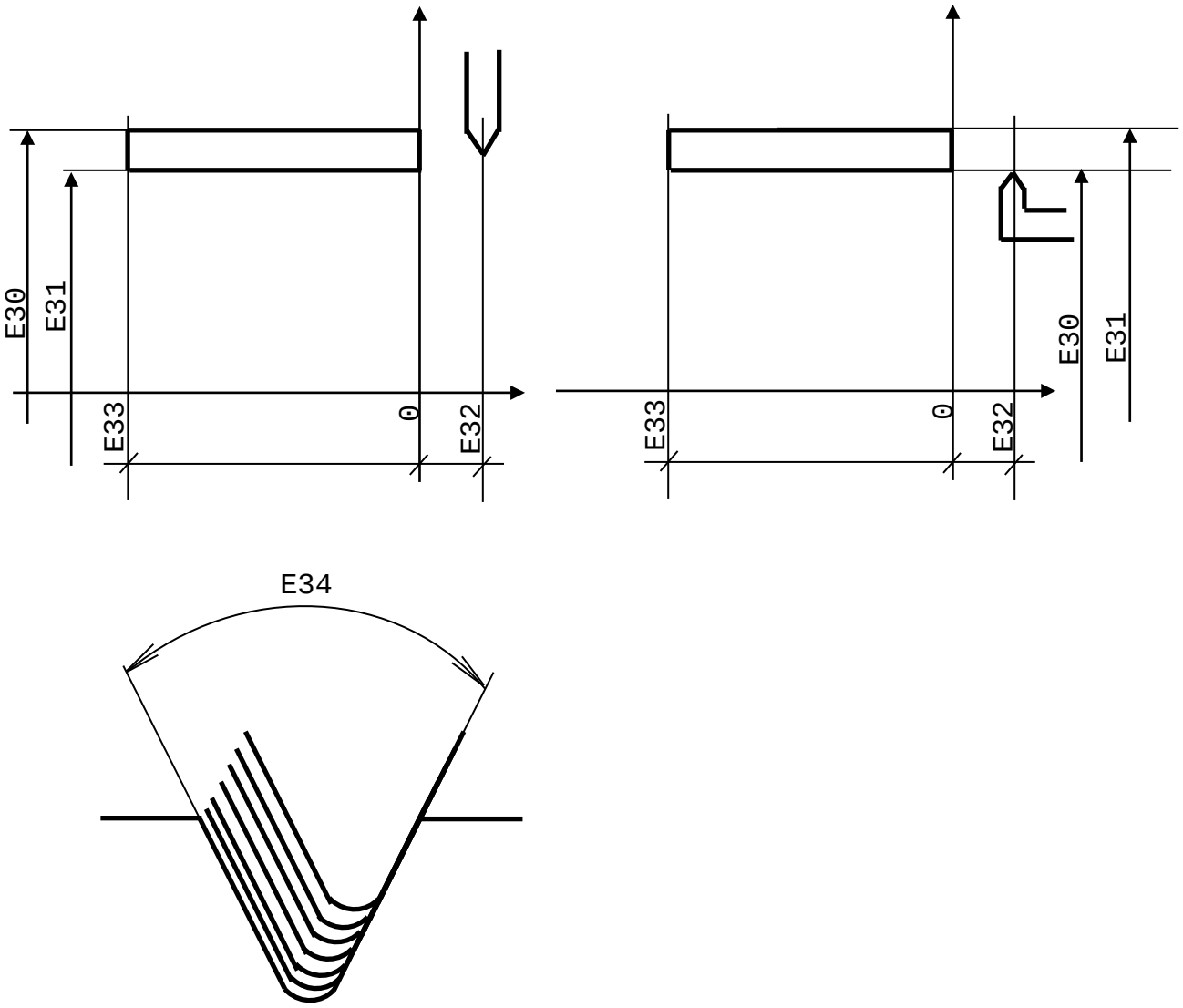


Рисунок А.99

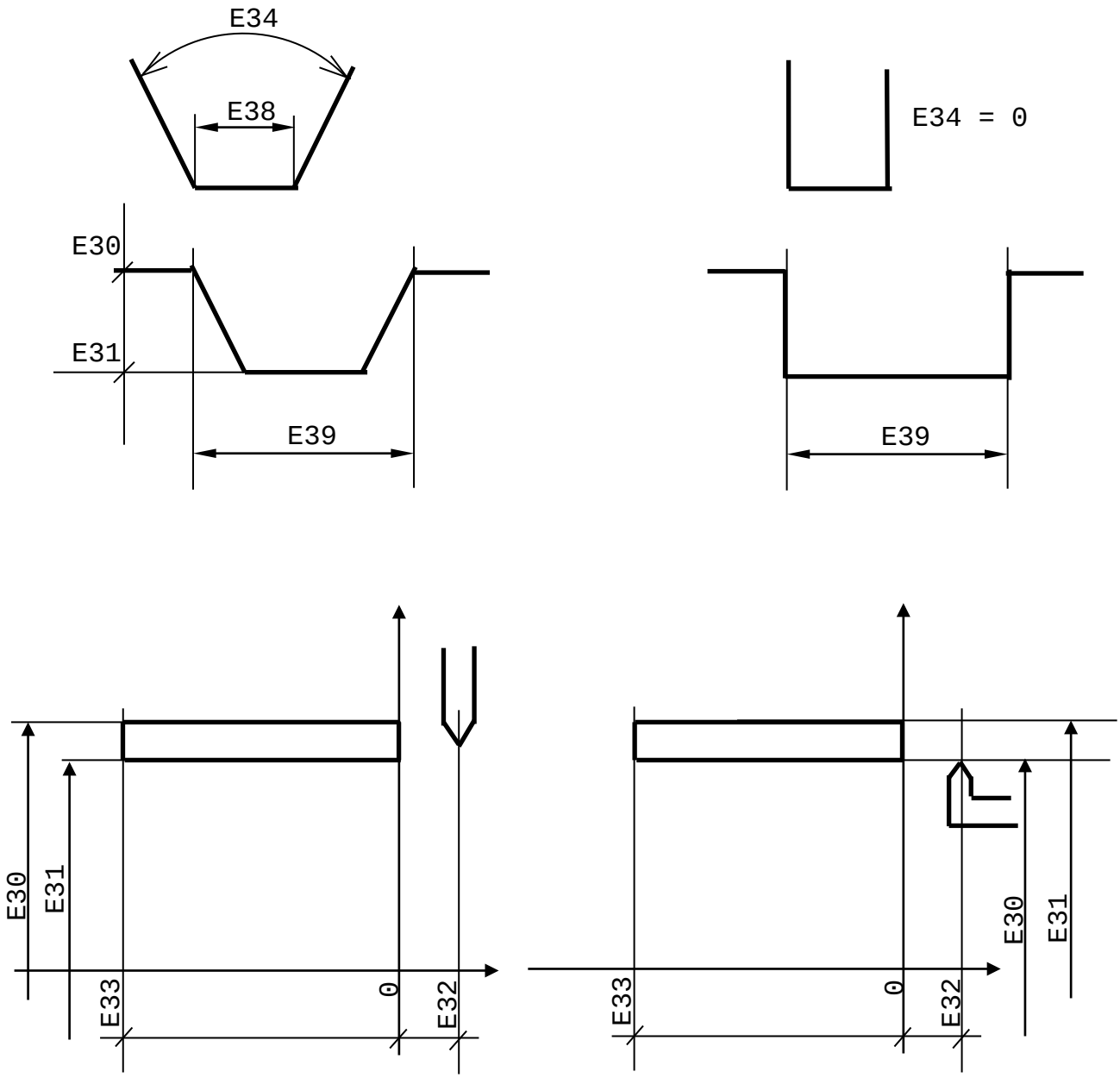


Рисунок А.100

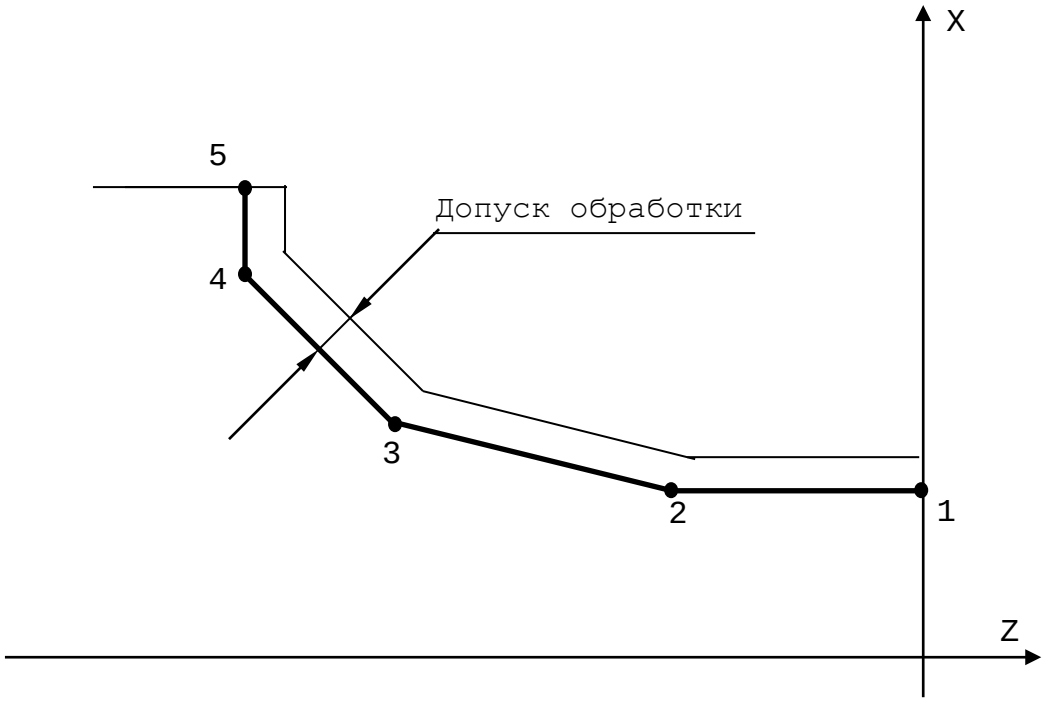


Рисунок А.101

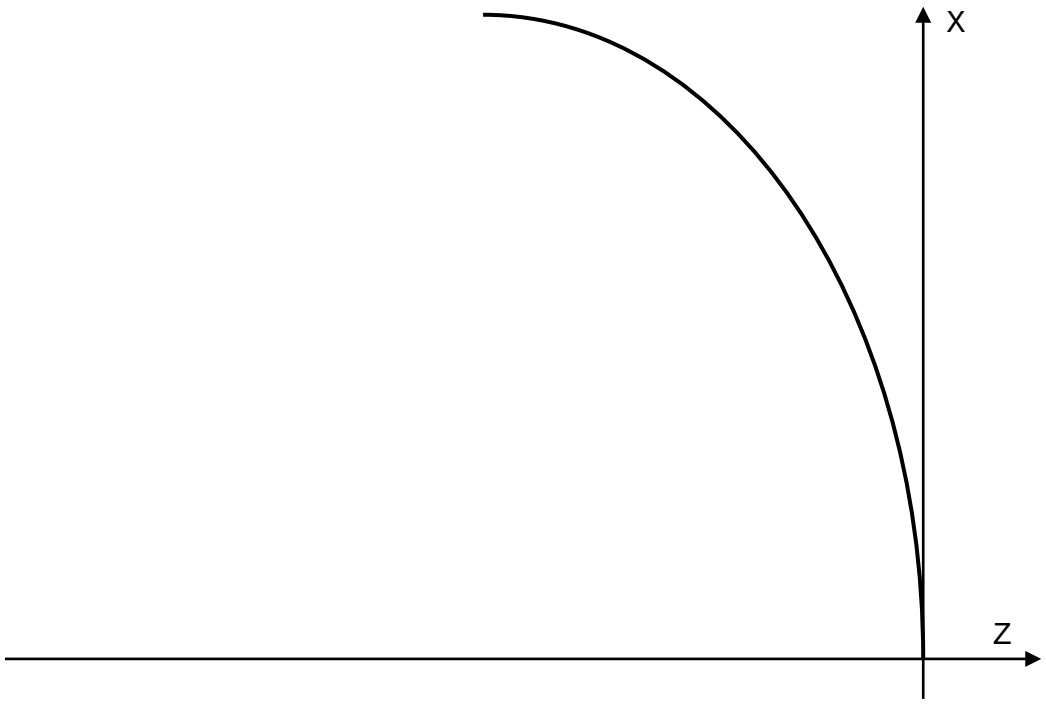


Рисунок А.102

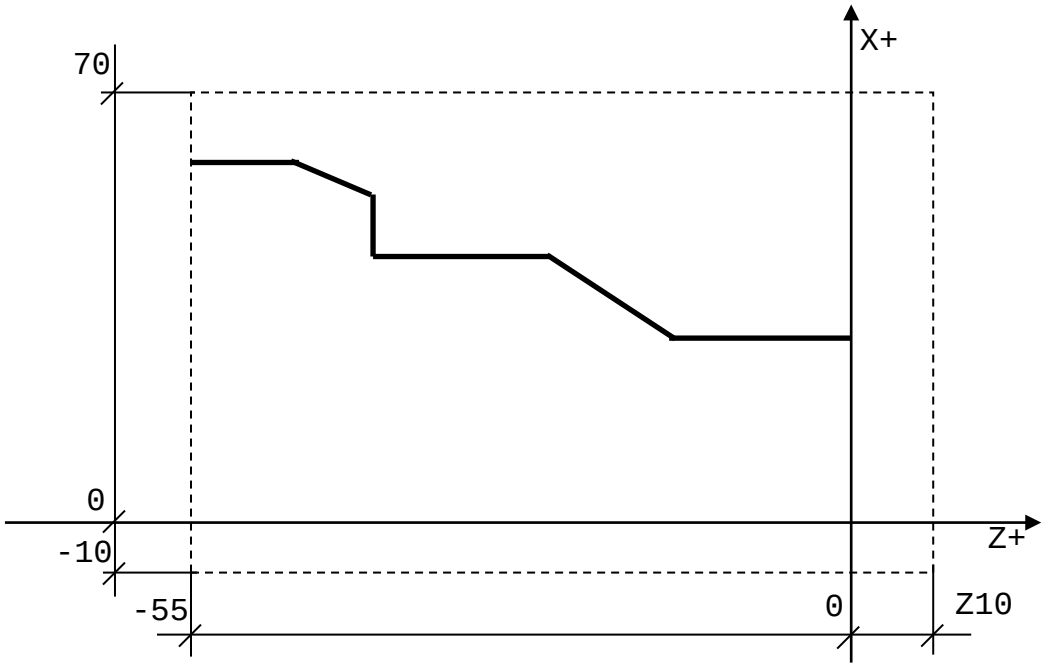


Рисунок А.103

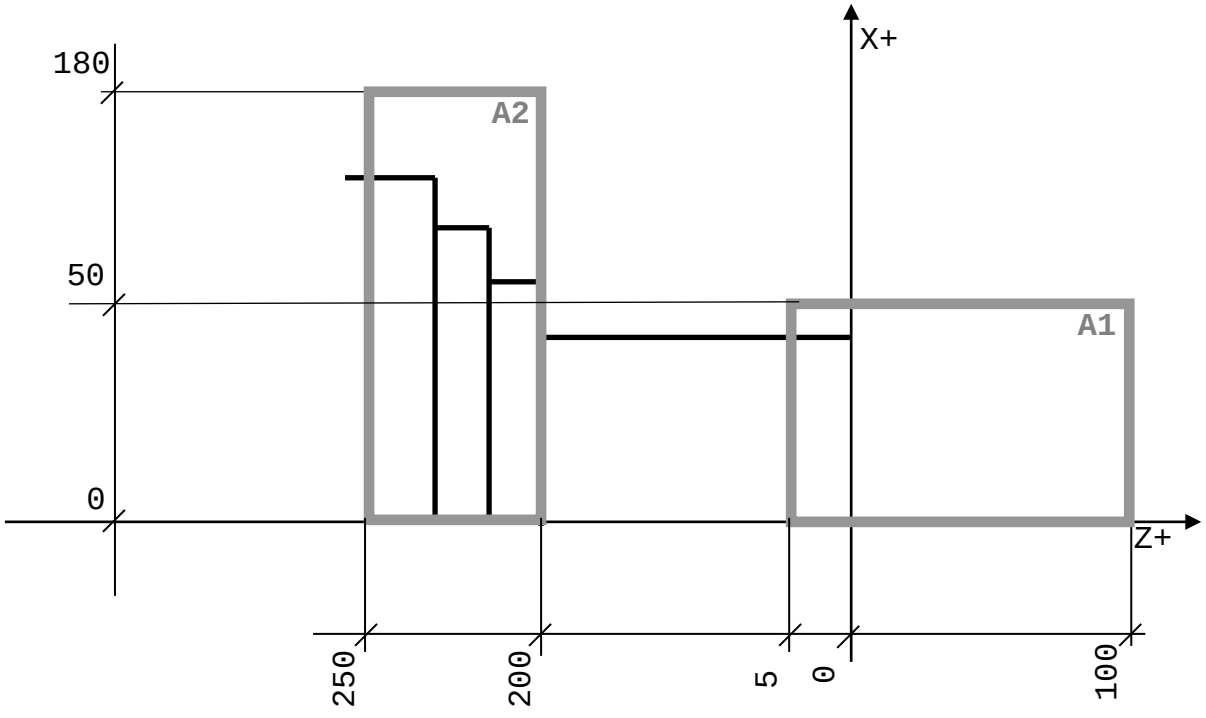


Рисунок А.104

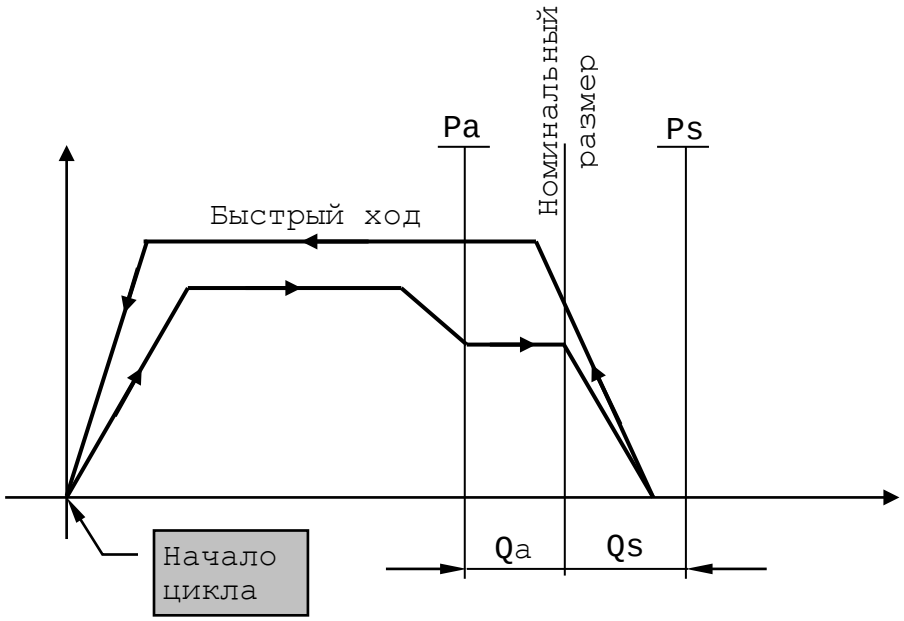


Рисунок А.105

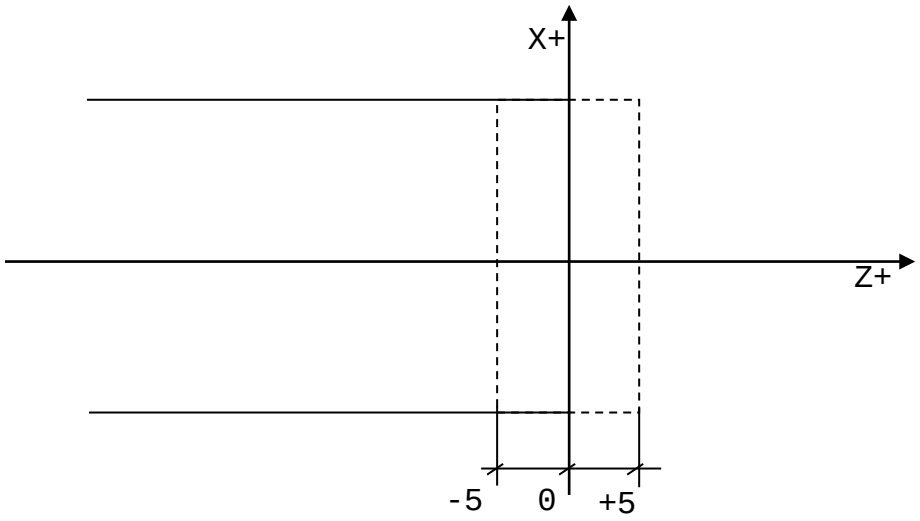


Рисунок А.106

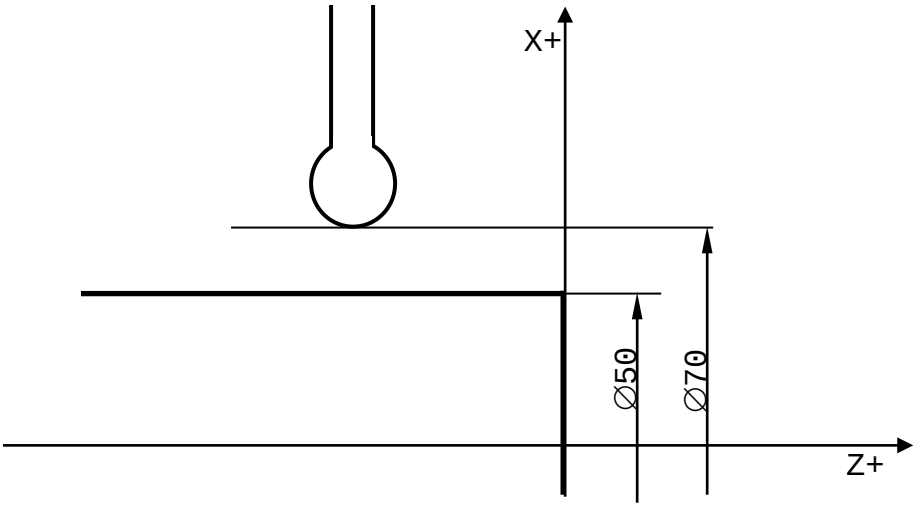


Рисунок А.107

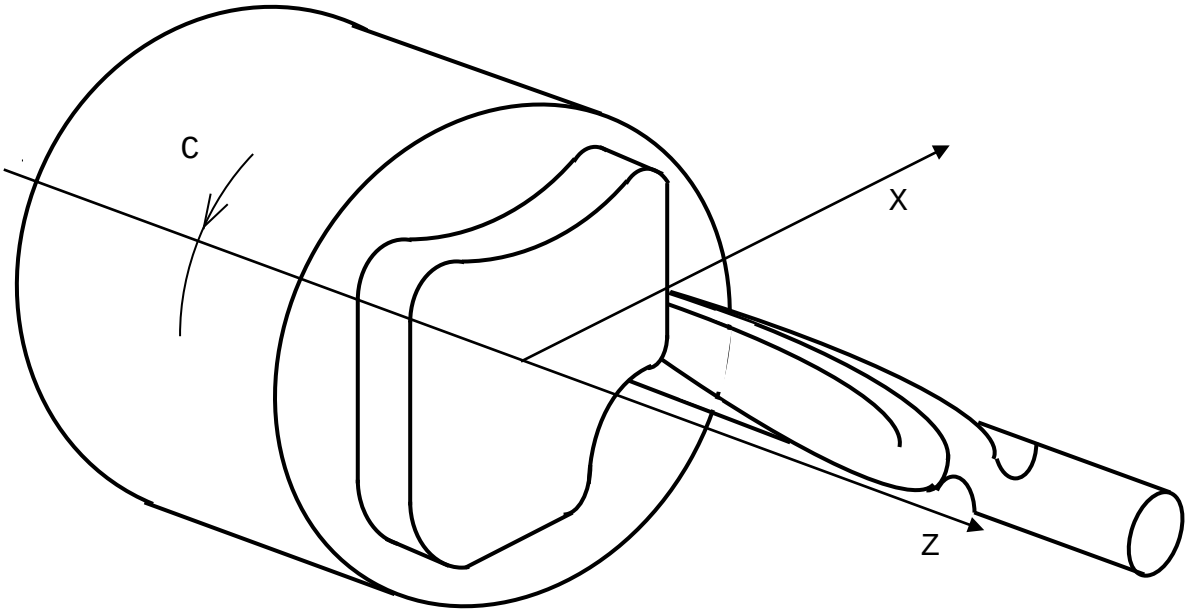


Рисунок А.108

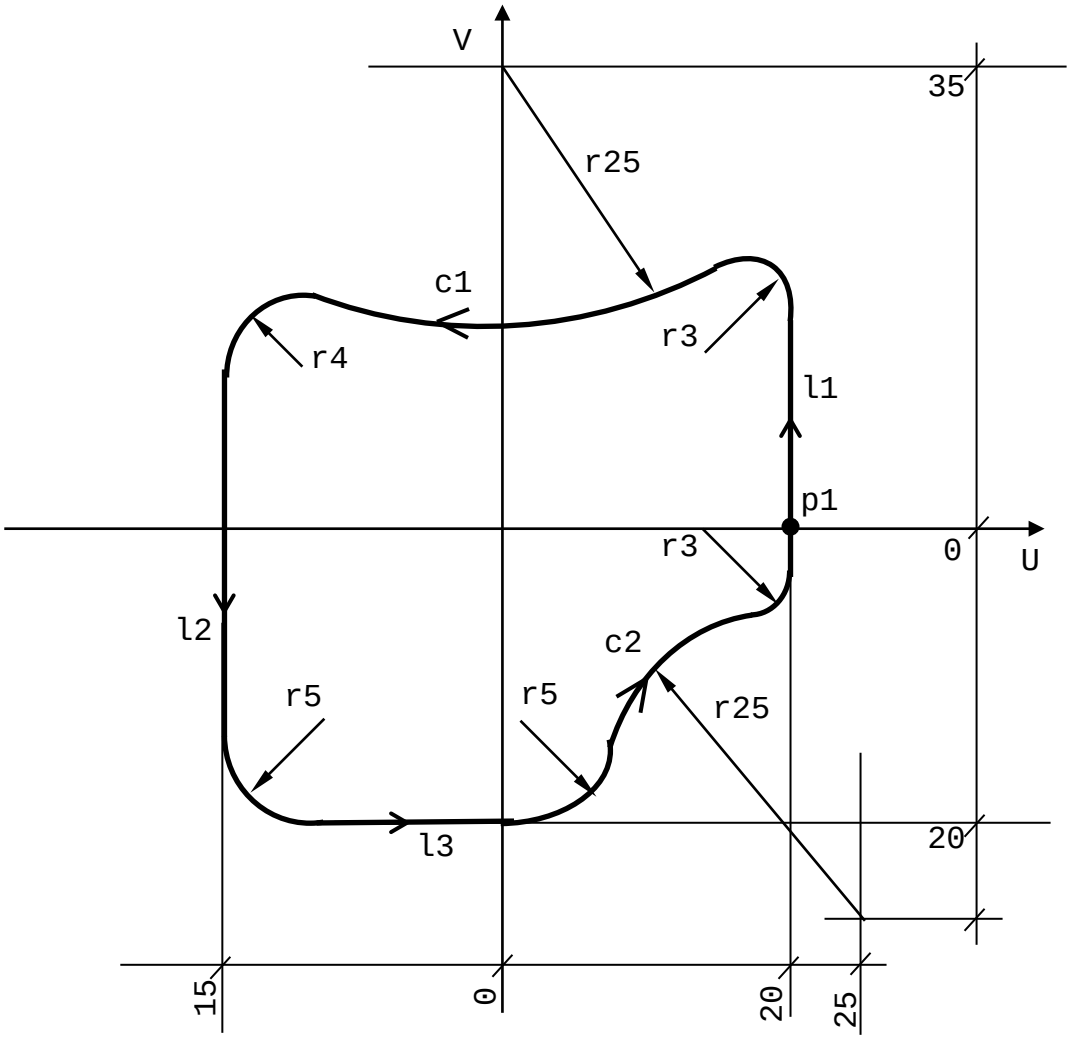


Рисунок А.109

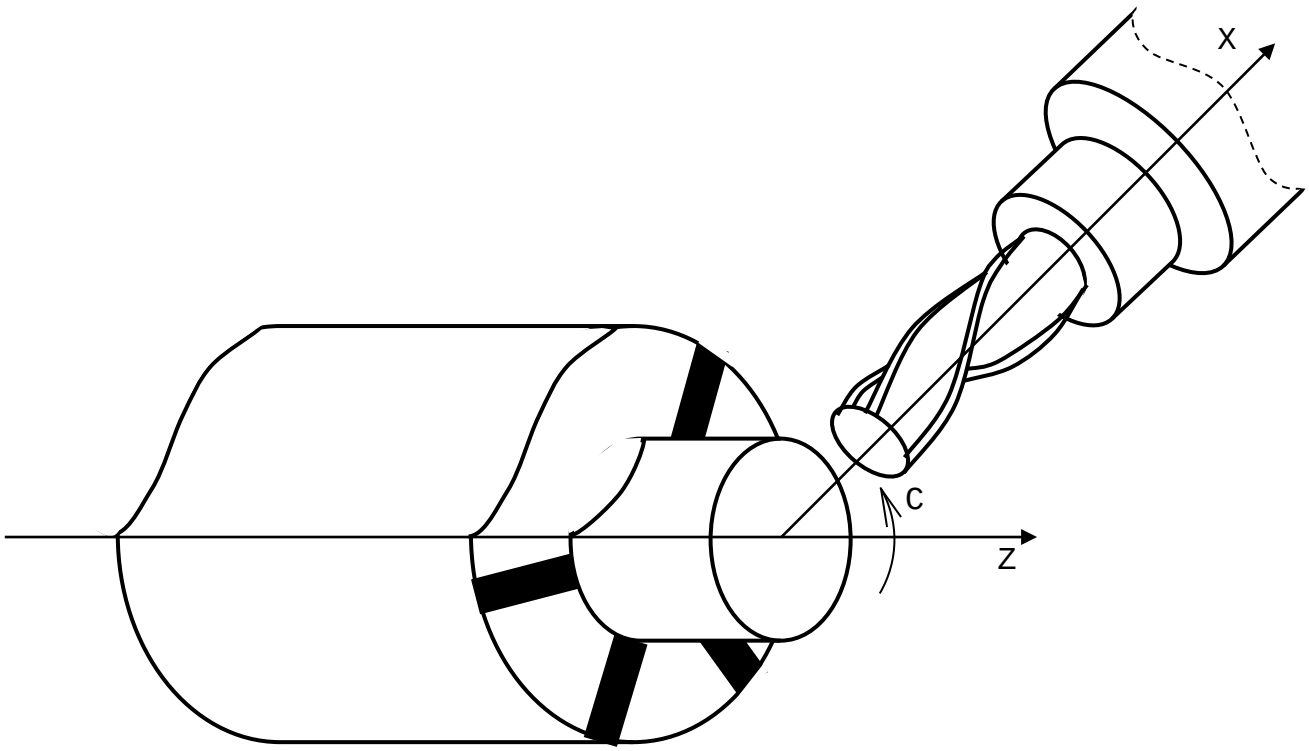


Рисунок А.110

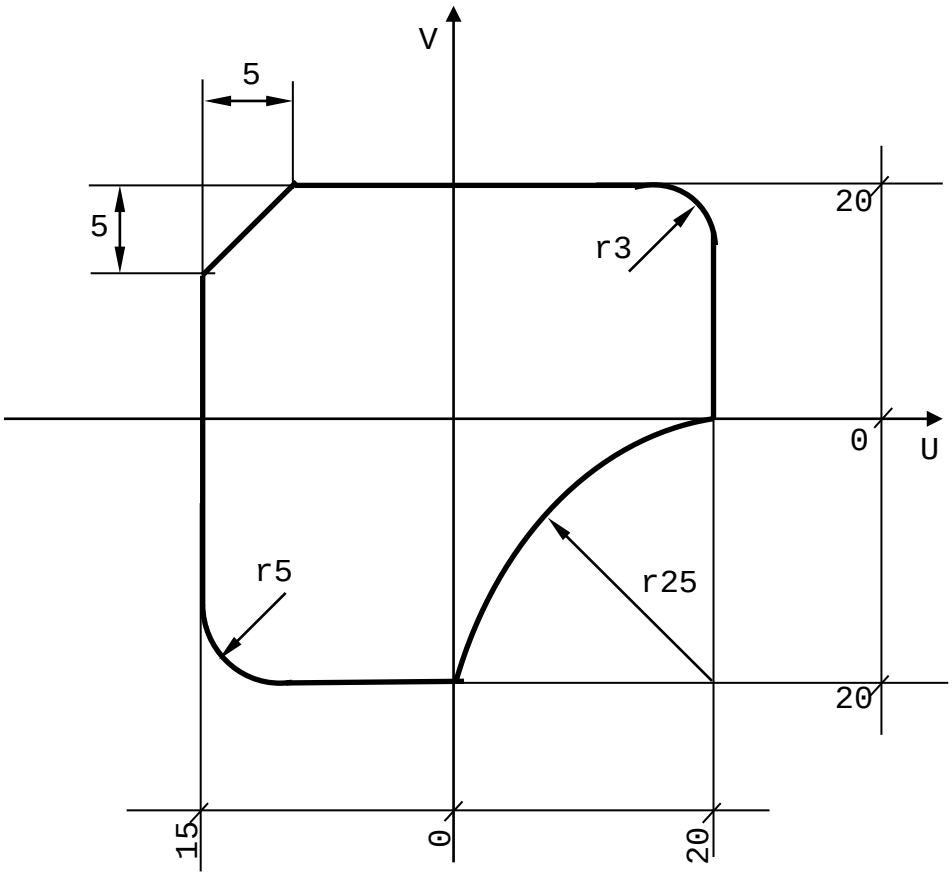


Рисунок А.111

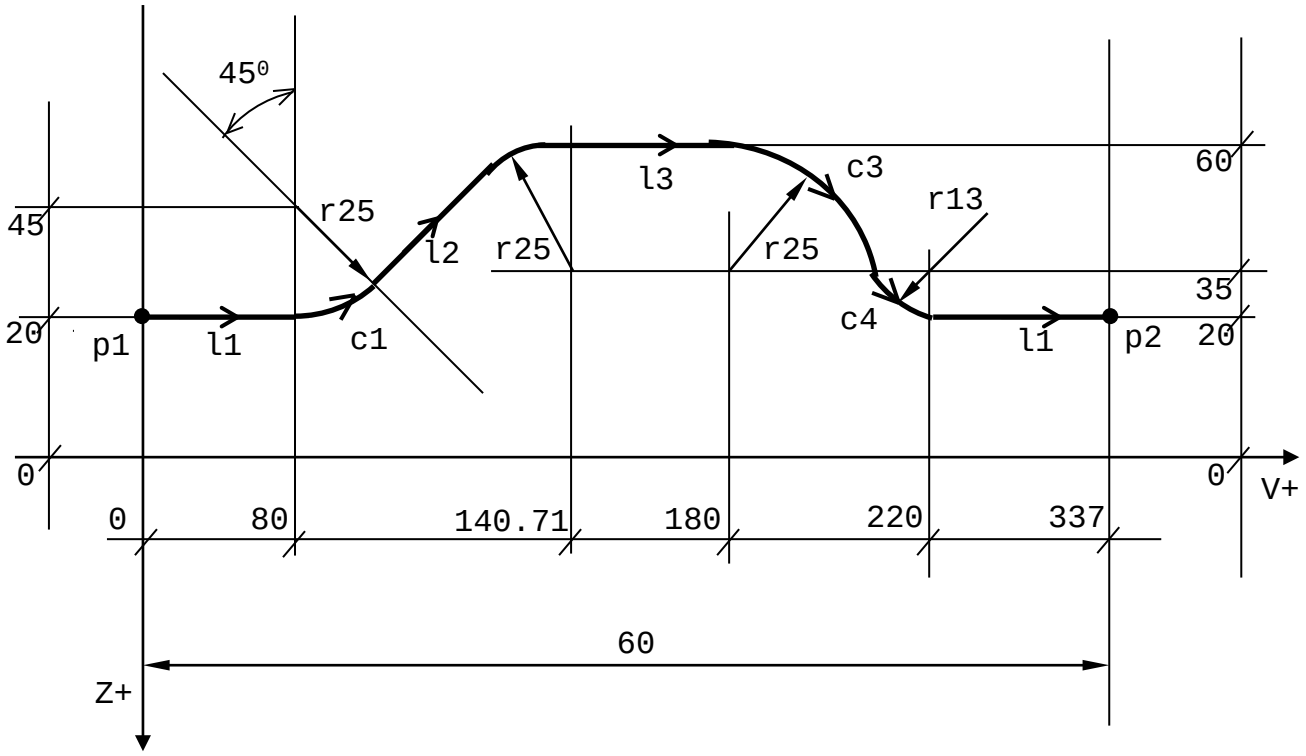
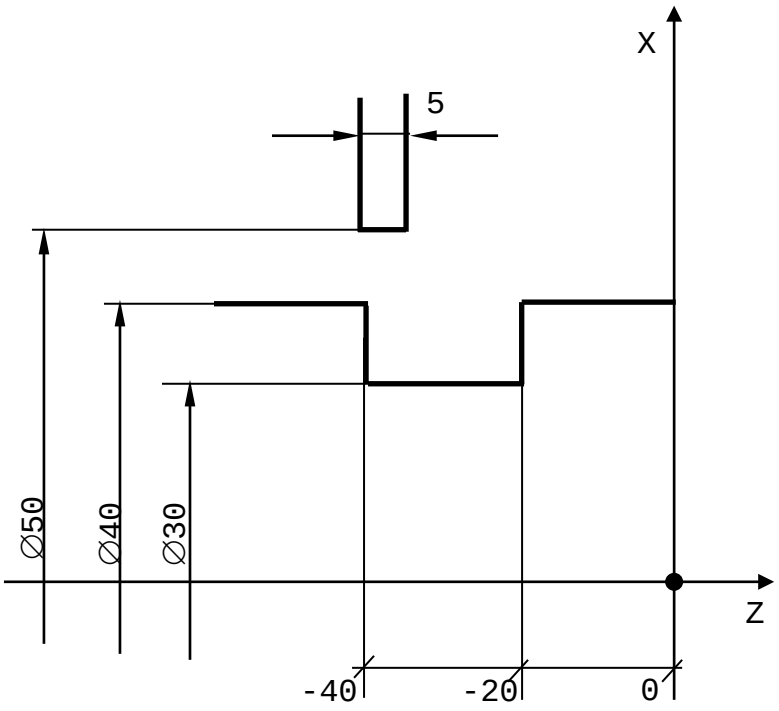
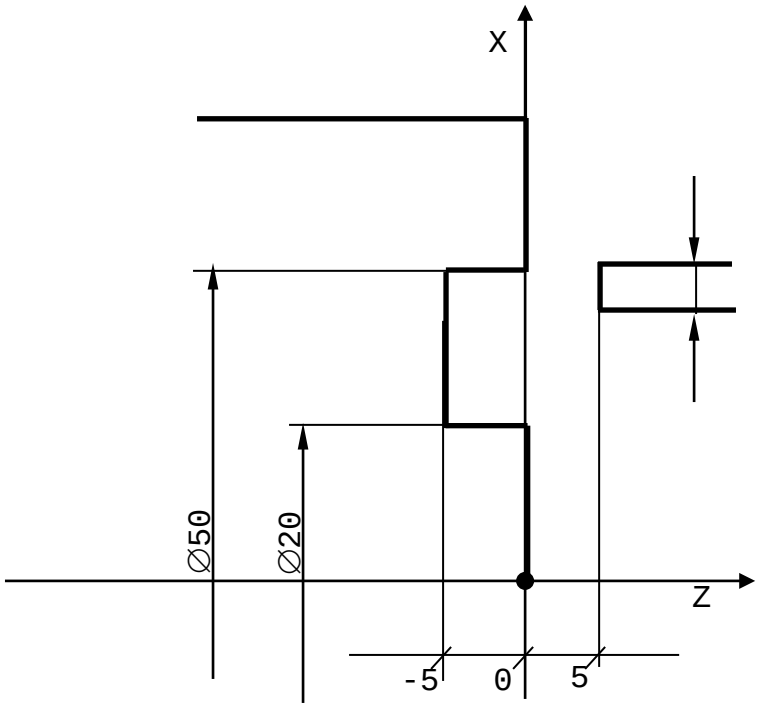


Рисунок А.112



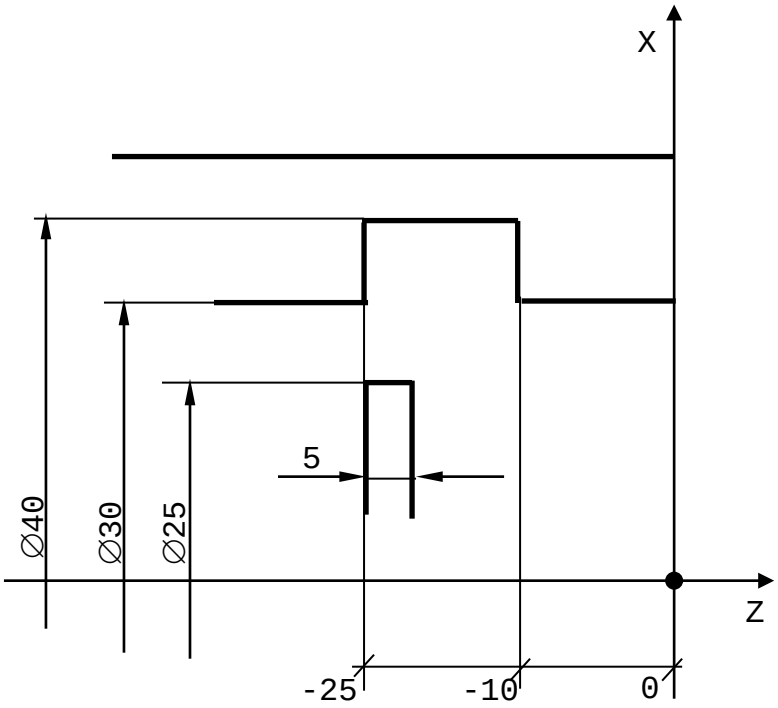

```
N1 T1.1 M6 S… F…
N2 G X30 Z-40
N3 TMR = 2
N4 (TGL,Z-20,X30,K5)
N5 G X… Z…
```

Рисунок А.113




```
N1 T1.1 M6
N2 G X20 Z5
N3 TMR = 2
N4          (TGL,Z-
5,X50,K5)
```

Рисунок А.114




```
N1 T1.1 M6 S… F…
N2 G X25
N3 Z-25
N4 TMR = 2
N5 (TGL,Z-10,X40,K5)
N6 G X… Z…
```

Рисунок А.115

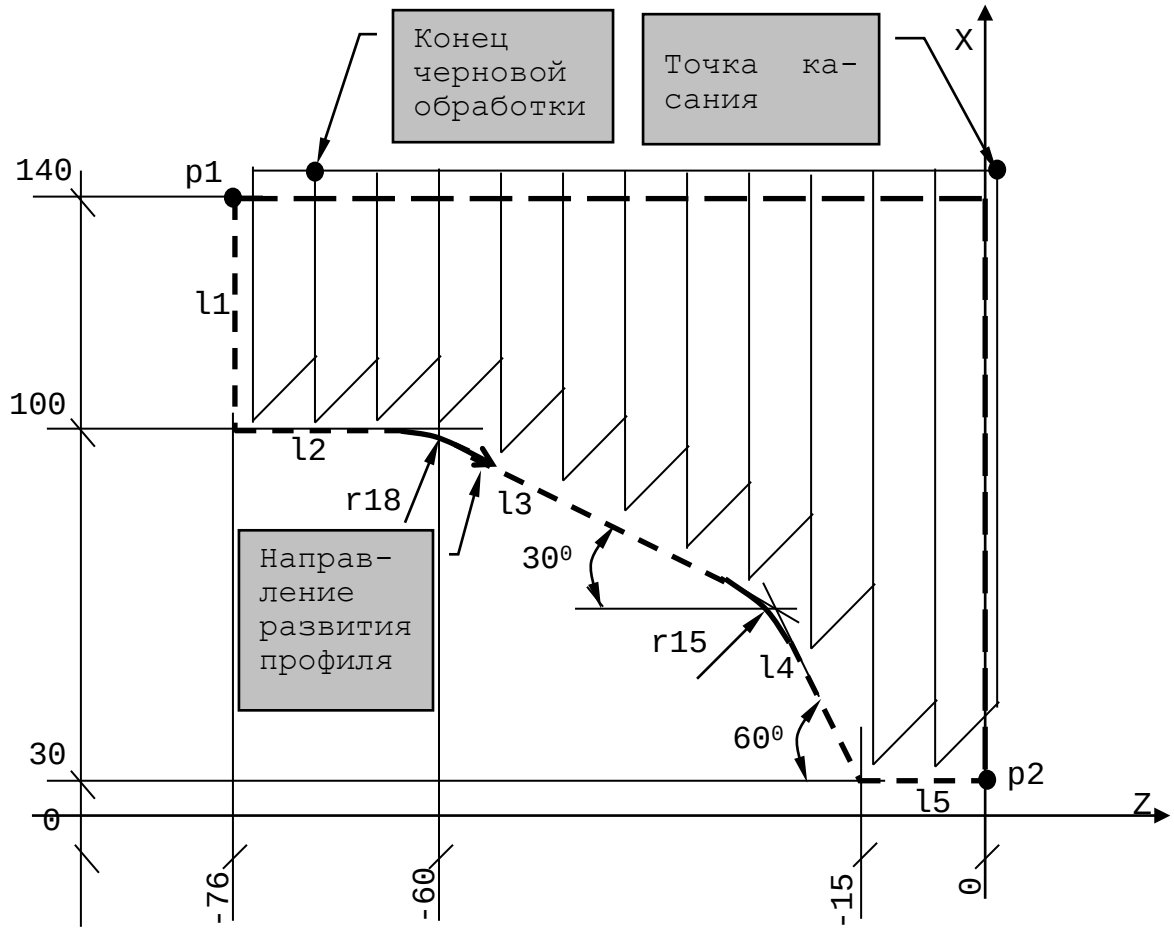


Рисунок А.116

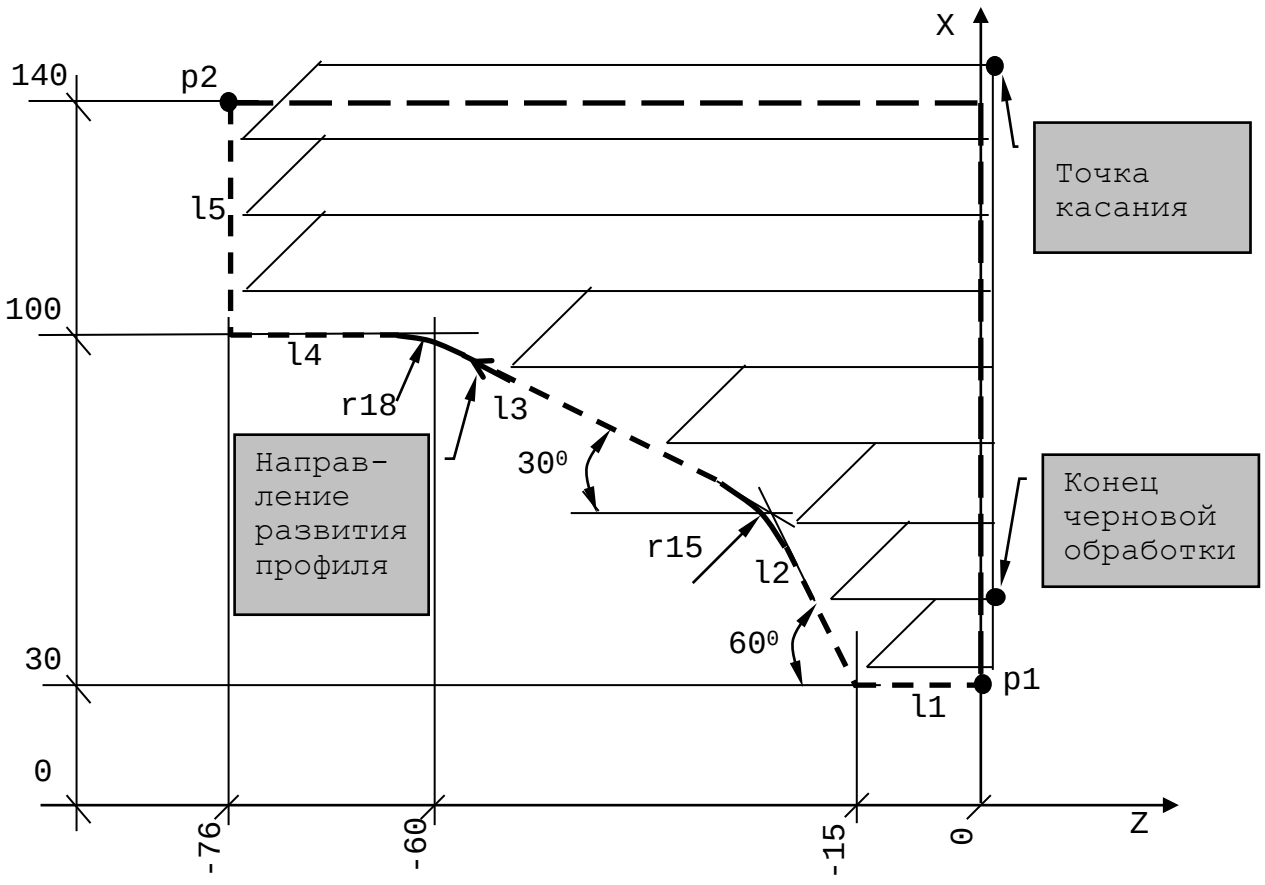


Рисунок А.117

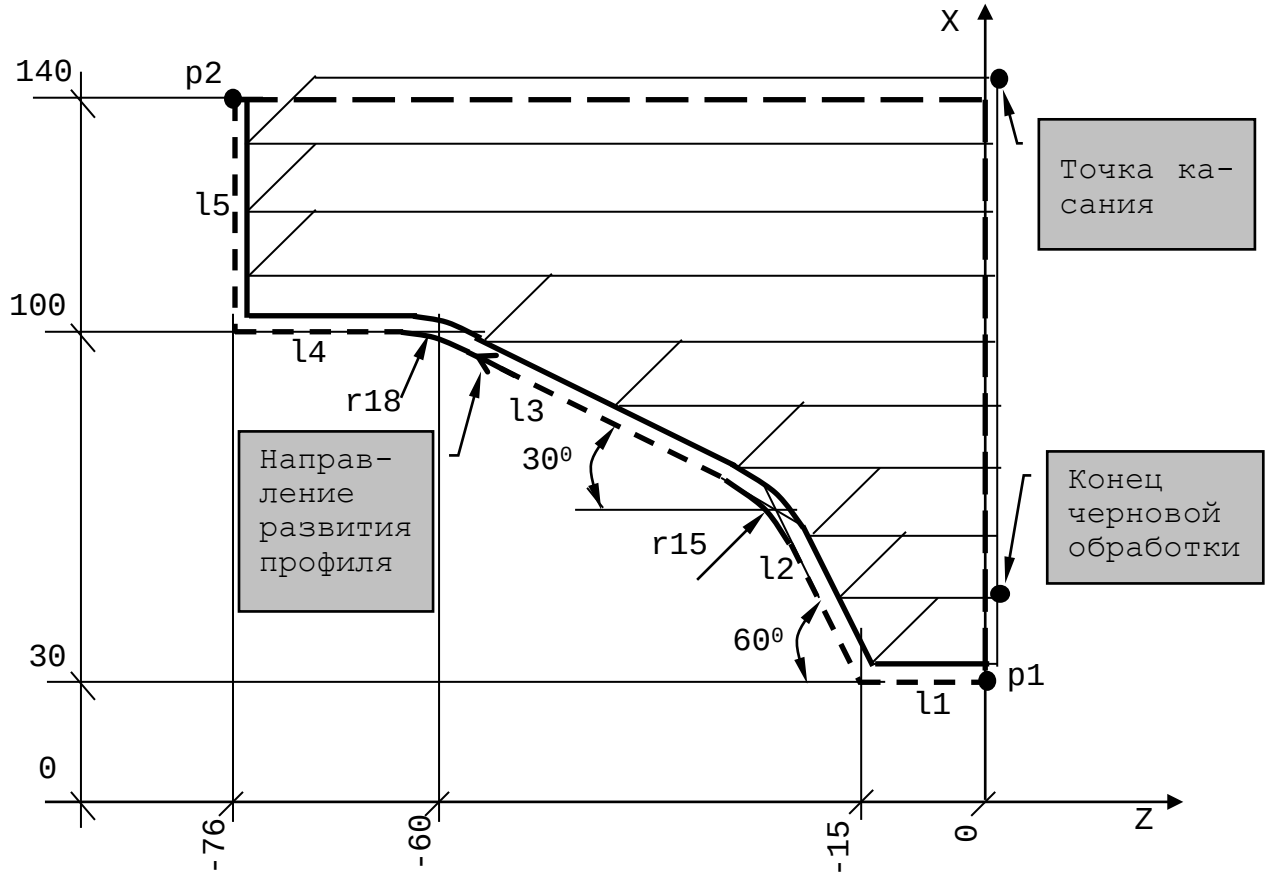


Рисунок А.118

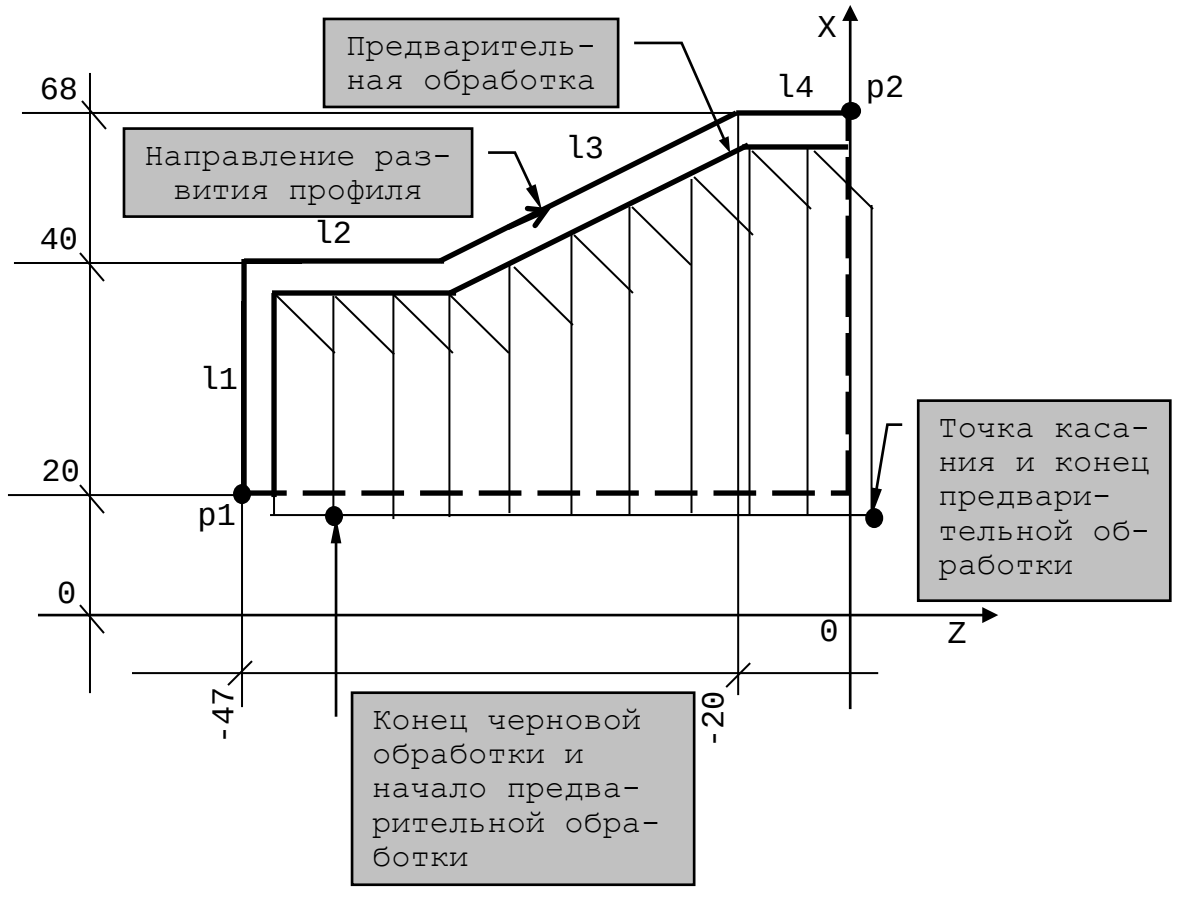


Рисунок А.119

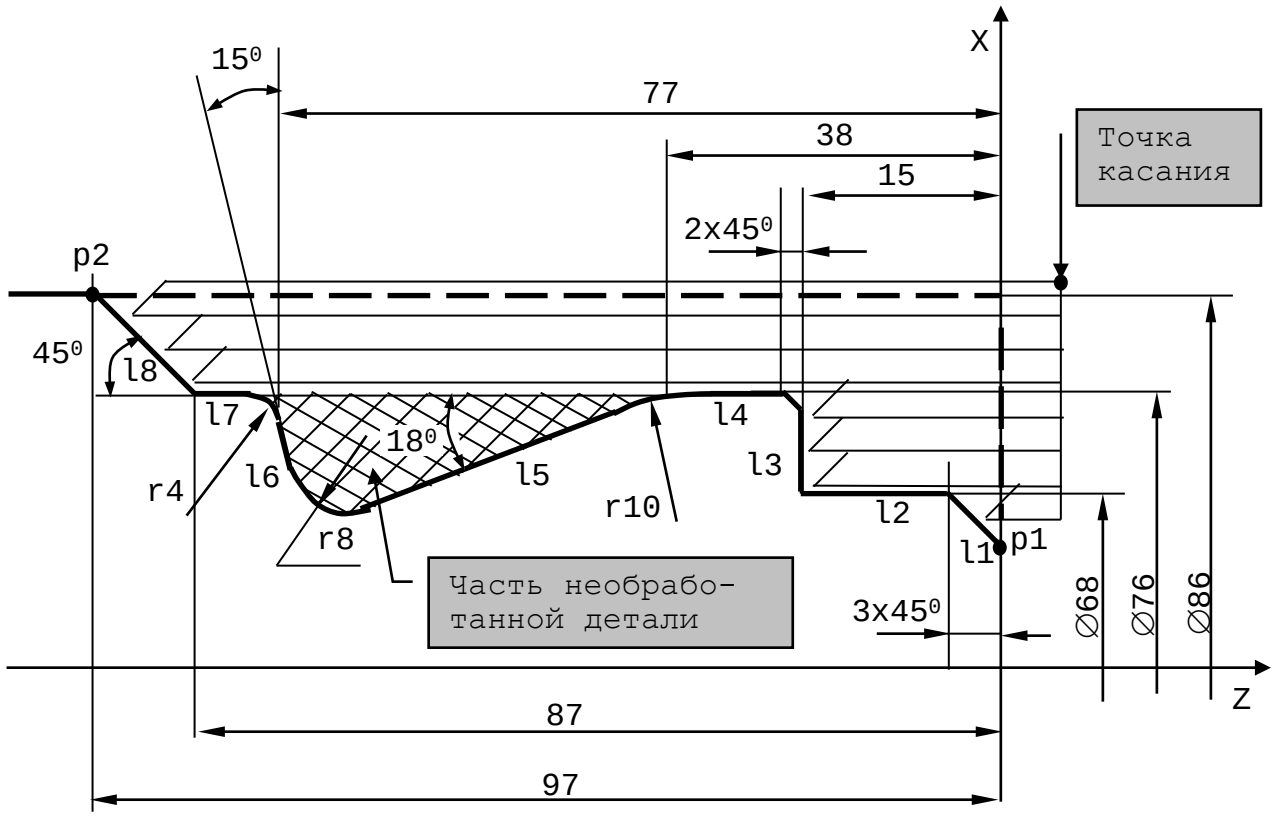


Рисунок А.120

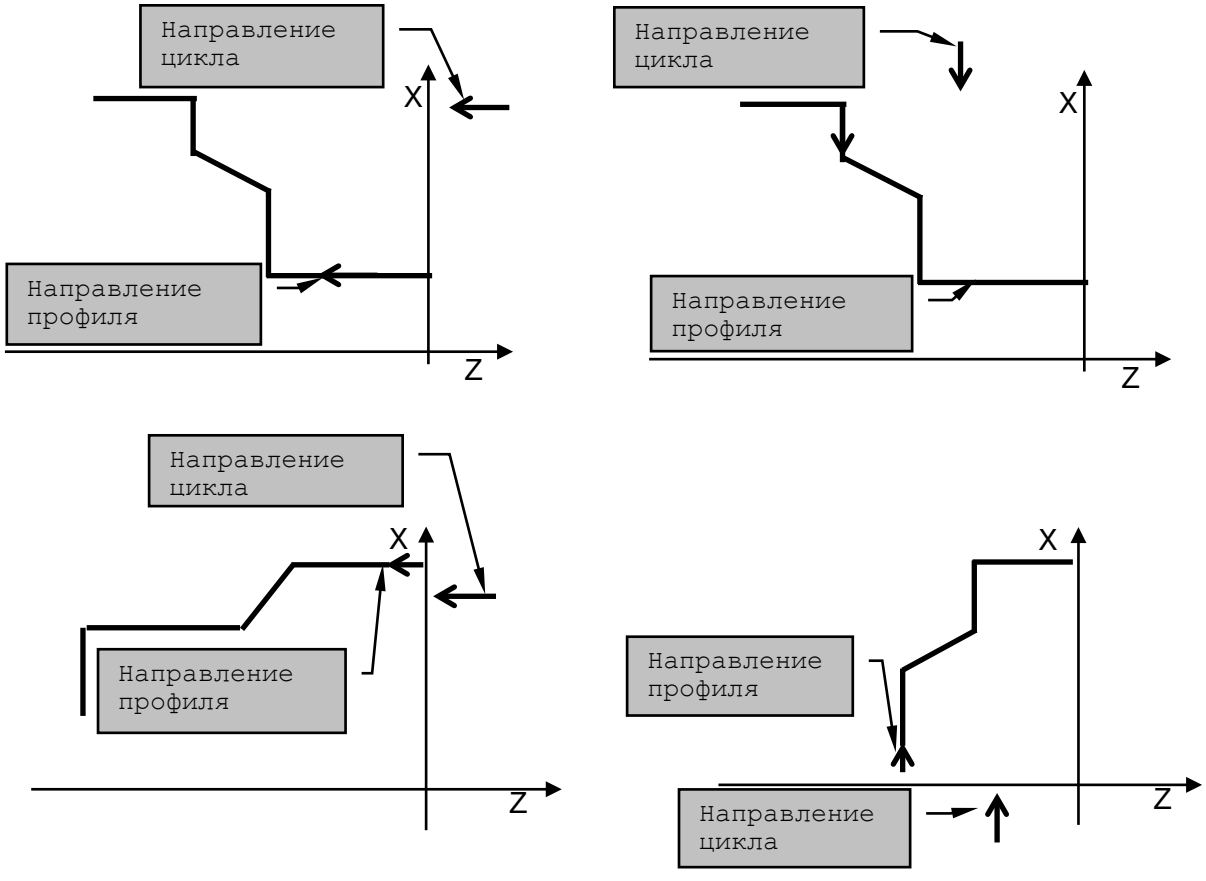


Рисунок А.121

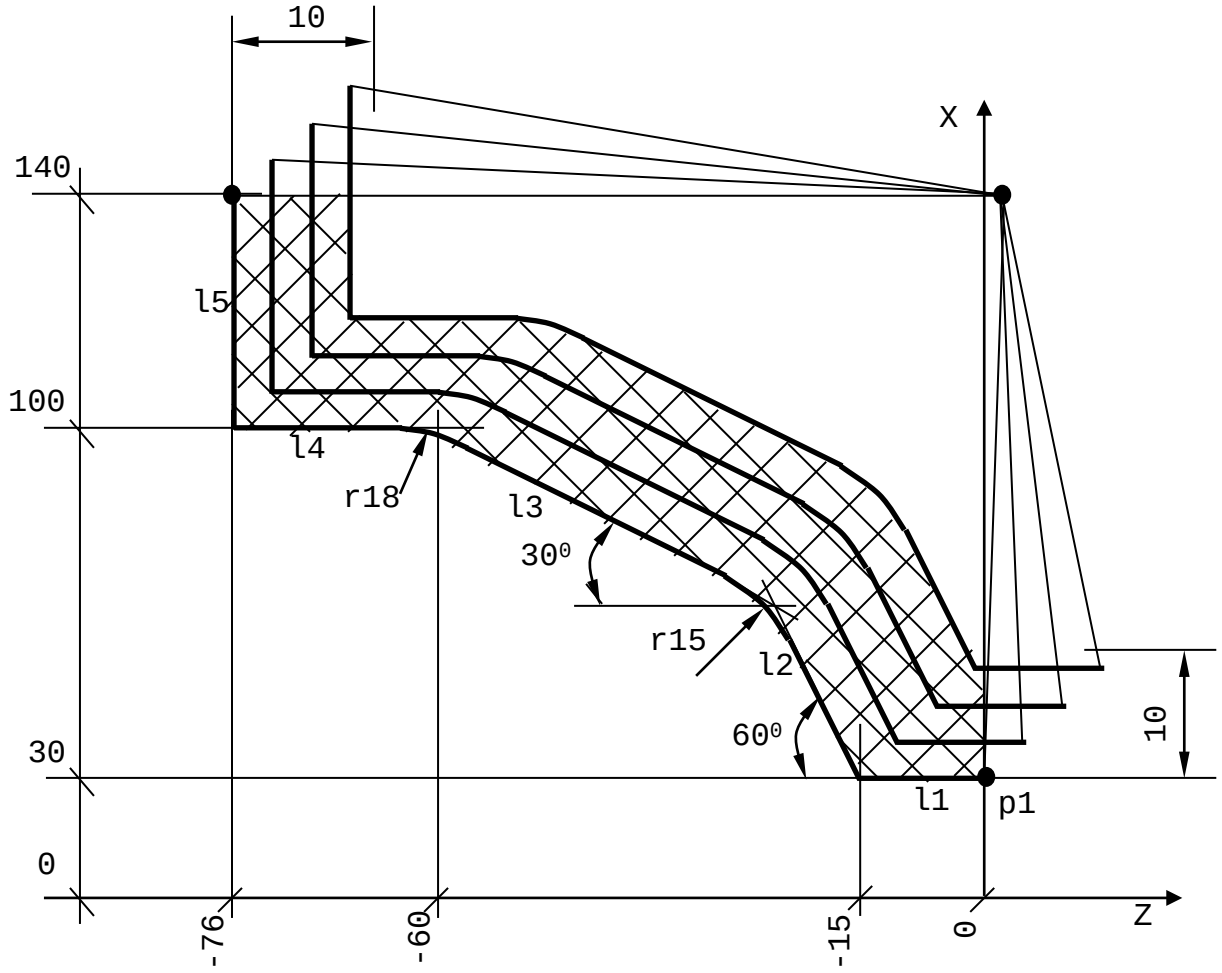


Рисунок А.122

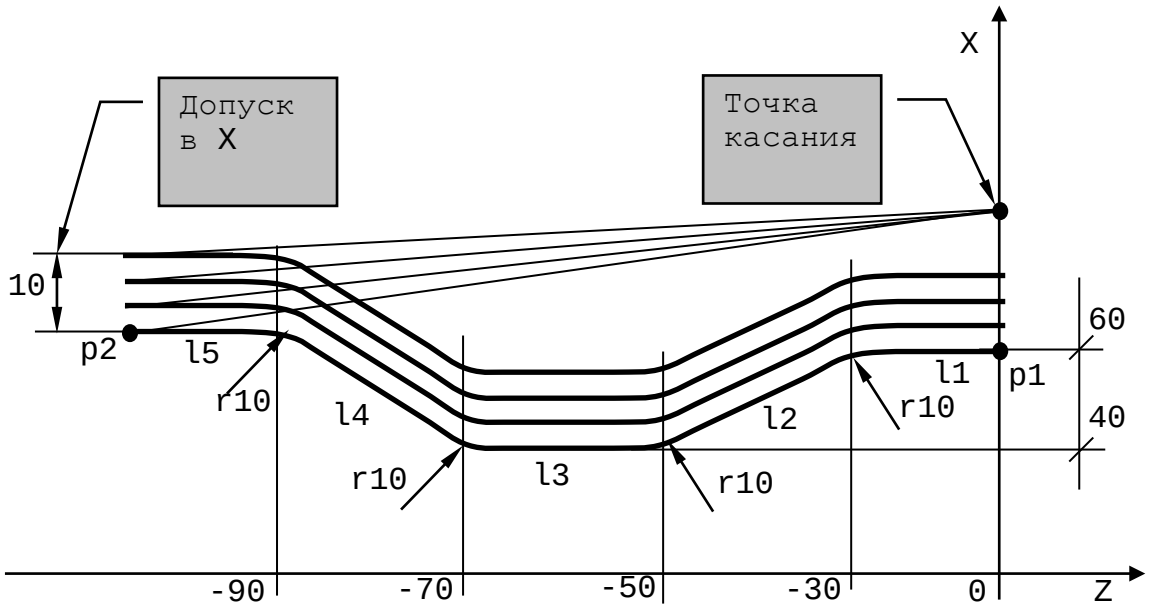


Рисунок А.123

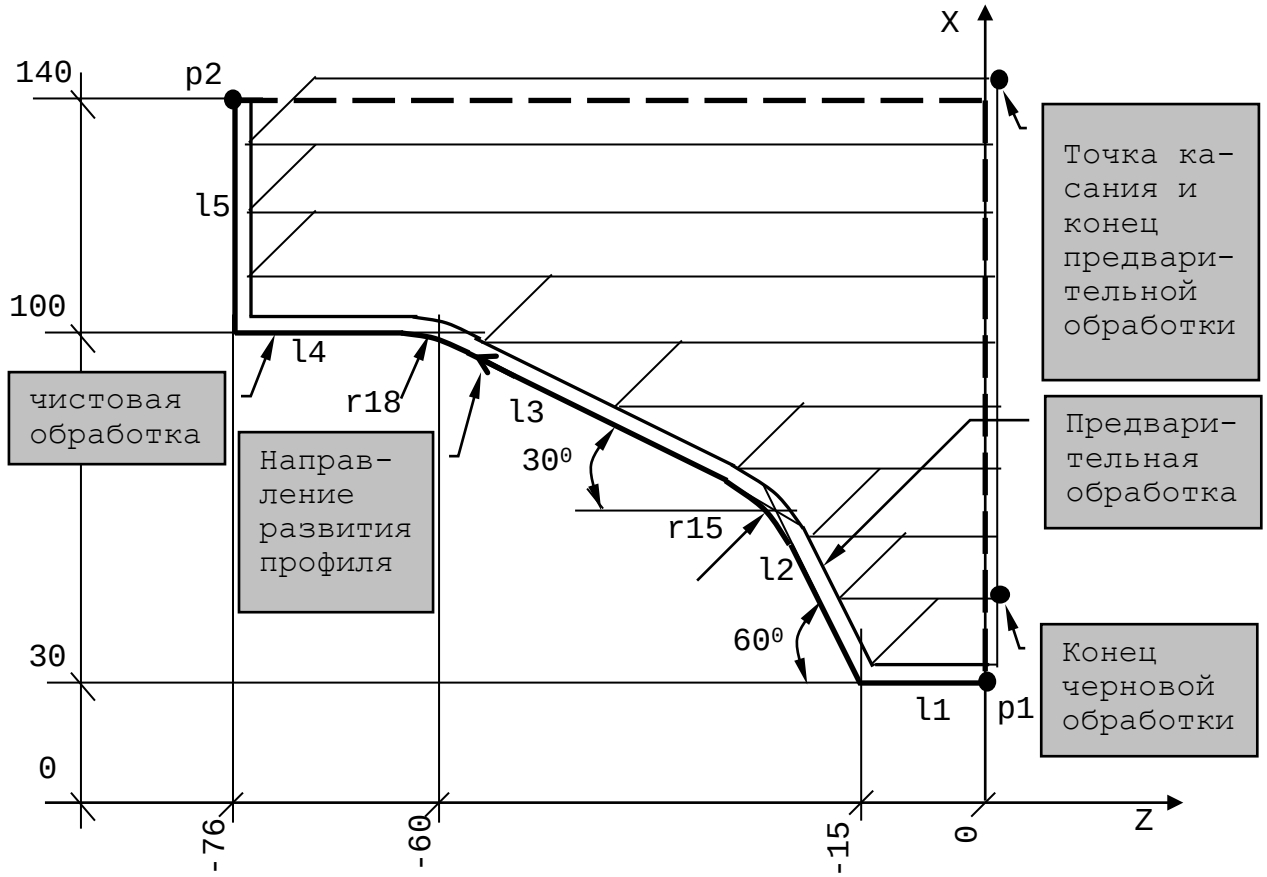


Рисунок А.124