

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ  
КУРГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра автоматизации производственных процессов

## **ПРОЕКТИРОВАНИЕ УПРАВЛЯЮЩИХ ПРОГРАММ ДЛЯ МНОГООПЕРАЦИОННЫХ СТАНКОВ С ЧПУ**

### **МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

для самостоятельной работы по дисциплинам  
«Программное управление технологическим оборудованием»,  
«Управление системами и процессами», «Управление  
станками и станочными комплексами»  
студентов специальностей

**220301 – «Автоматизация технологических  
процессов и производств (в машиностроении)»,**

**151001 – «Технология машиностроения»,**

**151002 – «Металлообрабатывающие станки  
и комплексы»**

Кафедра автоматизации производственных процессов

Дисциплины: «Программное управление технологическим оборудованием», «Управление системами и процессами», «Управление станками и станочными комплексами»

Составили: канд. техн. наук, доцент Н.Б.Сбродов,  
старший преподаватель А.В.Пережогин

Утверждены на заседании кафедры АПП « 06 » сентября 2007 г.

Рекомендованы методическим советом университета  
« 12 » октября 2007 г.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ	С. 4
1 ВХОДНОЙ ЯЗЫК УСТРОЙСТВА ЧПУ NC210	5
1.1 Информация управляющих программ	5
1.2 Системы координат	7
1.3 Программирование координатных перемещений	9
1.4 Примеры программирования перемещений	13
1.5 Формальные параметры	16
1.6 Коррекция положения и размеров инструмента	17
1.7 Подпрограммы, циклы, инструкции переходов	18
1.7.1 Инструкции переходов	18
1.7.2 Инструкции циклов	19
1.7.3 Подпрограммы	19
1.8 Программирование режимов резания и функций цикловой автоматики станка	19
2 ОСНОВЫ ПРОЕКТИРОВАНИЯ УПРАВЛЯЮЩИХ ПРОГРАММ	20
2.1 Функциональная структура УП	20
2.2 Основные требования к УП	21
2.3 Порядок разработки и документирования управляющей программы	25
2.3.1 Формирование траектории	25
2.3.2 Второй этап проектирования – разработка структуры УП и спецификаций входящих программ	28
2.3.3 Третий этап – кодирование УП	28
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	30

## ВВЕДЕНИЕ

Разработка управляющей программы (УП) для станка с ЧПУ - завершающий этап проектирования операционного технологического процесса. На этом этапе решаются следующие задачи:

- дальнейшая детализация технологической информации;
- ее формализованное представление;
- запись на входном языке устройства ЧПУ.

Первая из перечисленных задач сводится к выбору координатных систем и фиксации их в рабочем пространстве станка, формированию траекторий инструментов и расчету координат опорных точек, корректировке расчетных режимов резания в соответствии с ограничениями станочных приводов.

Вторая задача связана с распределением памяти устройства ЧПУ, отводимой для хранения значений переменных (формальных параметров), используемых в УП: адресов инструментов в станочном накопителе, параметров коррекции их положения и размеров, смещений рабочих координатных систем, погрешностей базовых элементов приспособлений, режимов резания и др.

Третья задача - это "кодирование" УП в соответствии с ограничениями входного языка устройства ЧПУ.

Последовательное решение всех трех задач определяет основное содержание этапа проектирования УП. Вслед за этим этапом следуют этапы отладки УП. При этом трудоемкость отладки УП и ее корректировки в процессе эксплуатации во многом зависит от технических решений, принимаемых разработчиком на всех этапах ее проектирования. Немаловажное значение имеет здесь и качество проектной и эксплуатационной документации, разрабатываемой в процессе проектирования УП.

Накопленный к настоящему времени опыт разработки относительно сложных УП для многооперационных станков с ЧПУ позволяет сформулировать основной принцип построения таких программ - это принцип единства функциональной структуры УП. Согласно этому принципу программа формируется, как типовая последовательность функциональных блоков, каждый из которых реализуется также типовыми (для конкретного станка и устройства ЧПУ) языковыми средствами. При этом адресация формальных параметров УП, определяющих положение, размеры и корректирующие смещения инструментов и рабочих координатных систем, также производится по единым типовым правилам.

Все это позволяет получить управляющую программу, которая легко читается, удобна для изучения, корректировки и размерной настройки в процессе эксплуатации.

Цель предлагаемых методических указаний - освоение методики

проектирования управляющих программ, изучение типовых приемов и получение практических навыков в программировании обработки деталей на станках с ЧПУ.

В качестве базового выбран многооперационный станок с ЧПУ модели VM501ПМФ4, оснащенный устройством ЧПУ модели NC210.

## **1 ВХОДНОЙ ЯЗЫК УСТРОЙСТВА ЧПУ NC210**

Входной язык устройства ЧПУ NC210 предоставляет пользователю широкие возможности для составления компактных управляющих программ. Среди таких возможностей следует отметить:

- практически неограниченное число рабочих координатных систем, выбираемых пользователем для удобства программирования;
- возможность построения циклических программ и программ с условными переходами;
- возможность использования подпрограмм ;
- возможность использования переменных (формальных параметров);
- программное присвоение значений переменным, наличие математических операций над ними.

Ниже приведено краткое описание входного языка.

### **1.1 Информация управляющих программ**

Программа состоит из последовательности кадров, которые позволяют описать цикл обработки. Каждый кадр - это последовательность информационных слов, определяющих операции, которые необходимо выполнить.

Слово состоит из адреса, за которым следует цифровое значение. Адрес представлен буквой, которая определяет тип инструкции (команды). Все цифровые значения, которые записаны за адресом, 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.

**Пример:** G1 X50.5 Y-3.15 F200 T1.1

Каждый кадр должен заканчиваться символом LF. Максимальная длина кадра - 128 символов.

В языке можно определить четыре типа кадров:

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

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

**Пример:** ; - ЭТО УПРАВЛЯЮЩАЯ ПРОГРАММА  
ОБРАБОТКИ ВАЛА СТУПЕНЧАТОГО 123-79

Кадры ISO – это кадры, слова которых предусмотрены стандартом ISO. Это основной вид кадров в программе, описывающий процесс обработки детали.

**Пример:** N23 G1 X500 Y20 F200

Кадры назначения непосредственно из программы пользователя позволяют определить величину нескольких глобальных параметров системы. Впоследствии эти параметры могут быть использованы в других кадрах того же или другого класса.

Кадры с трехбуквенными кодами - это кадры, в которых тип операции, выполнение которой предусмотрено, определен трехбуквенной командой (кодом).

**Пример:** N50 (URT,45).

Все кадры, кроме комментирующего, 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;
- 2) поле метки;
- 3) поле номера кадра.

Они могут присутствовать в кадре по одиночке или одновременно.

Поле подтверждения кадра и выведения его из рабочего состояния позволяет включить в программу кадры, выполнение которых зависит от параметра системы, названного USB. Если параметр является активным (равен «1»), кадр выполняется, в противном случае кадр рассматривается как комментирующий. Формат устанавливается знаком «/» в первой позиции кадра.

**Пример:** /N100G00X100

Поле метки позволяет дать символическое название кадру, которому оно принадлежит. Метка служит для возможности вызова кадра из

различных точек программы при помощи инструкций перехода. Метка - это алфавитно-цифровая последовательность символов, заключенная в знак «» (кавычки) , максимальная длина которой - шесть символов. Метка должна быть запрограммирована сразу же после поля «/», если оно присутствует.

**Пример:** "START"

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 (если они имеются).

**Пример:** N125

"START" N1

## 1.2 Системы координат

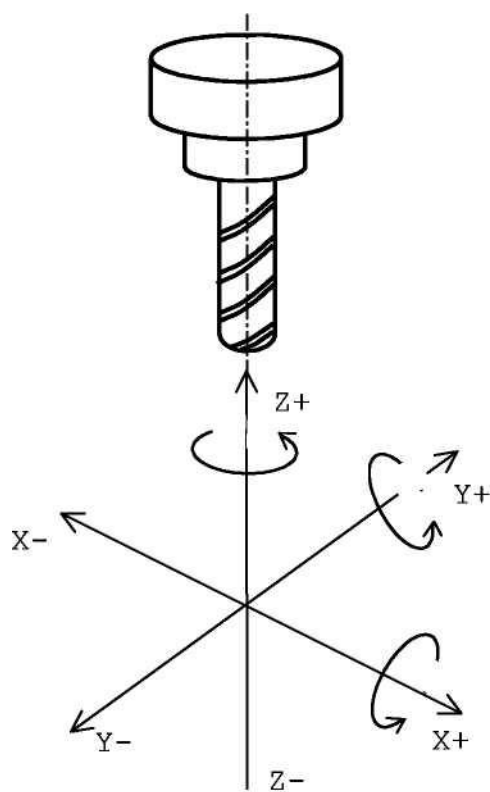
Расположение осей координат в станках с ЧПУ совпадает с направлением перемещений приводов подачи. Стандарт ISO определяет направление осей и их символику. За положительное направление оси считается то, при котором инструмент и деталь отходят друг от друга, независимо, что при этом будет двигаться ( рисунок 1).

Оси X, Y, Z называются первичными. Ось Z является осью рабочего шпинделя. Углы поворота вокруг первичных осей обозначаются A, B, C (соответственно для поворота вокруг осей X, Y, Z).

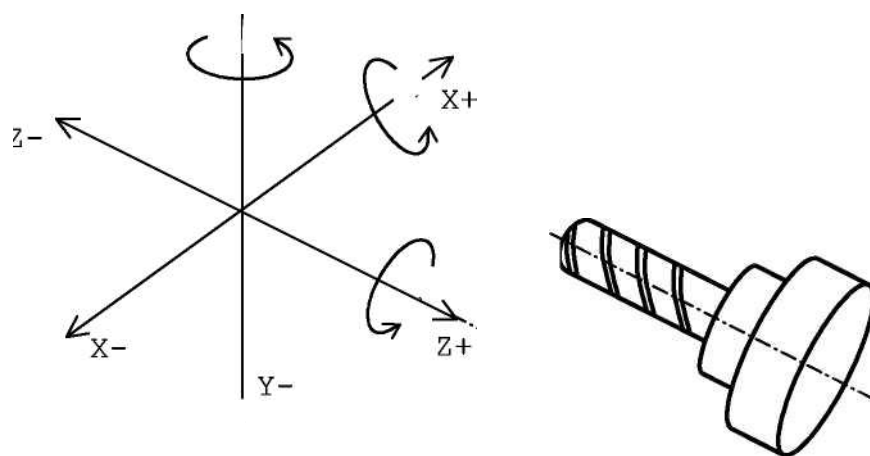
Начало станочной координатной системы соответствует "нулевым" показаниям датчиков положения рабочих органов станка. Как правило, рабочие органы станка находятся при этом в своих "крайних" положениях.

Современные УЧПУ, в т.ч. NC210, позволяют пользователю вводить большое количество дополнительных рабочих координатных систем, привязывая их начало (исходные точки) к базовым элементам зажимных приспособлений, к элементам обрабатываемой детали, относительно которых проставлены размеры на чертеже и т.п. Все это необходимо для облегчения расчета размерных перемещений и удобства программирования.

Вводя с клавиатуры УЧПУ NC210 трехбуквенный код ORA, относящийся к сконфигурированным осям станка, например: ORA, n, Z.., X..; можно определить максимально 100 абсолютных исходных точек (от 0 до 99). Исходные точки можно активизировать из программы трехбуквенным кодом UAO.



а) вертикальный шпиндель



б) горизонтальный шпиндель

Рисунок 1 – Расположение координатных осей

Оператор UAO выбирает одну из абсолютных начальных точек, ранее определенных командой ORA. Формат:

(UAO, n) ,

где n - определяет номер начальной точки, которую надо выбрать.

**Пример:** (UAO,1) - активизируется абсолютная начальная точка 1.



При включении питания или после команды «Сброс» автоматически активизируется нулевая начальная точка.

Кроме абсолютных исходных точек, используя трехбуквенный код UOT, в программе можно определить бесконечное количество временных исходных точек, привязанных к любой из абсолютных. Оператор UOT выбирает абсолютную начальную точку, объявленную в кадре, изменяя ее временно на величину, равную запрограммированной. Формат:

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

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

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

**Пример (рисунок 2):**

(UAO,0) – активизируется абсолютная начальная точка 0.

..... – часть программы, отнесенная к абсолютной начальной точке 0 для всех осей.

(UOT,0, X100, Y100) – вводится временная начальная точка 1, смещенная на 100 мм по оси X и на 100 мм по оси Y относительно абсолютной начальной точки 0.

.....

.....

(UAO,1) – активизируется абсолютная начальная точка 1.

(UOT,1, X-250, Y50) – вводится временная начальная точка 3 с корректировками X-250 и Y50 по отношению к абсолютной начальной точке 1.

.....

.....

(UAO,0) – активизируется абсолютная начальная точка 0 для всех осей.

### 1.3 Программирование координатных перемещений

Координатные перемещения программируются словами X, Y, Z, задающими движение подач станка в направлениях, соответствующих координатных осей. Цифровая часть слова определяет координату конечной точки программируемого участка траектории в установленной координатной системе (если задана подготовительная функция G90) или величину перемещения (со знаком) по соответствующей оси (если задана подготовительная функция G91, определяющая режим программирования «в приращениях»).

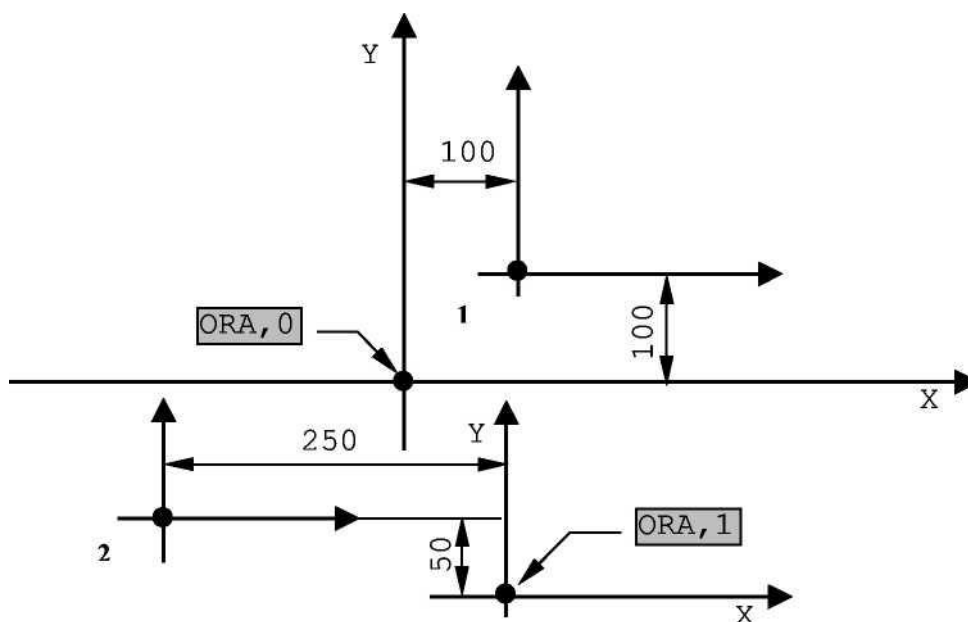


Рисунок 2 – Изменение системы отсчета координатных осей

Координаты и перемещения задаются в миллиметрах от + (-) 0.0001 до + (-) 99999.9999. Вращательные движения программируются от + (-) 0.0001 до + (-) 99999.9999 градусов.

Кроме указания координатных осей при программировании используются подготовительные G-функции (таблица 1)

Таблица 1 – Основные подготовительные функции, используемые при программировании координатных перемещений

Группа функции	G-функция	Назначение
1	G0	Движение в режиме позиционирования (на скорости быстрых ходов)
	G1	Движение по прямой на рабочей подаче (линейная интерполяция)
	G2	Движение по окружности по часовой стрелке (круговая интерполяция)
	G3	То же против часовой стрелки
2	G4	Выдержка времени в конце кадра

1	G33	Нарезание резьбы с постоянным или переменным шагом
3	G40	Отмена компенсации (коррекции) радиуса инструмента
	G41	Компенсация (коррекция) радиуса инструмента (инструмент слева)
	G42	Компенсация (коррекция) радиуса инструмента (инструмент справа)
4	G80	Отмена постоянных циклов
	G81	Постоянный цикл сверления
	G82	Постоянный цикл растачивания
	G83	Цикл глубокого сверления (с разгрузкой стружки)
	G84	Постоянный цикл нарезания резьбы метчиком
	G85	Постоянный цикл рассверливания
	G86	Постоянный цикл развертывания
	G89	Постоянный цикл развертывания с остановкой
5	G90	Задание перемещений в абсолютных координатах
	G91	Задание перемещений в относительных координатах (“приращениях”)

В одном кадре УП могут быть указаны G-функции, относящиеся к разным группам. Каждая подготовительная функция, однажды указанная в УП, действует до тех пор пока не будет отменена G-функцией этой же группы.

Тип движения определяется подготовительными функциями G0, G1, G2, G3, G33.

Быстрое позиционирование осей (G0) определяет линейный тип движения, скоординированный по всем осям, запрограммированным в кадре с быстрым ходом.

Линейная интерполяция (G1) определяет линейное одновременное движение, скоординированное по всем осям, которые запрограммированы в кадре, с заданной скоростью обработки.

Формат:

**G1 [ДРУГИЕ G] [ОСИ] [ОПЕРАНД КОРРЕКТИРОВКИ]  
[СКОРОСТЬ ПОДАЧИ],**

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

**[СКОРОСТЬ ПОДАЧИ]** - выражает рабочую скорость (слово F), с которой выполняется движение. В случае отсутствия используется ранее запрограммированная скорость. Слова, заключенные в квадратные скобки, могут отсутствовать в кадре.

Круговая интерполяция (G2 - G3) определяет круговое движение по часовой стрелке (G2) или против часовой стрелки (G3). Это движение является скоординированным и одновременным по всем осям, запрограммированным в кадре с заданной скоростью обработки.

Формат:

**{G2 или G3} [ДРУГИЕ G] [ОСИ] I J [СКОРОСТЬ ПОДАЧИ]  
[ОПЕРАНДЫ КОРРЕКТИРОВКИ],**

где **[СКОРОСТЬ ПОДАЧИ]** - скорость подачи;

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

**I** и **J** - являются адресными словами, выражающими координаты центра окружности, цифровая часть которых может быть выражена в явной или неявной форме (параметр E) .

Максимальная программируемая дуга - 360 градусов

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

Формат:

**{G2 или G3} [ДРУГИЕ G] [ОСИ] R+ [СКОРОСТЬ ПОДАЧИ]  
[ОПЕРАНДЫ КОРРЕКТИРОВКИ],**

где **[СКОРОСТЬ ПОДАЧИ]** - скорость подачи;

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

**R** - адресное слово, выражающее радиус дуги окружности, цифровая часть которой может быть выражена в явной или неявной форме (параметр E) ; знак «+» или «-» перед адресным словом R выбирает одно из двух возможных решений: «+» - для дуги до 179.999°; «-» - для дуги от 180° до 359.999°.

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.

Формат кадра постоянного цикла следующий:

**G8X [ДРУГИЕ G] [R1[R2]] КООРДИНАТА ЦИКЛА  
[ДОПОЛНИТЕЛЬНЫЕ ОПЕРАНДЫ] [СКОРОСТЬ ПОДАЧИ]**

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).

Последовательность движений при постоянных циклах можно представить следующим образом:

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

#### 1.4 Примеры программирования перемещений

Пример линейной интерполяции и быстрого позиционирования из точки 1 в точку 2 приведён на рисунке 3.

**Пример линейной интерполяции (в абсолютных координатах G90):**  
N15 G1 G90 X100 Y50 F180.

**Пример линейной интерполяции (в относительных координатах G91):**

N15 G1 G91 X90 Y40 F180.

**Пример быстрого позиционирования:** N20 G0 G90 X100 Y50.

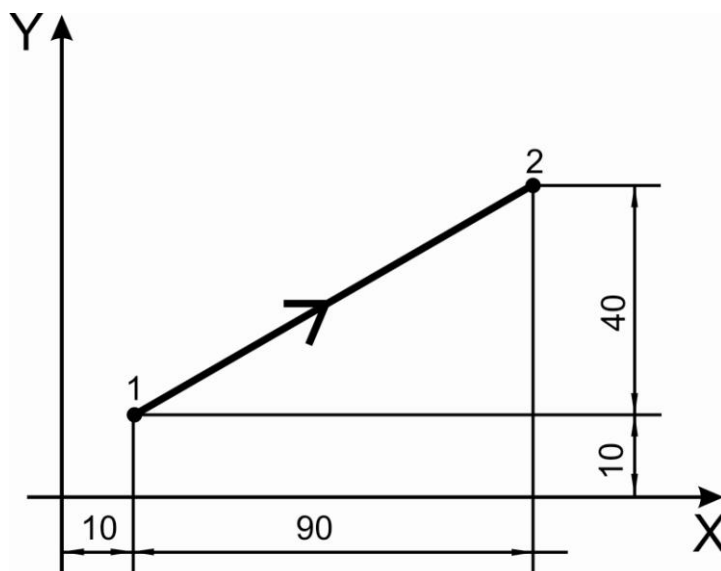


Рисунок 3 – Пример линейной интерполяции

Пример круговой интерполяции приведён на рисунке 4.

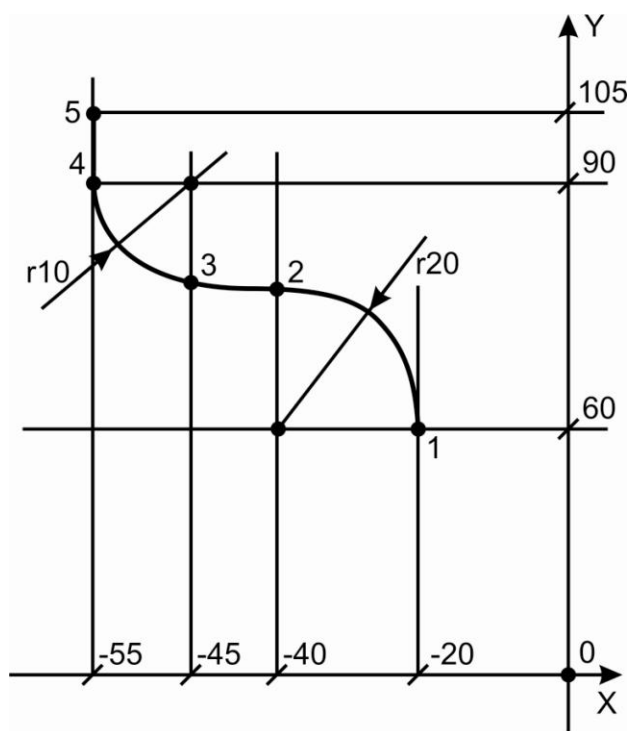


Рисунок 4 – Пример круговой интерполяции

**Пример круговой интерполяции (в абсолютных координатах G90):**  
 N10 G1 G90 X-20 Y60 F200 – линейное перемещение в точку 1;

N20 G3 X-40 Y80 I-40 J60 – перемещение по дуге 1-2;  
 N30 G1 X-45 – линейное перемещение в точку 3;  
 N40 G2 X-55 Y90 I-45 J90 – круговая интерполяция по дуге 3-4;  
 N50 G1 Y105 – линейное перемещение в точку 5.

**Пример круговой интерполяции (в относительных координатах G91, в «приращениях»):**

N10 G1 G91 Y60 F200 – линейное перемещение в точку 1;  
 N20 G3 X-20 Y20 I-40 J60 – перемещение по дуге 1-2;  
 N30 G1 X-5 – линейное перемещение в точку 3;  
 N40 G2 X-10 Y10 I-45 J90 – круговая интерполяция по дуге 3-4;  
 N50 G1 Y15 – линейное перемещение в точку 5.

Пример программирования дуги менее 360 градусов через задание координат конечной точки и радиуса приведён на рисунке 5.

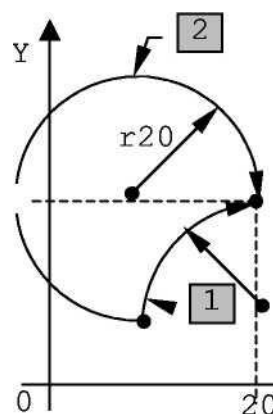


Рисунок 5 – Пример круговой интерполяции

G02 G90 X20 Y20 R+20 F100 – перемещение по дуге 1.  
 G02 G90 X20 Y20 R-20 F100 – перемещение по дуге 2.

Пример применения постоянного цикла сверления G81 показан на рисунке 6.

**Пример:**

N33 G81 R3 Z-15 M3 - задание параметров постоянного цикла сверления G81;  
 N34 X15 Y15 - движение к точке 1 и выполнение цикла;  
 N35 Y60 - движение к точке 2 и выполнение цикла;  
 N36 X80 - движение к точке 3 и выполнение цикла;  
 N37 Y15 - движение к точке 4 и выполнение цикла;  
 N38 G80 Z50 M5- отмена действия цикла.

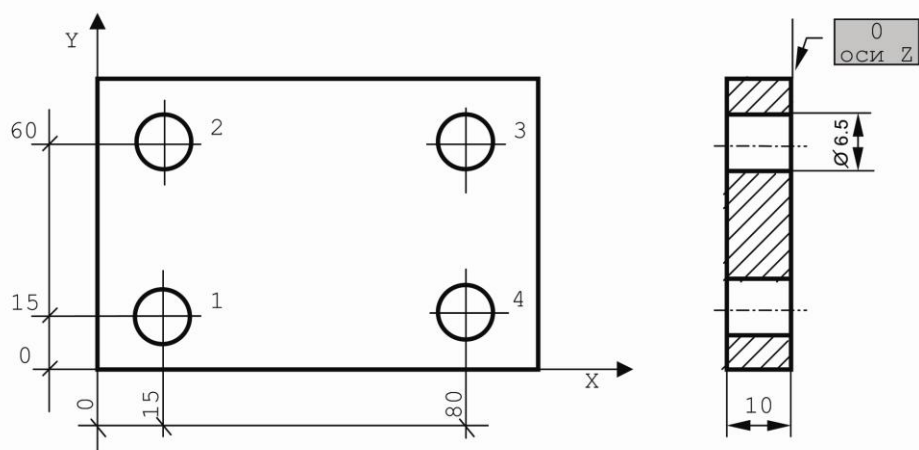


Рисунок 6 - Пример применения постоянного цикла сверления G81

### 1.5 Формальные параметры

Вместо цифровой части слов УП могут использоваться переменные (формальные параметры), значения которых могут быть занесены в память УЧПУ с пульта оператора или определены непосредственно в программе путем присвоения им численного значения. При помощи параметров можно осуществлять арифметические и тригонометрические операции.

Формальный параметр обозначается в программе символом E, после которого записывается номер. Максимальное число параметров E не ограничено.

**Примеры** использования параметров в словах УП:  
 X + E15; Y - E20; T E55; S E12.

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 и др.

Индекс A может быть параметром E или цифровым значением.

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



Можно выполнять выражения, содержащие постоянные, параметры, функции.

**Пример:**  $N1\ E37=E31*\text{SIN}(E30)+123.4567/\text{SQR}(16).$

**Пример** кадров назначения для вычисления переменных:

"LAB 1"  $E51 = -0.00000124 + 5/E35;$

$E7 = 81;$

$E10 = 1;$

$E25 = E25 + 30.$

Использование формальных параметров позволяет широко использовать аппарат подпрограмм, повышает "гибкость" программы и создает возможность ее настройки на конкретные условия эксплуатации. При этом следует помнить, что все формальные параметры имеют глобальный характер, т.е. могут использоваться или модифицироваться как в головной программе, так и в любой из подпрограмм.

## 1.6 Коррекция положения и размеров инструмента

Другим средством повышения гибкости и универсальности разрабатываемых УП является аппарат коррекций положения и размеров инструмента. Компенсация радиуса инструмента позволяет осуществить программирование контуров профиля без учета радиуса инструмента. Используя коррекцию, можно проектировать программы, практически не зная точных размеров инструментов и расположения базовых элементов приспособления, а затем настраивать программу непосредственно на рабочем месте в соответствии с конкретными параметрами инструментальной наладки.

При программировании коррекций используются подготовительные функции G41 – коррекция радиуса инструмента (инструмент слева), G42 – коррекция радиуса инструмента (инструмент справа). Коррекция отменяется функцией G40. Корректировка радиуса действует в перпендикулярном направлении к запрограммированному профилю. Величина коррекции заносится в ячейку памяти (корректор) УЧПУ. Номер корректора задается в слове T (смена инструмента).

Этот вид коррекции позволяет программировать движение центра фрезы непосредственно по обрабатываемому контуру. При этом фактическая траектория фрезы будет формироваться автоматически путем эквидистантного смещения заданной в УП линии на величину, записанную в корректоре, в направлении "от контура к инструменту".

Функция G41 используется в тех случаях, когда инструмент располагается слева от контура детали (если смотреть по ходу его движения), G42 - когда инструмент расположен справа.

**Пример программирования коррекции (рисунок 7):**

1. Инструмент справа от детали:

N10 G1 G42 X-50 Y15 F200 - выход инструмента на скорректированную траекторию осуществляется отдельным кадром УП (кадр "ввода коррекции"); обрабатываемый контур обозначен сплошной толстой линией.

N20 X-20 Y45 – обработка участка траектории с учетом корректирующего смещения, занесенного в соответствующий корректор для данного инструмента; траектория движения инструмента показана пунктирной линией справа от обрабатываемого контура.

2. Инструмент слева от детали:

N10 G1 G41 X-50 Y15 F200

N20 X-20 Y45

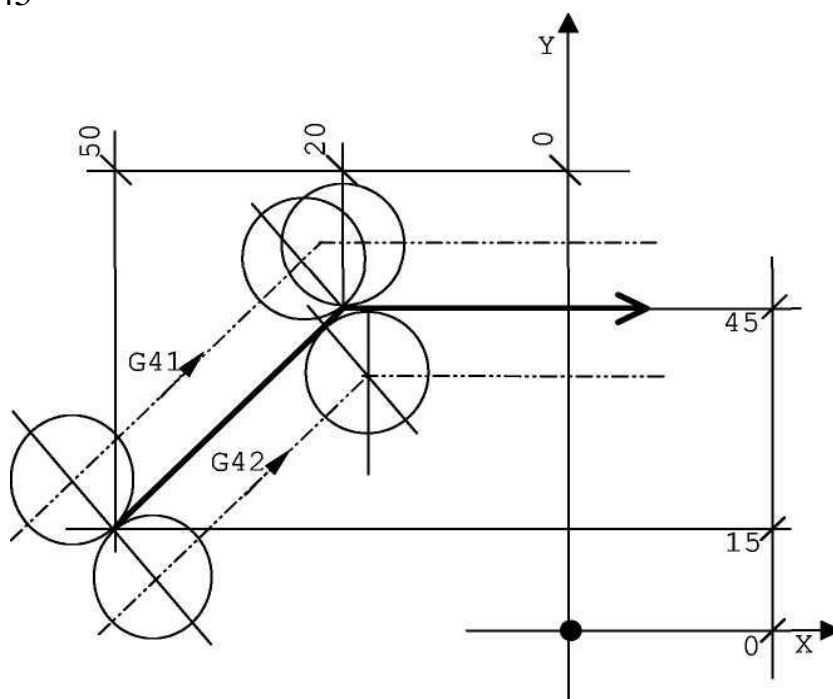


Рисунок 7 – Коррекция радиуса инструмента

## 1.7 Подпрограммы, циклы, инструкции переходов

При отсутствии в УП специальных "указаний" кадры обрабатываются в порядке их записи в тексте программы. Для изменения естественной последовательности обработки кадром можно использовать:

- безусловные и условные переходы;
- инструкции цикла;
- обращение к подпрограммам.

### 1.7.1 Инструкции переходов

Внутри программы можно программировать переходы посредством программирования инструкций, содержащих метку для передачи управ-

ления. Метка - это алфавитно-цифровая последовательность, состоящая из 6 символов, заключенных в знак « » (кавычки), которая должна быть запрограммирована перед номером кадра и после знака «/» в случае, если кадр разделен дробной чертой.

**Пример:** /"НАЧАЛО"N125

Переходы могут быть условными и безусловными.

**Пример:**

N5 (BNC, МЕТКА1) – переходит безусловно к метке МЕТКА1;

N10 (BGT,E1,123, END) - переходит к метке END, если значение переменной E1 больше 123;

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

### 1.7.2 Инструкции циклов

Используя трехбуквенный 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 трехбуквенным кодом ERP.

**Пример:** (RPT, 10)

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

### 1.7.3 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 объема используемой памяти. Подпрограмма вызывается трехбуквенным кодом CLS.

**Пример:** N35 (CLS,PROG1).

## 1.8 Программирование режимов резания и функций цикловой автоматики станка

К таким функциям относятся: назначение скорости вращения шпинделя и подачи, поиск и смена инструмента, включение охлаждения, зажим-разжим направляющих станка и др.

Скорость шпинделя станка задается словом S (от англ. SPINDL), в

котором указывается величина скорости привода главного движения.

Программируется от 0.01 до 99999.99. Может выражать:

- число оборотов/мин шпинделя (если в программе задана подготовительная функция G97);
- скорость резания в м/мин (для G96).

Скорость подачи задается словом F (от англ. FEED), в котором указывается значение подачи. Программируется от 0.01 до 99999.99. Чаще всего задается в мм/мин (должна быть задана подготовительная функция G94).

Номер устанавливаемого инструмента задается словом T (от англ. TOOL), в котором задается № требуемого для обработки инструмента и номер коррекции для данного инструмента. Программируемая величина: от 1.0 до 9999.9999. Цифры перед десятичной точкой определяют инструмент, после – номер корректора. Число коррекций определяется в фазе установки. Коррекция приводится в действие при помощи функции M06 (смена инструмента). Величины коррекции относятся к длине и диаметру инструмента. Корректировка длины инструмента может быть применена к любой оси станка. Выбор зависит от названия оси, к которой присоединена корректировка длины. Корректировка длины приводится в действие без использования других подготовительных функций. Корректировка диаметра инструмента, вызванная одновременно с корректировкой длины, приводится в действие при помощи функций компенсации радиуса инструмента G41/G42

Остальные функции программируются "вспомогательными" командами M. Состав и значения функций M определяются особенностями конкретного станка. Основные из них:

- 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;
- M30 - конец программы и возврат к первому кадру;

## **2 ОСНОВЫ ПРОЕКТИРОВАНИЯ УПРАВЛЯЮЩИХ ПРОГРАММ**

### **2.1 Функциональная структура управляющей программы**

Управляющая программа – это модель технологической операции,

то есть специфическая форма её описания на входном языке устройства ЧПУ. Отличительной особенностью такого описания является высокая степень детализации технологической информации, достаточной для однозначного восприятия и исполнения всех содержащихся в УП команд.

Как и всякая модель, УП отражает структуру реального объекта – технологической операции и может быть представлена в виде последовательности функциональных блоков, каждый из которых описывает определённый технологический переход (или часть перехода).

Например, в состав УП может входить блок установки заготовки в рабочую позицию, блок смены инструмента, блок подвода инструмента в рабочую зону, блок обработки заготовки и др. (Разумеется, каждый такой блок должен быть оптимально запрограммирован с учётом конструктивных особенностей станка и возможностей входного языка устройства ЧПУ).

Как уже отмечалось, один из главных принципов построения относительно сложных УП является принцип единства их функциональной структуры. Согласно этому принципу любая УП представляется стандартной последовательностью функциональных блоков.

Структура УП, иллюстрируемая блок-схемой на рисунке 8, может быть рекомендована для станков, в которых цикл смены инструмента более длителен, чем цикл переориентации заготовки на поворотном столе.

Содержание функциональных блоков приведено в таблице 2.

Функциональный блок Б1 является обязательным лишь для станков, оснащенных средствами автоматической загрузки/выгрузки палет с заготовками в рабочую зону.

## **2.2 Основные требования к управляющей программе**

Управляющая программа должна обеспечивать изготовление на станке годных деталей с требуемой производительностью, т.е. должна быть работоспособной. Требования работоспособности УП реализуются как на стадии технологического проектирования (рациональный выбор состава и последовательности переходов, комплекта режущих инструментов и траекторий, их движения, режимов резания и т.д.), так и в процессе разработки и отладки программы.

При всей важности и безусловности выполнения требования работоспособности УП, при ее создании приходится учитывать также и множество других, часто противоречивых требований, реализация которых обеспечивает "оптимальность" программы на всех трех стадиях ее жизненного цикла: при разработке, отладке и в процессе эксплуатации.

Трудоемкость разработки, отладки и корректировки УП можно существенно снизить, если следовать определенным правилам ее построения. Многие из этих правил непосредственно вытекают из

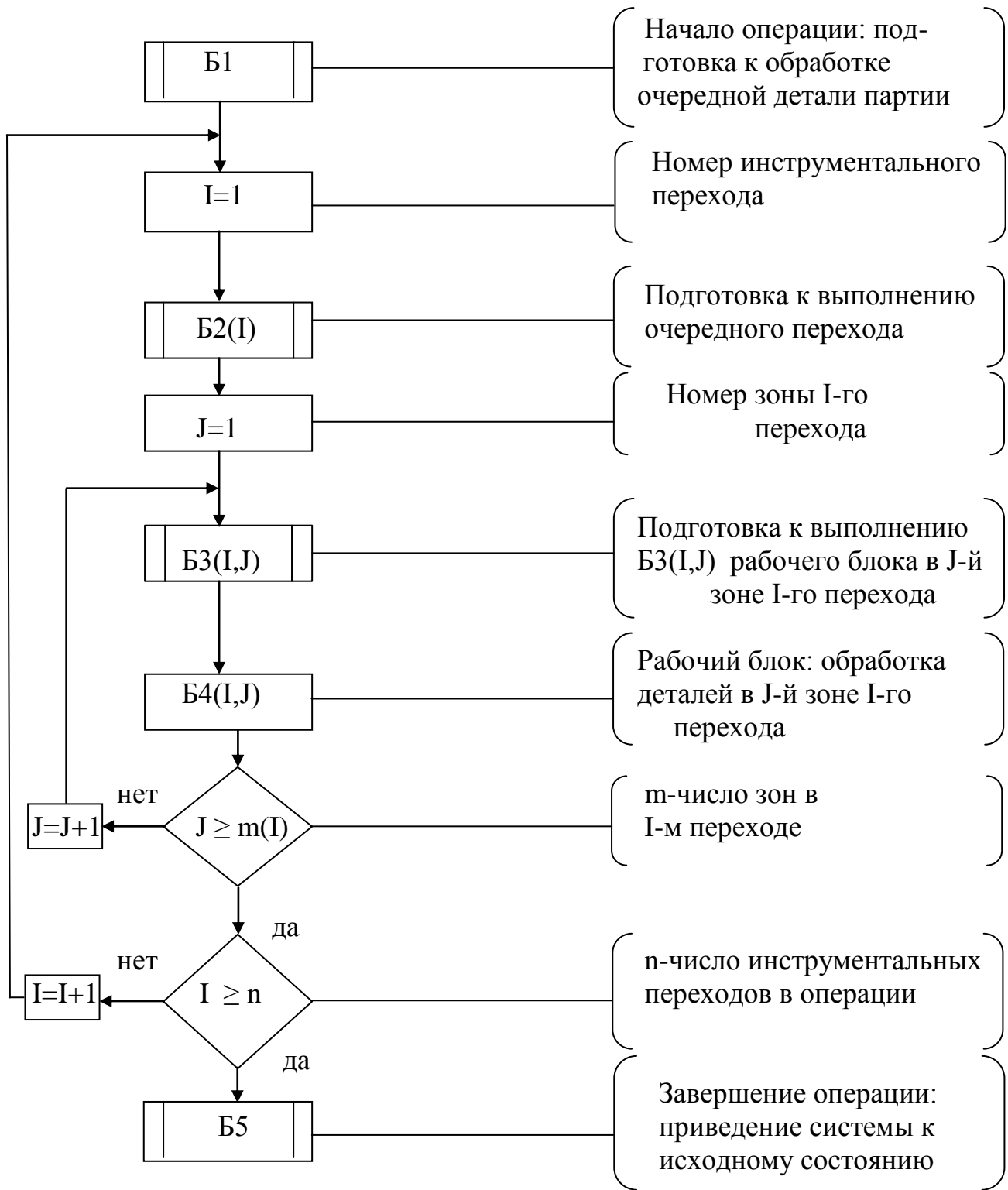


Рисунок 8 - Типовая структура управляющей программы

Таблица 2 – Состав функциональных блоков

Обозначение и наименование блока	Основные функции
Б1: Начало операции	<ol style="list-style-type: none"> <li>1. Установка рабочих органов станка в позицию замены заготовок</li> <li>2. Запуск цикла замены заготовки на рабочем столе станка</li> </ol>
Б2: Начало инструментального перехода	<ol style="list-style-type: none"> <li>1. Установка шпиндельной головки в позицию смены инструмента с отменой ранее введенной коррекции длины инструмента, выключением вращения шпинделя и подачи СОЖ в зону резания</li> <li>2. Запуск цикла смены инструмента</li> <li>3. Установка рабочей координатной системы</li> <li>4. Перемещение шпинделя в Z-плоскость безопасных перемещений с одновременным вводом коррекции длины инструмента</li> </ol>
Б3: Начало зоны	<ol style="list-style-type: none"> <li>1. Ориентация поворотного стола в рабочую позицию</li> <li>2. Установка инструмента по осям X и Y в начальную точку рабочей траектории J-й зоны</li> <li>3. Задание параметров режима резания</li> <li>4. Установка локальной координатной системы J-й зоны с учетом дополнительного ее смещения относительно рабочей системы для компенсации погрешностей расположения базовых элементов приспособления или обрабатываемой заготовки</li> </ol>
Б4: Рабочий блок	<ol style="list-style-type: none"> <li>1. Включение вращения шпинделя и подачи СОЖ в зону резания: (MO3 , MO8)</li> <li>2. Подвод инструмента по оси Z в плоскость рабочих перемещений</li> <li>3. Перемещение инструмента по рабочей траектории. При работе осевым инструментом использовать постоянные длины циклы G81-G89 в последней точке - отменить постоянный цикл (G80) - при фрезеровании - в первом кадре траектории ввести коррекцию G41/42. В предпоследней точке контурной траектории отменить контурную коррекцию (G40)</li> <li>4. Отвод инструмента по оси Z в Z-плоскость безопасных перемещений</li> <li>5. Возврат в рабочую координатную систему</li> </ol>

Б5: Завершение операции	<ol style="list-style-type: none"> <li>1. Отвод рабочих органов в исходную точку</li> <li>2. Запуск цикла выгрузки инструмента из шпинделя в инструментальный магазин</li> <li>3. Завершение УП (М30)</li> </ol>
-------------------------	--

сформулированного выше принципа единства структуры программы, последовательное выполнение которого на всех этапах разработки позволяет "собирать" УП из типовых программных блоков, каждый из которых выполняет конкретные функции и часто может быть оформлен в виде подпрограммы с параметрами фиксированного назначения.

Разработка управляемой части такой программы, содержащая подготовительные блоки (Б1-Б3, Б5 - см. таблицу 2), строится на базе постоянного «макета», требующего лишь заполнения конкретными числовыми параметрами. Рабочие блоки (Б4), обеспечивающие обработку поверхностей заготовки, также могут быть сформированы (частично или полностью) с использованием типовых процедур, содержащихся в библиотеке технологических подпрограмм, формируемой разработчиком.

Следует отметить, что рассмотренная методика проектирования УП на базе типовых решений достаточно просто поддается автоматизации, что создает предпосылки существенного снижения трудоемкости разработки УП.

Для того, чтобы программа могла эффективно эксплуатироваться в производственных условиях, при ее разработке необходимо учитывать ряд дополнительных (эксплуатационных) требований, выполнение которых расширяет функциональные возможности программы, повышает ее надежность, создает удобные средства отладки, размерной настройки и корректировки при изменении условий эксплуатации.

Программа, составленная с учетом эксплуатационных требований, должна обеспечивать:

1. Возможность "неполной" обработки детали (группы деталей), установленной в рабочей зоне станка. Такая возможность оказывается полезной при технологической отладке ("пробная обработка"); при необходимости исправления брака на ранее обработанной заготовке, при эксплуатации программы с неполным комплектом инструментов и т.д. Неполная обработка детали обеспечивается исключением при отработке УП отдельных переходов или технологических зон путем "обхода" определенных групп кадров УП с помощью операторов безусловного перехода или исключения кадров, имеющих поле подтверждения.
2. Автоматическую компенсацию погрешностей изготовления (сборки) инструментов и приспособления, а также погрешностей расположения базовых поверхностей обрабатываемой заготовки. Компенсация



погрешностей базирования заготовки особенно необходима при работе со сменными приспособлениями-спутниками, каждый из которых имеет индивидуальные размерные характеристики. Программная реализация функции компенсации погрешностей обеспечивается операторами смещения координатных осей (UOT) с использованием E-параметров, которым заранее присваиваются значения индивидуальных погрешностей.

## **2.3 Порядок разработки и документирования управляющей программы**

### **2.3.1 Формирование траектории**

Процесс разработки УП может быть представлен тремя последовательными этапами, каждый из которых использует результаты предыдущего и обеспечивает необходимыми данными последующий.

Основное содержание первого этого этапа – формирование траекторий инструментов. В качестве исходных данных на этом этапе используется схема компоновки технологической системы (карта наладки, расчётно-технологическая карта), на которой указывается расположение заготовки или комплекта заготовок в рабочей зоне станка. Пример такой схемы для станка VM501ПМФ4 приведён на рисунке 9. На схеме изображено базовое приспособление (в данном случае - четырёхместное); даны его размеры и расположение базовых элементов в каждой из 4-х зон.

На этой же схеме показана основная - рабочая координатная система (РКС) и локальные координатные системы (ЛКС), определяются смещения осей каждой ЛКС.

Расположение ЛКС зоны выбирается из соображений удобства определения координат опорных точек траекторий инструментов. Рекомендуется связывать ЛКС зоны с элементами приспособлений, по которым базируется обрабатываемая заготовка.

На этой же схеме обозначается положение Z-плоскости безопасных перемещений и Z-плоскости начальных/конечных точек рабочей траектории инструмента в зоне.

Плоскость безопасных перемещений – это Z-плоскость, в которую отводится шпиндель станка для переориентации поворотного стола при переходе к обработке очередной зоне.

Плоскость начальных/конечных точек рабочей траектории – это Z-плоскость, в которую выводится рабочая точка инструмента для перемещения к очередной обрабатываемой поверхности внутри одной зоны.

Результатом 1-го этапа разработки УП являются:

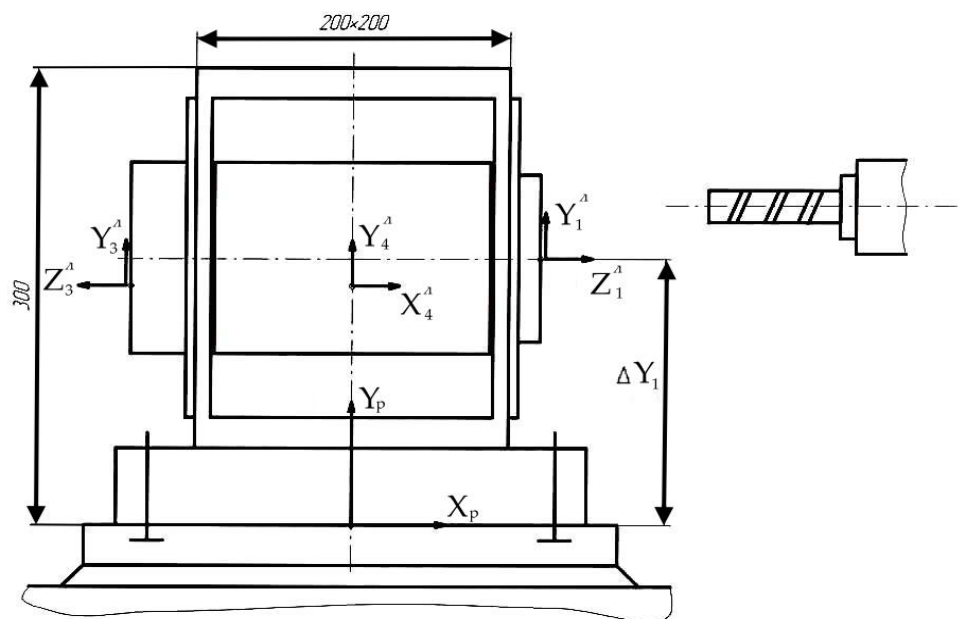
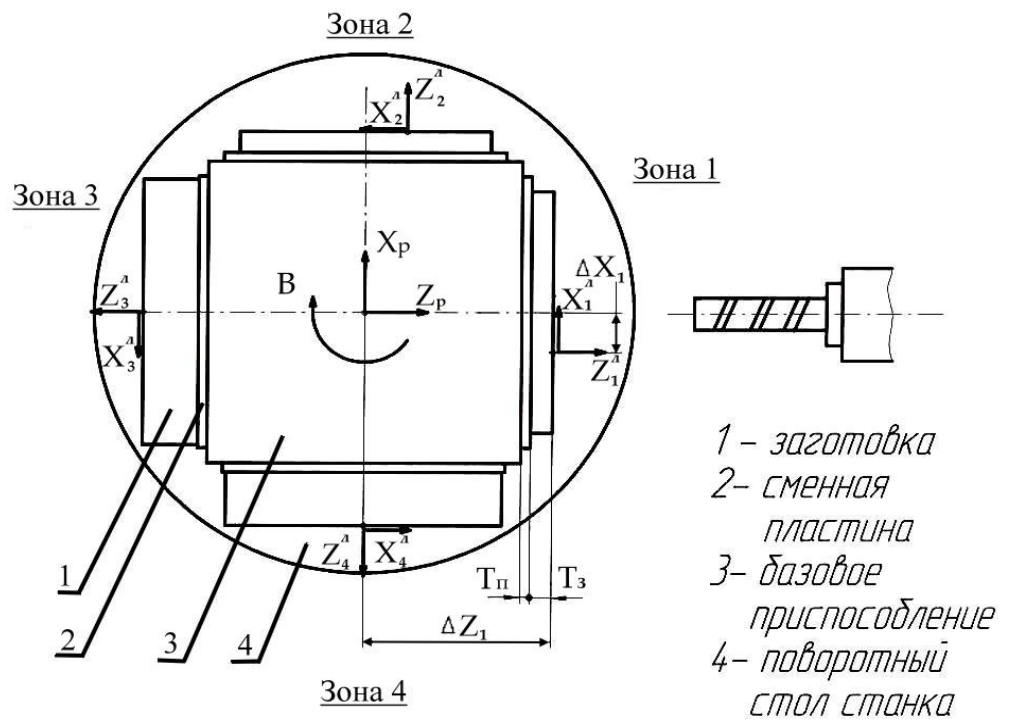


Рисунок 9 – Схема установки заготовок в рабочей зоне станка

1. Схема компоновки технологической системы (рисунок 9), дополненная указанием расположения координатных систем (РКС и ЛКС зон), плоскости безопасных перемещений и плоскостей начальных/конечных точек рабочих траекторий.
2. Таблица переходов и зон, содержащая информацию о последовательности работы инструментов и их основных параметрах, составе зон внутри инструментальных переходов, о положении плоскости безопасных перемещений и плоскостей начальных/конечных точек рабочих траекторий для каждой из зон.  
Рекомендуемая форма такой таблицы приведена на рисунке 10.

Переходы			Инструмент		Перечень номеров зон		
№	Наименование	T	Наименование	Диаметр			
1	Центровать 12 отв.	1	Сверло центр.	-	1,2,3,4		
2	Сверлить отв. 1,3,8	2	Сверло	9	1,3,4		
.	.	.	.	.	.		
N	Фрезеровать паз	12	Фреза концевая	18	2,3		
Безопасная плоскость: z 200 мм							
Зоны:							
№ зоны	В	Смещение ЛКС, мм			Начальная точка рабочей траектории (в ЛКС), мм		
		$\Delta X$	$\Delta Y$	$\Delta Z$	$X_n$	$Y_n$	$Z_n$
1	0°	- 60	80	140	0	0	0
2	90°	- 60	80	140	0	0	0
3	180°	50	120	140	- 50	0	30
4	270°	50	120	140	- 50	0	0

Рисунок 10 - Форма таблицы переходов и зон

### **2.3.2 Второй этап проектирования – разработка структуры управляющей программы и спецификаций входящих программ**

УП строится по иерархической трехуровневой структуре. Число уровней ограничивается допустимой глубиной вложения подпрограмм (для УЧПУ NC210 – не более 2-х).

Верхний уровень – головная программа, обеспечивающая последовательность выполнения инструментальных переходов и технологических зон.

Средний уровень - подпрограммы, реализующие рабочие блоки УП (одна зона – одна подпрограмма), а также вспомогательные подпрограммы, обеспечивающие смену заготовок и инструментов, расчет числа циклов работы инструментов для замены их дублерами, компенсацию погрешностей расположения базовых элементов приспособления и т.д.

Нижний уровень – подпрограммы, реализующие обработку типовых элементов (обработка отверстий, фрезерование пазов, канавок и т.д.).

Завершение 2-го этапа проектирования УП оформляется следующими документами:

1. Структурная схема УП.
2. Спецификация входящих программ .

Пример структуры УП приведен на рисунке 11, пример спецификации входящих программ – на рисунке 12.

### **2.3.3 Третий этап – кодирование управляющей программы**

Третий этап является завершающим в процессе проектирования УП. Его результатом является набор текстов всех программ на входном языке УЧПУ в соответствии со спецификацией, разработанной на втором этапе.

При написании текстов следует руководствоваться правилами, приведёнными в первом разделе настоящего руководства с учетом рекомендуемого содержания функциональных блоков УП, приведённого в таблице 2.

В качестве исходных документов, используемых при кодировании УП, являются все графические и табличные материалы, разработанные на двух предшествующих этапах.



Рисунок 11 - Пример структуры УП

Деталь:		Операция:	
Спецификация входящих программ			
Обозначение	Наименование программы	Используемые Е-параметры	Используемые подпрограммы
-	Головная программа	E0 – E10 E60; E80	PROG1, PROG11 PROG12, PROG21 PROG22
PROG1	Подпрограмма смены инструмента	E0, E80	-
PROG11	Подпрограмма обработки 1-й и 2-й зоны 1-го перехода	-	PROG81
PROG12	Подпрограмма обработки 1-й и 2-й зоны 2-го перехода	-	PROG81
PROG21	Подпрограмма обработки 3-й и 4-й зоны 1-го перехода	-	
-	-	-	-
-	-	-	-
PROG81	Подпрограмма сверления	E11, E12	

Рисунок 12 - Пример спецификации входящих программ

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Устройство числового программного управления NC110, NC200, NC210: Руководство программиста.- СПб: Балт-Систем, 2004.- 173 с.
2. Сосонкин В.Л. Системы числового программного управления.- М.: Логос, 2005.- 293 с.
3. Сосонкин В.Л. Программное управление технологическим оборудованием.- М.: Машиностроение, 1991.- 512 с.

Сбродов Николай Борисович

**ПРОЕКТИРОВАНИЕ УПРАВЛЯЮЩИХ ПРОГРАММ ДЛЯ  
МНОГООПЕРАЦИОННЫХ СТАНКОВ С ЧПУ**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

для самостоятельной работы по дисциплинам  
«Программное управление технологическим оборудованием»,  
«Управление системами и процессами», «Управление  
станками и станочными комплексами»  
студентов специальностей  
220301 – «Автоматизация технологических  
процессов и производств (в машиностроении)»,  
151001 – «Технология машиностроения»,  
151002 – «Металлообрабатывающие станки  
и комплексы»

Редактор – Н.М.Кокина

---

Подписано к печати	Формат 60x84 1/16	Бумага тип. №
Печать трафаретная	Усл.печ.л. 2,0	Уч.-изд.л. 2,0
Заказ	Тираж 150	Цена свободная

---

Редакционно-издательский центр КГУ.  
640669, г. Курган, ул. Гоголя, 25.  
Курганский государственный университет.