



Системы управления CNC8X9 - DUAL

ПРОИЗВОДСТВО И СЕРВИС

MEFI, s.r.o.
Peroutkova 37
150 00 PRAHA 5

tel: 251 045 113
fax: 251 045 112
e-mail: mefi@mefi.cz
[http: //www.mefi.cz](http://www.mefi.cz)



Инструкция по программированию

Содержание

1. ВВЕДЕНИЕ В ПРОГРАММИРОВАНИЕ.....	1-1
1.1. Основные понятия	1-1
1.2. Код входной информации	1-2
2. СТРУКТУРА ПАРТПРОГРАММЫ.....	2-1
2.1. Слово партпрограммы	2-1
2.1.1. Структура слова	2-1
2.1.2. Запись цифры - величины	2-1
2.1.3. Виды слов	2-2
2.2. Блок партпрограммы	2-2
2.2.1. Следующие инструкции по составлению программного блока	2-6
2.3. Создание партпрограмм, подпрограмм, макроциклов и твердых циклов	2-6
2.3.1. Начало партпрограммы	2-7
2.3.2. Конец партпрограммы	2-7
2.3.3. Подпрограмма (PPRG).....	2-7
2.3.4. Пример партпрограммы с подпрограммами.....	2-8
2.3.5. Макроцикл (MRC).....	2-9
2.3.6. Скачок в партпрограмме, макроцикле	2-10
3. ПРОГРАММИРОВАНИЕ КООРДИНАТ.....	3-1
3.1. Система координат – обозначение координат.....	3-1
3.2. Свойства одновременно управляемых координат	3-1
3.3. Введение траектории	3-2
3.4. Нулевые точки станка и программы	3-3
3.4.1. Определение нулевых точек станка (NBS).....	3-3
3.4.2. Референционная точка станка.....	3-4
3.4.3. Нулевая точка программы.....	3-4
3.4.4. Перемещение нулевых точек (группа G5).....	3-4
3.4.5. Заполнение таблицы сдвига начала.....	3-5
3.4.6. Заполнение таблицы сдвига из партпрограммы функциями G92 и G93.....	3-5
3.4.7. Заполнение таблицы сдвига с редактора файла TAB0.POS.....	3-7
3.4.8. Заполнение таблицы сдвига интерактивным вводом	3-7
3.4.9. Вызов сдвигов нулевых точек.....	3-7
4. ВИДЫ ДВИЖЕНИЯ.....	4-1
4.1. Установка координат - функция G00	4-1
4.2. Линейная интерполяция - функция G01	4-1
4.3. Круговая интерполяция - функция G02, G03	4-2
5. РЕЗКА РЕЗЬБЫ НОЖОМ	5-1
5.1. Резка резьбы без выхода из траектории	5-1
5.2. Запрограммирование резьбы с выходом из траектории	5-2
5.2.1. Угол выхода из резьбы	5-2
5.3. Резка резьбы с въездом и выходом (выходом из траектории)	5-3
5.4. Резка резьбы с большим шагом	5-5
6. ВВЕДЕНИЕ СДВИГА	6-1
6.1. Постоянная скорость резки (KRR) G96 и G97	6-2
7. КОРРЕКЦИЯ ИНСТРУМЕНТА.....	7-1
7.1. Файл TAB0.KOR и таблица коррекций в запоминающем устройстве.....	7-1
7.2. Коррекция радиуса с эквидистантой.....	7-3
7.2.1. Включение коррекции радиуса и ход коррекции	7-5
7.2.2. Выражение коррекции радиуса	7-7
7.2.3. Предельный угол для вкладывания обрабатываемых продуктов.....	7-8
7.3. Коррекция радиуса для токарных станков	7-9
7.4. Индикация для токарных станков при использовании коррекции радиуса	7-11
7.5. Управление скоростью в случае коррекций диаметра	7-13

7.6. Тест неразрывности для коррекции радиуса	7-14
7.7. Уточнение круговой интерполяции	7-15
7.8. Выбор плоскостей коррекции и интерполяции	7-16
7.9. Коррекция длины	7-17
7.9.1. Коррекция на длину инструмента	7-17
7.9.2. Коррекция длины – способ управления А	7-17
7.9.3. Коррекция длины – способ управления В	7-19
7.9.4. Коррекция длины – способ управления С	7-19
8. ПРОГРАММИРОВАНИЕ ТЕХНОЛОГИЧЕСКИХ ФУНКЦИЙ.....	8-1
8.1. Временная задержка	8-1
8.2. Обороты шпинделя	8-1
8.2.1. Стоп шпинделя в ориентировочной точке и управление в связи с положением	8-2
8.3. Охлаждение инструмента.....	8-2
8.4. Прерывание и конец партпрограммы	8-2
8.5. Укрепление и освобождение обрабатываемого продукта.....	8-3
8.6. Смена инструмента и обрабатываемого продукта	8-3
8.7. Вспомогательные М-функции группы M14	8-3
8.8. Вспомогательные М-функции групп M10, M11, M12, M13	8-3
8.9. Номер инструмента - функция Т	8-3
8.10. Вспомогательные функции Н, Р	8-3
9. ПАРАМЕТРИЧЕСКОЕ ПРОГРАММИРОВАНИЕ - ФУНКЦИЯ R.....	9-1
9.1. Декларация параметров	9-1
9.2. Вызов параметров	9-2
9.3. Десятичная точка в декларации параметров	9-3
10. ВИНТОВАЯ ЛИНИЯ	10-1
11. АРИФМЕТИКА ПАРАМЕТРОВ.....	11-1
11.1. Общие принципы	11-1
11.2. Управление двойной точностью.....	11-1
11.3. Структура параметров управления.....	11-2
11.4. Пример использования арифметики параметров	11-6
12. ПРОГРАММИРОВАНИЕ И ОПРЕДЕЛЕНИЕ ТВЕРДЫХ ЦИКЛОВ.....	12-1
12.1. Создание твердых циклов	12-1
12.2. Примеры твердых циклов сверления для фрезеровки	12-2
12.3. Возможности обработки твердых циклов.....	12-7
12.4. Твердые циклы для токарных станков	12-8
12.5. Примеры твердых циклов для токарных инструментов.....	12-9
12.5.1. Продольная черновая обработка - G81	12-9
12.5.2. Пример использования твердого цикла G81 для продольной черновой обработки	12-11
12.5.3. Поперечная черновая обработка - G82.....	12-12
12.5.4. Сверление глубоких дыр - G83.....	12-14
12.5.5. Резка резьбы на цилиндрической плоскости - G84.....	12-16
12.5.6. Резка резьбы на цилиндрической плоскости с выездом - G85.....	12-21
12.5.7. Снятие фаски под углом - G86.....	12-25
12.5.8. Резка резьбы для шаровой плоскости - G87	12-27
13. ПЕРЕСЧЕТ КООРДИНАТ	13-1
13.1. Общие принципы	13-1
13.2. Отражение	13-2
13.3. Изменение масштаба	13-4
13.4. Независимый сдвиг	13-6

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А – ПЛАВНАЯ СВЯЗЬ МЕЖДУ БЛОКАМИ.....A-1

Приложение А1 – соединение блоков без плавной связиA-1

Приложение А2 – плотное соединение блоковA-2

Приложение А3 – ручное управление плавным соединением.....A-2

Приложение А4 – автоматическое определение плавного соединенияA-4

Приложение А5 – графическая диагностика плавного соединенияA-5

Приложение А6 – комплексная скоростьA-7

Приложение А6.1 – длина блоков для повышения скорости.....A-10

Приложение А6.2 – критерий точностиA-11

Приложение А6.3 – динамический критерийA-13

ПРИЛОЖЕНИЕ В – ПРИОРИТЕТНЫЙ БЛОК.....B-1

Приложение В1 – установка приоритетного блокаB-2

1

1. ВВЕДЕНИЕ В ПРОГРАММИРОВАНИЕ

1.1 Основные понятия

Для числовых управляемых машин к передаче информации об обрабатываемом полуфабрикате или его части используется кодированная запись называемая **партпрограмма** (часть программы).

В современных системах последовательность этой информации записывается в дублированное запоминающее устройство. В управляющих системах MEFI CNC836/CNC846/CNC856 (далее только CNC8x6) составляет вместимость дублированного запоминающего устройства для партпрограммы приблизительно несколько сот Кбайтов в системе оснащенной памятью CMOS (уже не поставляется), десятки Мбайтов в системе оснащенной флеш диском сотни Мбайтов или несколько Гбайтов в системе оснащенной постоянным диском (хард диском).

Кроме партпрограмм в дублированное запоминающее устройство записаны также таблицы коррекций, сдвиги начала и параметры. Партпрограммы (и их логические части, т.е. подпрограммы, макроциклы, постоянные циклы) в память записаны в форме, которую система может в течение управления в реальном времени легко обрабатывать и переводить в мгновенные инструкции для управления машиной.

Партпрограмма, созданная технологом и записанная на подходящем медиуме (дискетка, серийный коммуникационный адаптер, диск вышестоящего компьютера и т.п.), потом внешним устройством вводится в авансированное запоминающее устройство системы. Партпрограмму можно создать или машинной обработкой на компьютере после введения основных геометрических и технологических данных и параметров станка (AUTOPROG, KOVOPROG, APT и т.п.), или вручную расчетом окончательных точек всех элементарных движений инструмента и дополнением соответствующих технологических функций. Созданную партпрограмму вводим в запоминающее устройство системы (см. Инструкцию по обслуживанию). Партпрограмму можно записать в запоминающее устройство системы также прямо с пульта управления.

Кодированная запись геометрии и технологии создают части партпрограммы - последовательности допустимых знаков (т.наз. адресов). Эти последовательности знаков должны однозначно описывать порядок обработки на конкретном станке, они должны быть однозначно идентифицированы, как комплект, и в форме выписки на принтере или на экране дисплея они должны быть легко внятыми и разборчивыми.

При создании партпрограммы необходимо исходить из следующих данных:

- a) геометрия станка (координатная система, направление осей, нулевые точки)
- b) геометрия полуфабриката (возможность коллизии обрабатываемого полуфабриката с инструментом в течение обработки, размещение обрабатываемого полуфабриката в системе координат станка)
- c) геометрия инструмента (размеры, форма, коррекция траектории инструмента в форме обрабатываемого полуфабриката)
- d) геометрия окончательного продукта (определена по чертежу частей)
- e) технологические и режущие условия (скорость резки, толщина стружек и т.п.)
- f) остальные условия, важные для деятельности станка (охлаждение, величина сдвига, обороты, номера

инструментов, коррекция и т.п.)

Устройство партпрограммы должно потом удовлетворять предписанному синтаксису записи, чтобы была гарантированная однозначность выражения.

Партпрограмма состоит из ряда отдельных элементарных операций, т.наз. **блоков**. Блок состоит из самостоятельных данных, т.наз. **слов**. Каждое слово потом содержит (за исключением некоторых слов) **адреса**, определяющие вид информации и **нумерическое обозначение**, определяющее величину размера, или кодовое определение любой функции или операции. Один блок может быть расположен в нескольких строках, следующий блок определяется только адресом N!

Пример блока партпрограммы:

N20 G1 X10.355 Z625.50 F300 S150 T12 M0

Словами в этом блоке являются: N20, G1, X10.355, Z625.50, F300, S150, T12, M0

Адресами являются: N, G, X, Z, F, S, T, M

Нумерическим обозначением является: 20, 1, 10.355, 625.50, 300, 150, 12, 0

Пример расположения одного блока (N10) в нескольких строках:

N10 R1=100.

R2=200.

R3=300.

N20 G54 G0 X0 Y0 Z0

1.2 Код входной информации

Система принимает входную информацию партпрограммы в форме текста, т.е. без паритета. Случайные диакритические знаки чешского языка могут появляться только в комментарии партпрограммы. Если должен чешский язык в комментариях правильно изображаться, должен в коде находиться код Каменицких. Адреса, т.е. знаки, которые система обрабатывает, указаны в таблице 1.



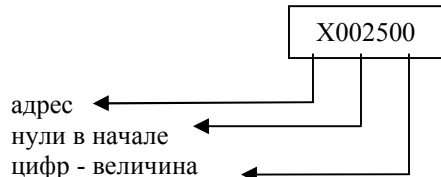
2. СТРУКТУРА ПАРТПРОГРАММЫ

2.1 Слово партпрограммы

2.1.1 Структура слова

Элементарным структурным элементом партпрограммы является т. наз. слово (инструкция программы). Каждое слово состоит из буквы адреса и одной или нескольких цифр номера с содержанием адреса со случайным знаком и (факультативным) десятичным знаком.

Пример:



2.1.2 Запись цифры - величины

- a) В записи величины вспомогательных функций могут пропускаться нули в начале до первой значительной цифры.

Напр.: F 1.4 S 35 и т.п.

- b) В записи величин координат без десятичного знака могут также пропускаться нули в начале до первой значительной цифры, данные обозначают траекторию в мм.

Напр.: X 3620 = 3620 мм

- c) В записи величин координат с десятичным знаком могут пропускаться нули в начале до первой значительной цифры перед десятичным знаком и в конце после последней значительной цифры за десятичным знаком.

Напр.:

X 36.12 = 36.12 мм

X 0.2 = 0.2 мм

X 384. = 384 мм

X .12 = 0.12 мм

- d) Знак "-" должен быть указан, но знак "+" не обязательно указывать.
 e) Если слово обладает нулевой информацией, должен быть записан минимум один ноль.
 f) Между адресом и нумерической величиной слова может, но не обязательно, находиться один ноль, или несколько пробелов. Наоборот **нумерическая величина, включая знаки, не должна быть прервана пробелом.** В случае параметрического программирования нумерическая величина

заменяется параметром (см. раздел о параметрическом программировании).

<i>Напр.:</i>	
X36.12	правильно
X 36.12	правильно (пробел за адресом)
X36. 12	ошибка (пробел в середине величины)
X- 36.12	ошибка (пробел за знаком -)

2.1.3 Виды слов

Согласно качеству информации, носителями которых являются отдельные слова партпрограммы, разделяем слова и их адреса на две основные группы, т.е. «с размером» и «без размера».

- а) слова с размером – выражают кроме величины также размер. Могут быть положительными и отрицательными, с десятичным знаком или без него. В эту группу входят адреса: A,B,C,X,Y,Z,U,V,W,I,J,K,F,R и адрес Q для временной задержки.
- б) слова без размера – выражают только величину. В эту группу входят адреса: D,G,M,S,T,P,H и адрес Q для числа повторов.

Некоторые слова партпрограммы система запоминает постоянно, т.е. их действительность затрагивает все следующие блоки партпрограмм, пока не записана в этот же адрес другая величина. Вторая группа состоит из слов, которые действительны только в блоке, в котором были запрограммированы.

2.2 Блок партпрограммы

Отдельные слова в партпрограмме составляются в программе блока. Программный блок состоит из основного информационного оборудования, в котором записаны данные о геометрии и технологии обработки. Формат блока обладает переменной длиной.

Максимальный формат блока обладает формой:

Пометка:

В скобках указано макс. количество цифр данного адреса. Цифры перед адресом G и M указывают количество групп.

N	(8)	11G	(2)	D	(2)
X	(+5.3)	Y	(+5.3)	Z	(+5.3)
4.	(+5.3)	5.	(+5.3)	6.	(+5.3)
I	(+5.3)	J	(+5.3)	K	(+5.3)
R	(+5.3) (+8)	&	(4)	F	(2..3) (5)
S	(4)	L	(4)	Q	(4)
H	(2)	P	(2)	T	(8)
10M	(2)	"КОММЕНТАРИЙ" или LF			

Значение адресов и им соответствующих величин указано в следующей таблице (словарь системы):

Адрес	Группа	Величина	Значение
%		от 1 до 999999	Номер партпрограммы
A		от 1 до 99999999	Номер блока партпрограммы, подпрограммы или макроцикла
/		---	Пропуск блока в режиме "/"
G	G0	00 01 02 03 30 33	Установка скоростным сдвигом Линейная интерполяция Круговая интерполяция CW Круговая интерполяция CCW Установка референции из программы Резка резьбы
	G1	17+ 18 19 14 15 16	Выбор плоскости XY (только для коррекции... Выбор плоскости YZ ... на диаметр инструмента) Выбор плоскости ZX (G41 и G42) Выбор плоскости Z4 (см. R340, R341) Выбор плоскости Y4 Выбор плоскости 4X
	G2	05 06 07 09 10 24 23 98	Копирование по образцу Копирование из запоминающего устройства Копирование в запоминающее устройство Винтовая линия (спираль) Определение шага спирали Прочное соединение блоков Плавная последовательность блоков Основное состояние группы 2
	G3	40+ 41 42	Отмена коррекции радиуса Коррекция радиуса налево Коррекция радиуса направо
	G4	26 27 28 29	Арифметические операции – 1. формула Арифметические операции – 2. формула Арифметические операции – 3. формула Арифметические операции – 4. формула
	G5	53 54 55 56 57 58x 59x	Сдвиг 0. Нулевой точки Сдвиг 1. Нулевой точки Сдвиг 2. Нулевой точки Сдвиг 3. Нулевой точки Сдвиг 4. Нулевой точки Сдвиг 5. Нулевой точки Сдвиг 6. Нулевой точки
	G6	94+ 95 96 97+	Сдвиг в мм/мин без KRR (постоянная скорость резки) Сдвиг в мм/об. без KRR Сдвиг в мм/мин с KRR Сдвиг в мм/об. с KRR
	G7	70+ 71x 72x 73x 79	Конец макроцикла или подпрограммы Вызов подпрограммы – в блоке должен быть одновременно запрограммирован адрес L Вызов макроцикла – в блоке должен быть одновременно запрограммирован адрес L Переход на программный блок, номер которого указывает величина - адрес L Ввод в макроцикл или подпрограмму Функция должна быть запрограммирована вместе с адресом L
	G8	80+ от 81 до 89	Отмена постоянного цикла Постоянные циклы по норме ISO

	G9	90+ 91	Ввод абсолютных величин координат Ввод повышающихся величин координат
	G10	04x 92x 93x 74 75	Временная задержка – длительность определена функцией Q. Дополнение величин в таблицы начала для данной G-функции из группы G5. Дополнение величин в таблицы коррекции. Дополнение величин сдвига начала для данной G-функции из группы G5, при которых произойдет отождествление нулевой точки программы для G-функции из группы G5 с точкой координатной системы, которая находится на расстоянии от действительного положения инструмента на величину запрограммированных координат в данном блоке. Программирование диаметра Программирование радиуса
	G11	34 35 36 37	Отмена всех пересчетов координат. Отражение, переводит направление в выбранных осях. Изменение масштаба. Независимый сдвиг.
&		от 0000 до 2222	Функция управления, определяющая соединение коррекции длины с соответствующими координатами в соответствующем направлении.
D		от 1 до 99	Номер статьи в таблице коррекции.
X		+69999.999 -69999.999	Ввод траектории в мм.
Y		+69999.999 -69999.999	Ввод траектории в мм.
Z		+69999.999 -69999.999	Ввод траектории в мм.
4.		+69999.999 -69999.999	Ввод траектории в мм.
5.		+69999.999 -69999.999	Ввод траектории в мм.
6.		+69999.999 -69999.999	Ввод траектории в мм.
I		+69999.999 -69999.999	Расстояние центра окружности в направлении первой интерполяционной координаты
J		+69999.999 -69999.999	Расстояние центра окружности в направлении второй интерполяционной координаты
R0 - R95		+69999.999 -69999.999	Размер параметра соединенного с отдельными координатами или технологическими функциями.
Q		от 1 до 9999	Количество повторов программы для G71, макроцикла для G72 или переходов на программируемый блок для G73. Размер временной задержки по 10мс при введении функции G04.
F		от 1 до 24 000 для сдвига в мм/мин. от 1 до 99 999 для сдвига в мм/об.	Сдвиг в мм/мин. или в мм/об.

L		от 1 до 9999	Номер выведенной подпрограммы или макроцикла при функции G71, G72, G79. Номер блока, на котором осуществляется переход при функции G73.
H		от 1 до 99	Вспомогательная функция
P		от 1 до 99	Вспомогательная функция
S		от 1 до 9999	Обороты шпинделя
T		от + до 69999999	Номер инструмента
M	M1	00 ^x 01 ^x 02 ^x 30 ^x	Запрограммированный стоп Выбираемый стоп (СТОП в режиме M01) Конец партпрограммы обратным ходом в начало партпрограммы. Пометка: установит M-функции по группам M2, M5, M6 и M8 на величину M функций, обозначенных +. дт-то см. M02
	M2	03 04 05 ⁺ 19(+)	Старт шпинделя "CW" Старт шпинделя "CCW" Стоп шпинделя Стоп шпинделя в ориентировочной точке
	M3	41 42 43 44 40 ⁺	Обороты шпинделя диапазон 1 Обороты шпинделя диапазон 2 Обороты шпинделя диапазон 3 Обороты шпинделя диапазон 4 Диапазон оборотов шпинделя рассчитан прямо из функции S
	M4	36 ⁺ 37	Сдвиг запрограммирован прямо Сдвиг уменьшен 1 : 100
	M5	07 08 09 ⁺ 17	Включить охлаждение 2 Включить охлаждение 1 Выключить охлаждение 1 и 2 Включить охлаждение 1 и 2
	M6	50 51 53 ⁺ 52	Включить охлаждение 3 Включить охлаждение 4 Выключить охлаждение 3 и 4 Включить охлаждение 3 и 4
	M7	10 ^x 11 ^x	Укрепление обрабатываемого полуфабриката Ослабление обрабатываемого полуфабриката
	M8	49 48 ⁺	Переход ручного FEED OVERRIDE Отмена перехода (включение) FEED OVERRIDE
	M9	06 ^x 60 ^x	Смена инструмента Смена обрабатываемого полуфабриката
	M10	10 - 99 ⁺	Функция согласно машинной константе R56
	M11	10 - 99 ⁺	Функция согласно машинной константе R57
	M12	10 - 99 ⁺	Функция согласно машинной константе R58
	M13	10 - 99 ⁺	Функция согласно машинной константе R59
	M14	10 - 99 ⁺	Все вспомогательные функции, которые не были в остальных группах указаны
*			Конец чтения партпрограммы включая ее подпрограммы

Объяснение знаков использованных в таблице:

x – Вспомогательная функция действительна только в блоке, в котором была указана.

+ - Номер функции, которая отменяет указанные номера в соответствующих группах, и которые возникнут временно в группах M2, M5 и M6, в случае функций M00, M01, также постоянно во всех

группах в случае функций MO2, M3O и после обнуления системы.

Пометка: Случайные следующие M-функции, не указанные в этом списке, должен поставить автор PLC программы для конкретного станка.

2.2.1 Следующие инструкции по составлению программного блока

- a) За адресом 4. или 5. можно запрограммировать письменный знак A,B,C,U,V,W (по договору с производителем станка).
- b) За адресом 6. можно запрограммировать письменный знак A,B,C, (по договору с производителем станка).
- c) КОММЕНТАРИЙ является последовательностью любых знаков (кроме кавычек), закрытой между кавычками. Если за комментарием указан знак конца строки (CR, LF), не должны быть указаны окончательные кавычки.

Пример:

```
N10 X100 Y100 " ЭТО КОММЕНТАРИЙ "
N20 X200 Y200 " ЭТО КОММЕНТАРИЙ
" "
"
N30 X100 "ЭТО КОММЕНТАРИЙ" Y200 Z300
N40 X100 "ЭТО КОММЕНТАРИЙ Y200 Z300
```

Пометка:

В блоке N40 в комментарий включается также Y200 Z300, или если не указаны вторые кавычки, считается концом комментария конец строки!

- d) Каждый запрограммированный блок должен начинаться с адреса N - (номер блока). Блок заканчивается со следующего знака N (номер следующего блока) или с конца файла (может быть указана *, которая, однако, обязательна только во время чтения по серийному каналу). Остальная очередность слов в одном блоке любая. Рекомендуется, однако, использовать стандартную очередность слов (см. максимальный формат блока).
- e) Система работает с переменной длиной блока. Это означает, что в каждом блоке может находиться любое количество слов. (В запоминающее устройство будет также записан пустой блок, обозначенный только номером блока, что, однако, не имеет практического значения). Каждый адрес (слово) может быть записан в одном блоке только один раз за исключением групповых функций (M,G), у которых может быть записана одна величина из каждой группы, и адреса R, который может в блоке использоваться несколько раз (макс. 95 раз).
- f) Количество всех знаков программного блока (включая разделители и комментарии) не ограничено. На строке (между знаками CR, LF), однако, может быть максимально 80 знаков.
- g) Вместо величины адреса возможно во все адреса кроме N записывать номер параметра R – см. раздел о параметрическом программировании.
- h) Как разделитель между отдельными словами блока можно использовать свободное количество пробелов. Пробелы могут быть более этого использованы также внутри слова, однако, только между адресом и номером (никогда не внутри нумерической величины!).
- i) Знак "/" (дробь) это единственный адрес без величины (номера за ним последующие не оцениваются). Во время выбора режима "/" (отмена блоков) блок, в котором указана "/", во время обработки партпрограммы игнорируется. Знак дробь должен быть указан за номером блока, которого он касается. Если был бы указан до знака N, был бы действителен для предыдущего блока.
- j) Если в блоке запрограммирована временная задержка (функцией G04) совместно с некоторой функцией из группы G7, действительна функция Q преимущественно для временной задержки. В этом случае количество подпрограмм, макроциклов или скачков используется только один раз (одинаково как во время Q=1).

2.3 Создание партпрограмм, подпрограмм, макроциклов и твердых циклов

Партпрограмма данной части обычно состоит из части собственной партпрограммы и из части т. наз. подпрограмм. Подпрограммы относятся только к данной партпрограмме, за концом которой должны быть непосредственно записаны. Партпрограмма не должна содержать подпрограммы.

2.3.1 Начало партпрограммы

Партпрограмма должна начинаться с номера партпрограммы. Он указана адресом % и максимально шестидекадным положительным не нулевым номером без десятичного знака, который является идентификатором партпрограммы в случае автоматического хода. Этот номер изображается во время хода партпрограммы в форме листинга, другое значение не имеет. Этот номер партпрограммы не должен соответствовать названию файла, в котором партпрограмма записана в дублированное запоминающее устройство (на диске или в запоминающем устройстве CMOS), хотя годится, чтобы название файла с партпрограммой и номер партпрограммы совпадали. Если за номером партпрограммы не указан никакой комментарий, изобразится в форме листинга кроме номера партпрограммы для информации также название файла в квадратных скобках. Если за процентом указан комментарий, изобразится этот комментарий.

Пример:

```
%1  
%123456 "КОММЕНТАРИЙ:  
%200
```

Если в течение чтения партпрограммы из входного оборудования не найден знак %, знаки игнорируются, т.е. до знака % может находиться любой знак кроме %.

2.3.2 Конец партпрограммы

Логический конец партпрограммы определен программированием функции M02 или M30 в последнем блоке собственной партпрограммы. За блоком с функцией M02 или M30 могут быть в партпрограмме запрограммированы только блоки отдельных подпрограмм, которые "вызываются" содержанием партпрограммы.

Физическое окончание всей партпрограммы, включая случайные подпрограммы, должно быть обозначено знаком * (звездочка).

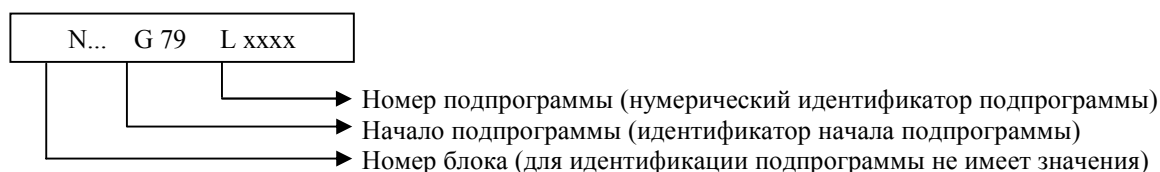
Пометка:

Во время записи партпрограммы в редактор системы звездочка не записывается, система ее дополнит сама автоматически. Звездочку не надо в партпрограмму записывать практически только в случае, если партпрограмма начинается с серийного канала.

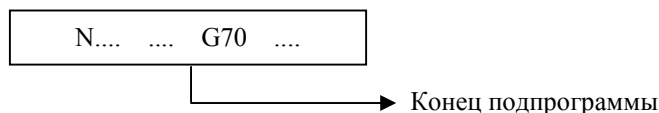
2.3.3 Подпрограмма (PPRG)

Под подпрограммой понимается определенная группа программных блоков, которые обладают стандартным входным и окончательным блоками. Подпрограмма логически принадлежит только к данной партпрограмме и может быть вызвана только из этой партпрограммы.

Каждая подпрограмма должна обязательно начинаться с блока, в котором кроме номера блока указана только функция G79 и адрес L с величиной, показывающей номер подпрограммы. Номер блока не имеет для идентификации подпрограммы никакого значения. Форма входного блока подпрограммы является следующей:

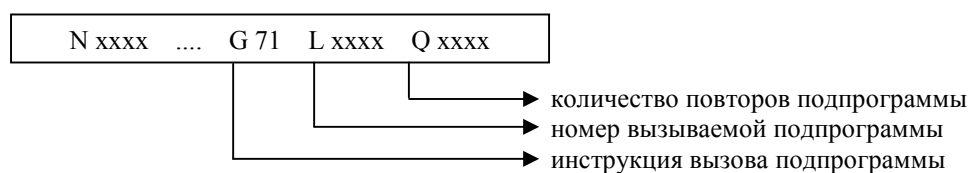


Подпрограмма должна быть закончена блоком, в котором указана функция G70. Для последнего блока подпрограммы разрешено программировать также другие функции, или движение. Последний блок подпрограммы имеет следующую форму:



Во время записи партпрограммы должны быть соответствующие подпрограммы записаны совместно с партпрограммой, к которой принадлежат (прямо за блоком, в котором запрограммирована функция M02 или M30), или если должны быть в одинаковом файле.

Максимальное количество подпрограмм в партпрограмме составляет 99. В соответствующей партпрограмме, которая начинается со знака % и заканчивается знаком * (собственная партпрограмма со случайными подпрограммами) не должны быть два или более программных блоков обозначены одинаковым номером блока (величина адреса N). Вызов подпрограммы можно выполнить в любом блоке собственной партпрограммы. Вызов подпрограммы имеет следующую форму:



Количество повторов подпрограмм, программируемое функцией Q, не должно быть указано. Потом подпрограмма выполняется только один раз (одинаково как во время программирования Q1). После окончания подпрограммы останутся величины технологических и вспомогательных функций в таком состоянии, как их установила подпрограмма. Подпрограмму можно редактировать в одинаковом объеме как собственную партпрограмму. В программном блоке партпрограммы, с которой вызывается подпрограмма, допустимо программировать также остальные технологические функции и функции сдвига. Собственный скачок на выбранную подпрограмму будет выполнен в заключение блока, т.е. после выполнения остальных программируемых операций. После выполнения подпрограммы сверления вернется на следующий блок партпрограммы, с которого был скачок выполнен.

2.3.4 Пример партпрограммы с подпрограммами:

%35	" НАЧАЛО ПАРТПРОГРАММЫ НОМЕР 35 "
N10 X0 Y0 Z0 G54	" ПЕРВЫЙ БЛОК ПАРТПРОГРАММЫ "
N20 X200 G1 F300	" ВТОРОЙ БЛОК ПАРТПРОГРАММЫ "
....	
....	
N80 G71 L100 Q2	" ВЫЗОВ ПОДПРОГРАММЫ НОМЕР 100 "
....	" ПОДПРОГРАММА БУДЕТ ВЫЗЫВАТЬСЯ ДВА РАЗА "
....	
N120 G71 L200	" ВЫЗОВ ПОДПРОГРАММЫ НОМЕР 200 "
....	
....	
N310 X0 Y0 Z0 G00	" ТЕКУЩИЙ БЛОК ПАРТПРОГРАММЫ "
N320 M30	" ЛОГИЧЕСКОЕ ОКОНЧАНИЕ ПАРТПРОГРАММЫ "

```

"
" СЛЕДУЮТ ДВЕ ПОДПРОГРАММЫ
"
" ПОДПРОГРАММА НОМЕР 100 "
"
N500 G79 L100      " ПЕРВЫЙ БЛОК ПОДПРОГРАММЫ НОМЕР 100 "
N510 ...
N520 ...
....
....
N590 G70           " ПОСЛЕДНИЙ БЛОК ПОДПРОГРАММЫ НОМЕР 100 "
"
" ПОДПРОГРАММА НОМЕР 200 "
"
N600 G79 L200      " ПЕРВЫЙ БЛОК ПОДПРОГРАММЫ НОМЕР 200 "
N610 ...
N620 ...
....
....
N700 G70           " ПОСЛЕДНИЙ БЛОК ПОДПРОГРАММЫ НОМЕР 200 "
*                  " ЗВЕЗДОЧКА – ФИЗИЧЕСКИЙ КОНЕЦ ПАРТПРОГРАММЫ"

```

2.3.5 Макроцикл (MRC)

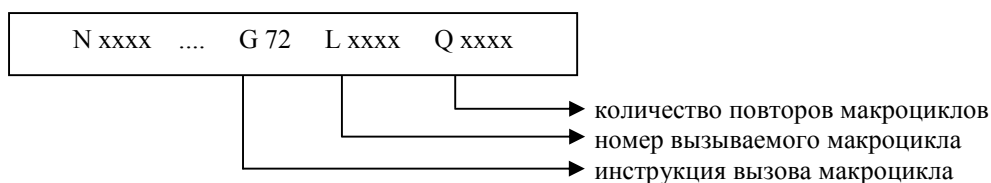
Под макроциклом понимается определенная группа программируемых блоков, из которых состоит партпрограмма, для типичной части или части производимой на конкретном станке.

Каждый макроцикл должен начинаться и заканчиваться блоком, который имеет одинаковую форму как блок для начала и конец подпрограммы. Макроцикл, в общем, относится ко всем уложенным партпрограммам, из которых может быть вызван (может быть вызван из собственных партпрограмм и ее подпрограмм).

Макроциклы принадлежат в т. наз. группу библиотечных партпрограмм и в системе они записаны как самостоятельные файлы с обязательным названием Lxxxx, где xxxx это номер макроциклов, указанного файла под адресом L (G79 Lxxxx). В систему не должны быть записаны два или больше файлов с макроциклами, обозначенными одинаковым названием.

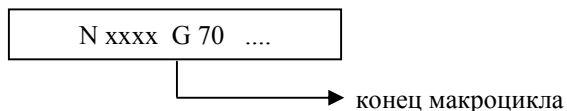
Частое использование макроциклов для акций, которые являются общими для всех партпрограмм, напр. установка в положение для смены инструмента и смена инструмента. Годится использовать также параметрическое программирование.

Вызов макроцикла можно выполнить из любого блока любой партпрограммы. Вызов имеет форму:



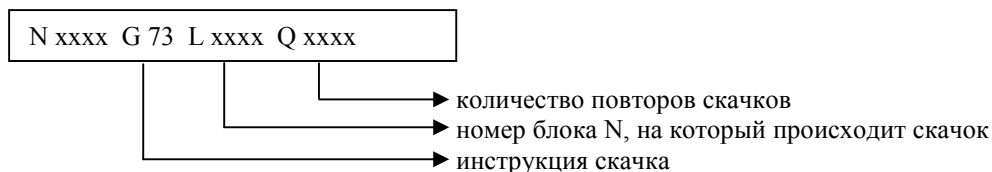
Требуемый макроцикл может быть также вызван из любого блока другого макроцикла (т. наз. вложением макроциклов). После окончания макроцикла останутся величины технологических и вспомогательных функций в таком состоянии, как их установил макроцикл.

В программном блоке партпрограммы (т.е. также подпрограммы) или макроцикла, с которого требуется вызвать другой макроцикл, допустимо программировать также остальные технологические функции и функции сдвига. Сам скачок на свободный макроцикл будет выполнен аналогично как в случае вызова подпрограмм в заключение блока. Каждый макроцикл должен быть закончен также как подпрограмма:



2.3.6 Скачок в партпрограмме, макроцикле

При помощи функции G73 можно запрограммировать скачок на другой программный блок. Блок, в котором запрограммирован скачок, имеет следующую форму:



Скачки разрешены только на запрограммированные блоки в рамках т. наз. программного комплекса, т.е. в рамках партпрограммы, или подпрограммы, или макроцикла в направлении назад и вперед (т.е. по направлению к началу или концу партпрограммы, подпрограммы или макроцикла). Нельзя тогда использовать напр. скачок из партпрограммы в подпрограмму.

Величина функции Q выражает количество повторов скачка. В случае если Q (n), потом отрезок программы между соответствующими блоками выполняется (n+1) раз.

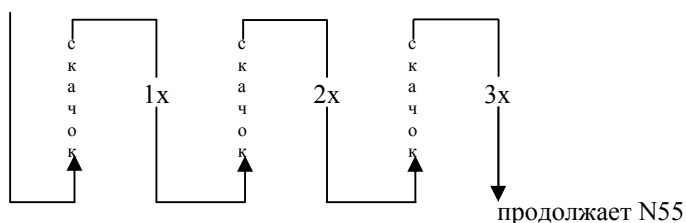
Пометка:

Вспомогательные функции L (номер вызванного PPRG, MRC и номер блока, на который будет выполнен скачок), Q (количество повторов вызова PPRG, MRC или количество скачков при SPRB) действительны только в блоке, в котором они запрограммированы.

Пример:

N 25
 N 26
 N 27

 N 54 G73 L26 Q3
 N 55



Часть партпрограммы между блоками N26 и N54 будет выполнена один раз и три раза повторяться, т.е. будет всего четыре раза выполняться.

В программном блоке, в котором запрограммирован скачок, допустимо программировать также остальные функции сдвига и технологические функции. Собственный скачок на определенный программный блок будет выполнен в заключение блока.



3. ПРОГРАММИРОВАНИЕ КООРДИНАТ

3.1 Система координат – обозначение координат

Система управления CNC8x6 может управлять макс. 6 одновременно управляемыми координатами (X, Y, Z, 4., 5., 6.). В одном блоке можно запрограммировать максимально 6 координат одновременно.

Координаты X, Y, Z создают основную систему координат (пространство). Оси этой пространственной прямоугольной системы являются параллельными главным направляющим плоскостям станка.

Ориентация осей, удовлетворяющая определению по нормам DIN 66217 и ISO/R 841, потом определена следующим способом: "Ось Z всегда параллельна оси рабочего шпинделя или с ней согласована. Положительное направление оси Z направлено от обрабатываемого полуфабриката к инструменту. Ось X находится в горизонтальной плоскости перпендикулярной к оси Z и параллельной поверхности укрепляющей плоскости стола. Она является главной осью движения в этой плоскости. На случай вращающегося инструмента с горизонтальной осью направлено положительное направление движения оси X направо при виде от шпинделя к укрепляющей плоскости стола. Ось Y потом дополняет координаты до нормальной (т.е. прямоугольной, правой) системы координат.

Направление координат принято в направлении относительного движения инструмента в отношении постоянного обрабатываемого полуфабриката. Движения инструментов обозначаются буквами без черточки (напр. X). Если в данной оси движется обрабатываемый полуфабрикат относительно постоянного инструмента, его движение находится в противоположном направлении и обозначаем его буквой с черточкой (напр. X').

С координатами, обозначенными 4. и 5., можно соединить обозначения адресов U, V, W, A, B или C. Соединение адресов с этими координатами выполняется установкой соответствующих машинных констант.

Координату 6. можно соединить с обозначением адресов A, B или C.

Во время соединения с адресами действителен принцип, что не должны быть две координаты обозначены одинаковым адресом.

Если не установлено требуемое соединение с адресами, стандартно соединяется 4. координата с адресом U, 5. координата с адресом V и 6. координата с адресом C. В следующих разделах будем эти координаты таким образом обозначать.

3.2 Свойства одновременно управляемых координат

Все координаты X, Y, Z, U, V управляются в связи с положением для рабочего сдвига и быстрого сдвига. Таким образом, для управляемых координат действительна во время их сдвига определенная

функциональная зависимость, которая гарантирует соблюдение движения по запрограммированной траектории.

Все движения (рабочие и быстрый сдвиг) выполняются плавным линейным достижением требуемой скорости в начале движения и линейным понижением требуемой скорости до минимальной скорости в конце движения. Это свойство системы называется разбег и замедление, оно гарантировано при всех видах интерполяции (кроме резки резьбы), и при override и СТОПе.

Крутизна разбега и замедления (размер ускорения и остановки) одинакова для рабочих сдвигов и быстрого сдвига. Размер крутизны разбега и замедления определяется во время наладки сервомеханизмов отдельных координат для положения и вводится в систему в качестве машинной константы.

В случае обработки непрерывной кривой линии разгон и замедление между отдельными блоками пропускается. Требуемая непрерывность траектории движения между отдельными блоками оценивается системой автоматически.

Координата С резервирована для ввода положения шпинделя для управления связи по положению.

3.3 Введение траектории

В случае прямолинейных координат векторная проекция кривой линии на систему координат (траектория движения координаты) указана в мм. Система позволяет разрабатывать самые мелкие приросты, составляющие 1 мкм. Для ротационных координат используется программирование в градусах или тысячных долях градуса.

Траекторию движения можно программировать выбором соответствующей G-функции из группы 9. или абсолютно, или с приростом (инкрементально).

Начало программы – это программистом выбранная точка, из которой программируется партпрограмма для обработки. Расстояние выбранного начала партпрограммы по сравнению с постоянной нулевой точкой станка (NBS) называется сдвиг нулевой точки.

Пример введения программируемого сегмента абсолютно и с приростом указан на рис. 1

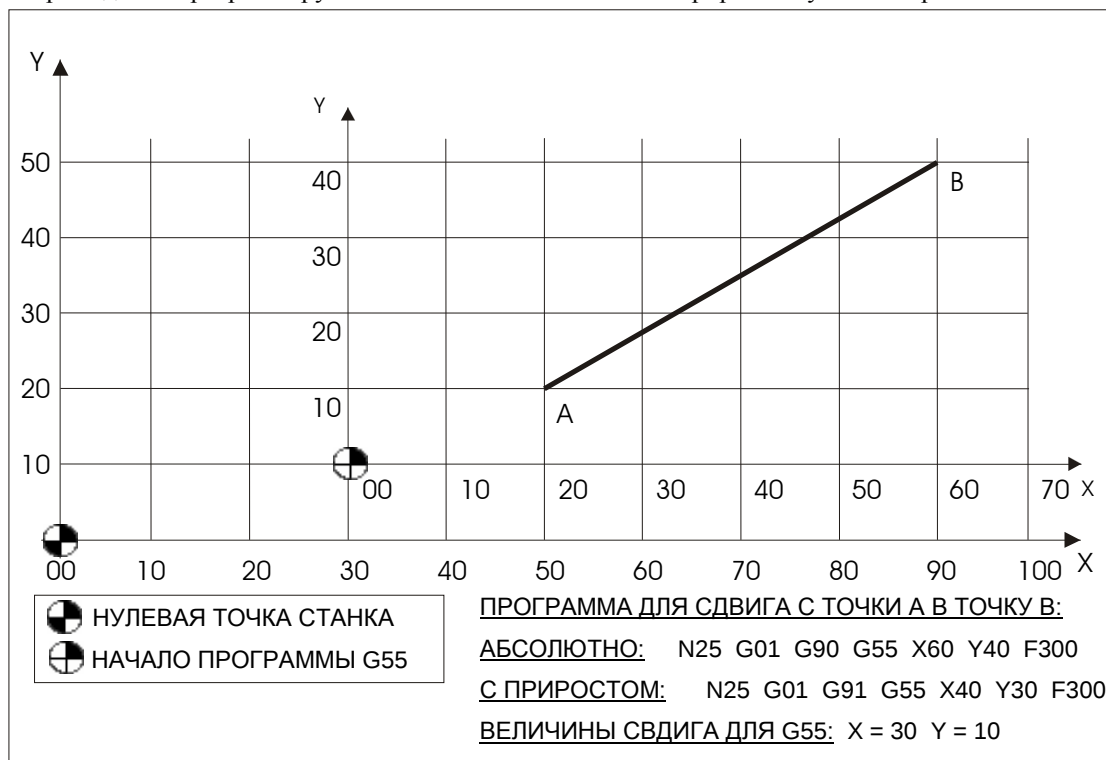


рис. 1

Величины координат можно программировать без десятичного знака или с десятичным знаком. Величина координат, запрограммированная без десятичного знака, выражает в случае линейной координаты (X, Y, Z, U, V, W, I, J, K) данные в мм, в случае ротационной координаты (A, B, C) данные в градусах (десятичный знак не должен указываться).

Пример:

X 100 = 100 мм = 100000 микронов
 Y 100.35 = 100.35 мм = 100350 микронов
 X 0.135 = 0.135 мм = 135 микронов
 Y .002 = 0.002 мм = 2 микрона

B 120 = 120°
 C 140.5 = 140.5°

3.4 Нулевые точки станка и программы

3.4.1 Определение нулевых точек станка (NBS)

Нулевая точка станка NBS – это постоянная точка станка и составляет начало основной связанной системы координат машины. Основная система координат с этой основной системой координат станка совпадет в случае установки исходного (референционного) положения в режиме РЕФЕРЕНЦИЯ. Расстояние референционной точки от нулевой точки станка вводится в качестве машинной константы. (Файл TAB0.REK, константы от 80 до 85)

Система координат программы с этой основной системой координат (станка и системы) совпадает при помощи функции G53 (или G54) при предположении, что величины сдвига для функции G53 (или G54) в таблице сдвига начала составляют 0 величину.

Пометка: Приоритетный сдвиг стандартно установлен на G53, изменение приоритетного блока можно выбрать приоритетным G54. Описание изменения приоритетного блока указано в самостоятельном разделе в приложении В - „Приоритетный блок“.

В случае если величины сдвига начала для функции G53 (G54) приобретают другие величины, чем ноль, потом система координат программы совпадает при помощи функции G53 (G54) со сдвинутой системой координат данной программы с точкой начала NBP0.

Установленные величины сдвига отдельных нулевых точек, указанных в таблице сдвига начала для функции G53 - G59, всегда относятся к нулевой точке станка NBS (без учета того, если для функции G53 (G54) установлен сдвиг, отличающийся от нуля).

Рекомендовано в таблице сдвига начала для приоритетной функции G53 (G54) запрограммировать нулевые величины сдвига координат. Эта функция преимущественно определена для трансформации системы координат программы в основную систему координат системы (станка и системы).

RB - Референционная точка машины является постоянной точкой станка, которая определена включением концевого выключателя и нулевым прохождением измерительной системы.

NBS – Нулевая точка станка, постоянная точка станка, которая находится на расстоянии на размер X_R по сравнению с референционной точкой станка RB. Величина X_R установлена как машинная константа в таблице машинных констант TAB0.REK и нельзя ее произвольно менять.

NBP0 - NBP6 – Нулевые точки программы, общие точки, в отношении которых выполняется программа, находящиеся на расстоянии на величину X_{NP} по сравнению с нулевой точкой станка NBS (величина сдвига X_{NPO} действительна для G53, X_{NP1} для G54, и т.п.).

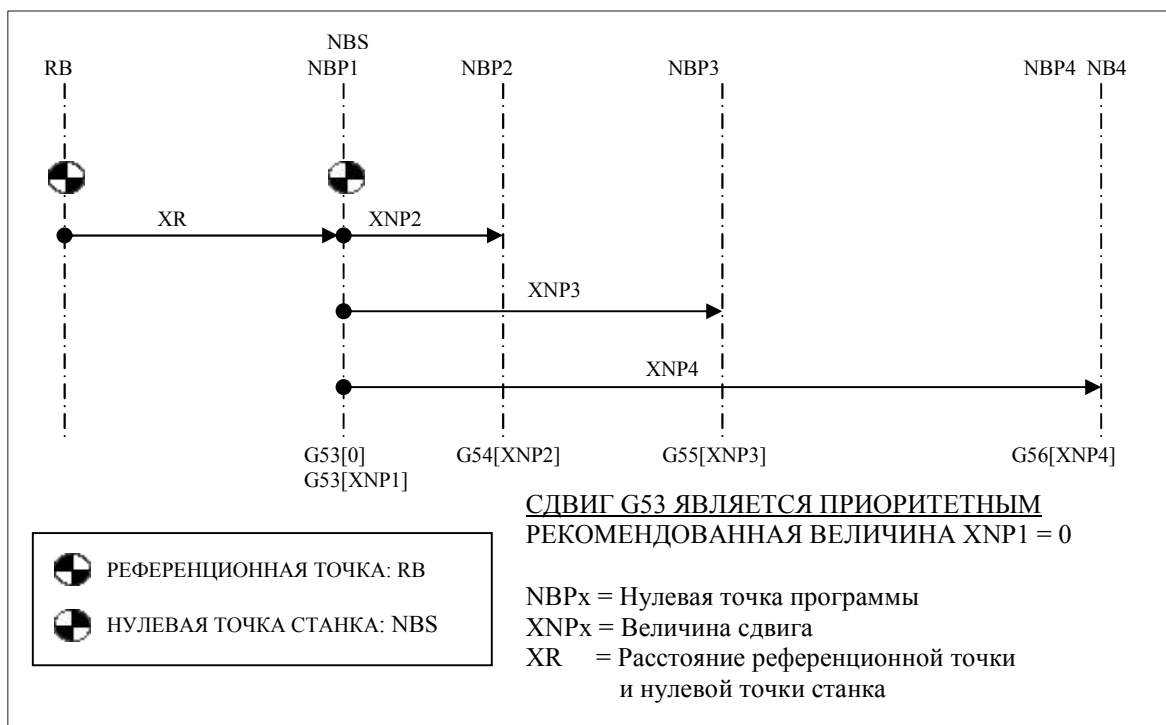


рис. 2

3.4.2 Референционная точка станка

Каждая ось системы координат обладает постоянной точкой станка, которая обозначена как референционная точка станка RB. Во время установки RB в режиме РЕФЕРЕНЦИЯ (установка исходного положения) в запоминающее устройство абсолютного положения координат в компьютер записывается величина X_R , установленная как машинная константа, таким образом, устанавливается совпадение основной системы координат с основной системой координат машины. (Величина X_R действительна для первой координаты (обычно X), Y_R для второй координаты (обычно Y, для токарных станков Z), и т.п.).

3.4.3 Нулевая точка программы

Система позволяет сдвинуть начало системы координат программы при помощи функции G53 - G59 в общую точку NBP0 - NBP6 в рабочем пространстве станка (рис.2).

3.4.4 Перемещение нулевых точек (группа G5)

Во время перемещения системы координат программы (нулевой точки) относятся все остальные абсолютные размеры к новой нулевой точке. В случае инкрементального программирования не влияет сдвинутая нулевая точка на точку окончания запрограммированной траектории. При абсолютном программировании запрограммированное окончательное положение траектории автоматически пересчитывается на новое окончательное положение, относящееся к соответствующей нулевой точке. Индикация абсолютного положения (мгновенное положение) относится всегда к некоторой точке станка или программы (референционной, нулевой точке станка или нулевой точке программы). В отношении, которой точки это выполняется, определено G функцией из пятой группы.

Величина сдвига вышеуказанной точки в отношении NBS (нулевой точки станка) установлена в следующей таблице сдвига начала:

FCE	ВЕЛИЧИНЫ СДВИГА В ОСЯХ						
	1.ось	2.ось	3.ось	4.ось	5. ось	6.ось	
G53	1 _{NP0}	2 _{NP0}	3 _{NP0}	4 _{NP0}	5 _{NP0}	6 _{NP0}	Постоянная приоритетная
G54	1 _{NP1}	2 _{NP1}	3 _{NP1}	4 _{NP1}	5 _{NP1}	6 _{NP1}	Постоянная функция (может быть приоритетной)
G55	1 _{NP2}	2 _{NP2}	3 _{NP2}	4 _{NP2}	5 _{NP2}	6 _{NP2}	Постоянная функция
G56	1 _{NP3}	2 _{NP3}	3 _{NP3}	4 _{NP3}	5 _{NP3}	6 _{NP3}	Постоянная функция
G57	1 _{NP4}	2 _{NP4}					Постоянная функция
G58	1 _{NP5}						Действительно только в одном блоке
G59	1 _{NP6}				5 _{NP6}	6 _{NP6}	Действительно только в одном блоке

При программировании сдвига нулевой точки в блоке при помощи функции G53 - G59 никакой сдвиг координат не осуществляется, если в этом блоке перемещение не определено по-другому. В программе использованные функции G53 - G59 должны иметь до своего использования в блоке определенные величины сдвига в отдельных координатах в таблице сдвига начала. Величины, определенные в таблице сдвига начала, останутся постоянно сохранены, пока они не переписаны в режиме редактирования новой величины или изменены интерактивным вводом (см. Инструкция по обслуживанию) или при помощи функций G92 или G93 из партпрограммы.

Функции G53 - G57 обладают постоянным действием, т.е. соответствующей G функцией выбрана система координат действительная для всех следующих блоков, пока она не переписана другой G функцией 5-ой группы. Функции G58 и G59 обладают действием только в блоке, в котором они запрограммированы, в следующем блоке система возвращается в сдвинутую систему координат, выбранную и запрограммированную ранее любой из функций G53 - G57.

3.4.5 Заполнение таблицы сдвига начала

Системную таблицу сдвига начала можно наполнить тремя способами:

- Из партпрограммы при помощи функций G92 или G93
- С пульта управления редактированием файла TAB0.POS
- С панели обслуживания интерактивного ввода.

Заполнение таблицы сдвига начала с панели обслуживающего персонала аналогично описано в Инструкции по обслуживанию системы управления CNC8x6. Не рекомендуется комбинировать заполнение сдвига из партпрограммы и с панели для обслуживающего персонала. Подходящим способом является выбор одного из способов и этот использовать.

3.4.6 Заполнение таблицы сдвига из партпрограммы функциями G92 и G93

Заполнение таблицы из партпрограммы на практике часто не используется. Если оно используется, не зависит это от содержания файла TAB0.POS, с которого после включения системы переписываются величины сдвига в системные таблицы сдвига. Сдвиг можно определить прямо в каждой партпрограмме.

Под определением сдвига нулевых точек понимаем соединение нумерических величин сдвига начала координат для отдельных G функций 5-ой группы. Величины сдвига начала уложены для отдельной G функции 5-ой группы в таблице. Таблицу сдвига начала можно переписывать и записывать в нее данные следующим способом:

- Запись величин таблиц сдвига начала из программного блока партпрограммы при помощи функции G92. Во время введения и выполнения программируемого блока с функцией G92 произойдет запись величин в таблицы сдвига начала для соответствующей функции G53 - G59, которая запрограммирована в этом блоке как величина соответствующих координат. Программный блок, в

который требуем сдвиг записывать, должен иметь напр. следующую форму:

Nxxxx G55 G92 X100.5 Y200. Z-300.650 U0 V10.

В таком образом введенный блок в таблицу G55 записывается сдвиг для оси X 100.5, для оси Y 200 и т. п. Вместо функции G55 может быть запрограммирована также другая функция из группы G[5]. В блоке произойдет установка величин в таблице сдвига для пяти координат, величина, записанная в таблице для шести координат, останется без изменения. Если в блоке запрограммирована функция G92, данные, записанные в отдельных координатах, обладают значением величин, которые будут записаны в таблицу. **В этом блоке не произойдет никакое движение!** В таблицу будут записаны только величины для программируемых координат. Если координаты не запрограммированы, сдвиг в таблице для этой координаты останется без изменений.

- b) Автоматическая установка величин таблицы сдвига начала из программируемого блока при помощи функции G93. Во время введения и выполнения программируемого блока с функцией G93 произойдет автоматическая установка величин сдвига начала для соответствующей функции G53 - G59, во время которой произойдет совпадение с нулевой точкой партпрограммы для данной функции G53 -G59 с определенной точкой системы координат, которая находится на расстоянии от действительного положения инструмента на расстоянии запрограммированных координат в данном блоке. Форма программируемого блока для этой операции совпадает с блоком, указанным в точке а) с той разницей, что функция G92 заменена функцией G93.

Операцию использования функции G93 согласно b) объясним на следующих двух основных примерах.

Величина запрограммированной координаты равна 0 (рис. 3).

В этом случае система координат программы совпадает для данной функции G53 - G59 со своим началом с точкой, в которой координата находится. Индикация абсолютного положения (мгновенное положение) будет в этом случае показывать величину 0.

Пример блока:

N 355 G55 G93 X0

Порядок определения новой величины сдвига ($X_{NP2} + a_p$):

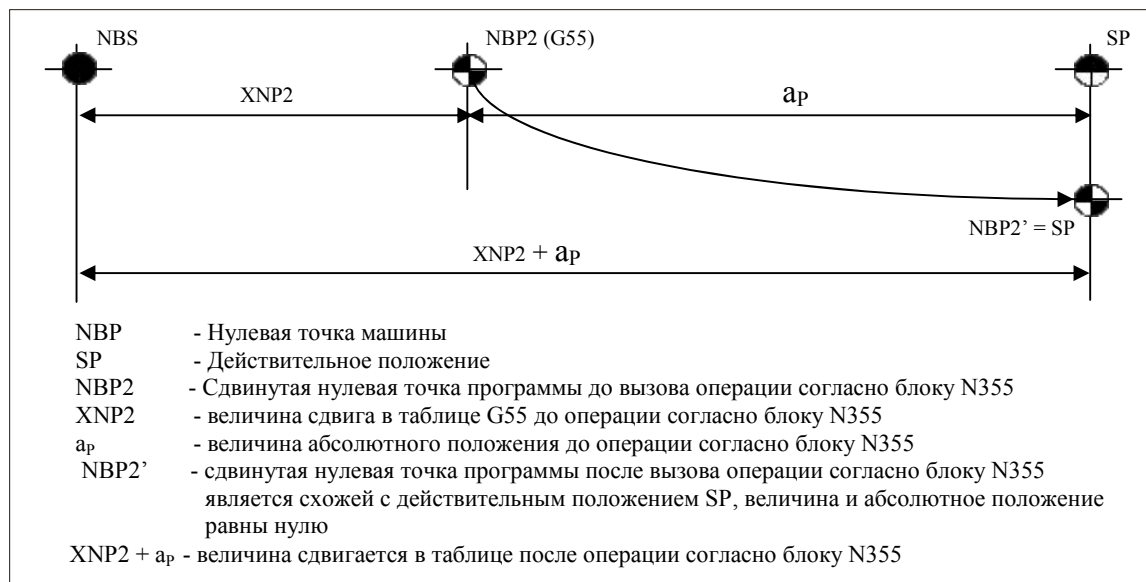


рис. 3

Величина программируемой координаты отличается от 0 (рис. 4)

В этом случае система координат программы совпадает для данной функции G54 и G59 своим началом с точкой, которая находится на расстоянии от точки, в котором координата находится от величины

запрограммированных координат в данном блоке. Индикация абсолютного положения (ОР-мгновенные положения) будет в этом случае равняться запрограммированным величинам координаты:

Пример блока:

N 356 G55 G93 X+10.000

Порядок определения новой величины сдвига ($X_{NP2'}$):

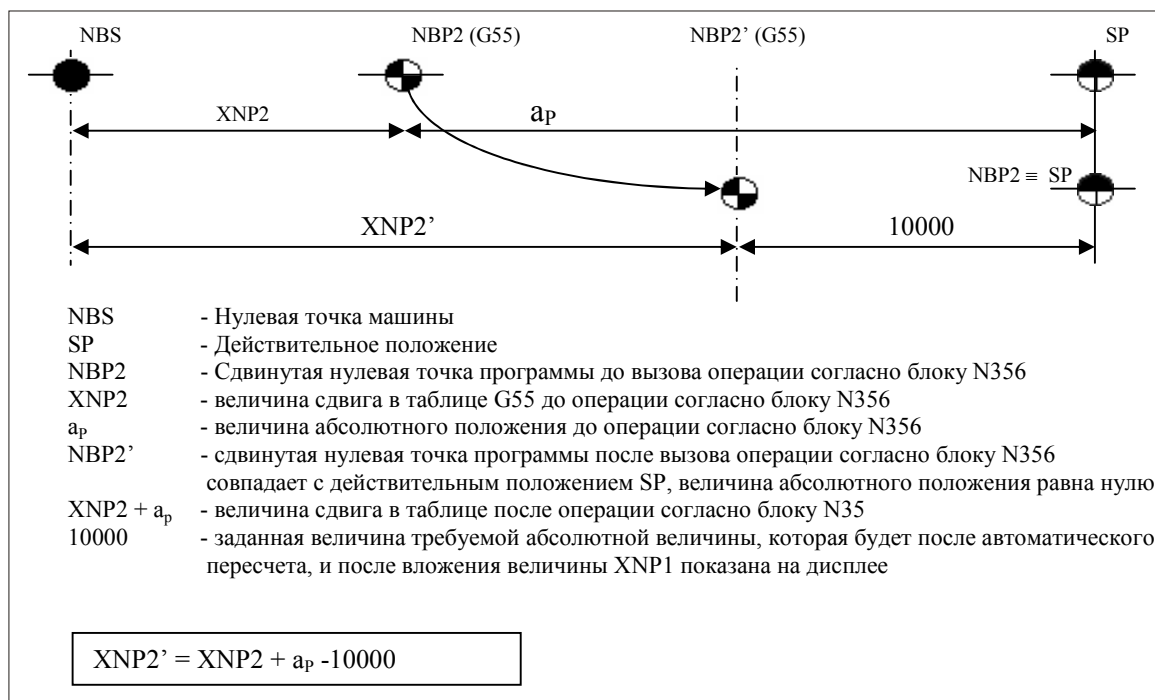


рис. 4

Предупреждение:

Величины, которые записываются при помощи функций G92 или G93 из партпрограммы в системные таблицы сдвига, не записываются в файл TAB0.POS!

3.4.7 Заполнение таблицы сдвига с редактора файла TAB0.POS

Этот способ использования является самым частым. Величины сдвига при помощи редактора урегулируются в файле TAB0.POS. Записью этого файла произойдет перепись величин с этого файла в системные таблицы сдвига и этот сдвиг действителен до следующего изменения, которое будет случайно в редакторе выполнено. Сдвиг, указанный в этом файле в системные таблицы переписывается также после включения системы, так что он сразу же действителен. В партпрограмме используются только функции G53 – G59 для выбора отдельного сдвига.

Подробное описание управления редактором для редактирования файла TAB0.POS указано в «Инструкции по обслуживанию».

3.4.8 Заполнение таблицы сдвига интерактивным вводом

Использование такое же, как было указано в предшествующем разделе, только способ записи в файл TAB0.POS другой – не при помощи редактора, а интерактивным управлением. Порядок описан в «Инструкции по обслуживанию».

3.4.9 Вызов сдвигов нулевых точек

Собственный сдвиг системы координат произойдет во время хода программы в момент выполнения блока с запрограммированной G-функцией 5-ой группы. В этом блоке и при G53 - G57, а также в остальных блоках (они постоянно действительны), начнут считаться программируемые координаты, так чтобы положение инструмента соответствовало позиции с учетом сдвига начала. Приоритетная G функция из группы 5 – это функция G53 (G54), которая устанавливается в случае старта новой программы в автоматическом режиме, или в случае достижения конца программы. Величины сдвига начала для функций G53 - G57, записанные в таблице сдвига, останутся сохранены постоянно, пока они не переписаны другой величиной (в начале и в конце программы не находятся под влиянием). Величину сдвига нулевой точки, действительной для любой вышеуказанной G функции, можно также определить в ходе выполнения некоторого автоматического режима во время установки программы и перехода в режиме ТАВ. При выполнении таблиц сдвига при помощи G92 и 93 выполняемый сдвиг применяется в следующем блоке.

Пример:

N10 G00 X10 Y20	“Если в первый блок не включены никакие сдвиги, действителен приоритетный сдвиг G53 (G54)
N20	“Действителен постоянно G53 (G54)
N30 G55 ...	“Включается сдвиг G55
N40	“Действителен сдвиг G55
N50 G58 ...	“Только в этом блоке включается и действителен сдвиг G58
N60	“В этом блоке возвращается действительный сдвиг G55 (хотя было запрограммировано)
N70 ...	“Действителен сдвиг G55

4

4. ВИДЫ ДВИЖЕНИЯ

4.1 Установка координат - функция G00

Под установкой координат понимается перемещение инструмента до конечной (программируемой) точки быстрым сдвигом. Конечное положение программируется в абсолютных или инкрементальных размерах (действительно, в общем, для всех видов движения). Это перемещение инструмента введено в блок функцией G00, которая одновременно является носителем информации для совершения движения быстрым сдвигом. Размер быстрого сдвига в системе устанавливается твердо как машинная константа и в блоке не программируется. В случае быстрого сдвига гарантирован плавный разъезд и остановка в начале и конце движения. В одном блоке можно программировать установку одной – шести координат.

4.2 Линейная интерполяция - функция G01

Линейная интерполяция выбирается функцией G01. В одном блоке можно запрограммировать интерполяцию между одной – шести координатами и запрограммированными координатами конечных точек в соответствующих осях. Таким образом, возможна также взаимная комбинация линейных и ротационных координат. Для линейной интерполяции необходима скорость сдвига под адресом F. Скорость F не должна быть указана в блоке с G01, а в любом предшествующем блоке.

Пример хода траектории в случае линейной интерполяции:

Абсолютное программирование:	Программирование с приростом:
N10 G01 G90 X 90.000 Y 50.000	N11 G01 G91 X 40.000 Y 30.000

Если исходной точкой интерполяции является точка А с координатами X=50, Y=20 в случае включенного сдвига G54, который сходится с NBS, и конечной точкой В с координатами X=90 Y=50, потом указанные блоки N10 и N11 выполняют одинаковую траекторию.

4.3 Круговая интерполяция - функция G02, G03

Круговая интерполяция выбирается или функцией G02 (движение по окружности по направлению часовых стрелок) или G03 (движение по окружности против направления часовых стрелок). Выражение направления круговой интерполяции (G02 или G03) в любой плоскости для системы координат (правой) определяется при виде на плоскость круговой траектории (рис.5).

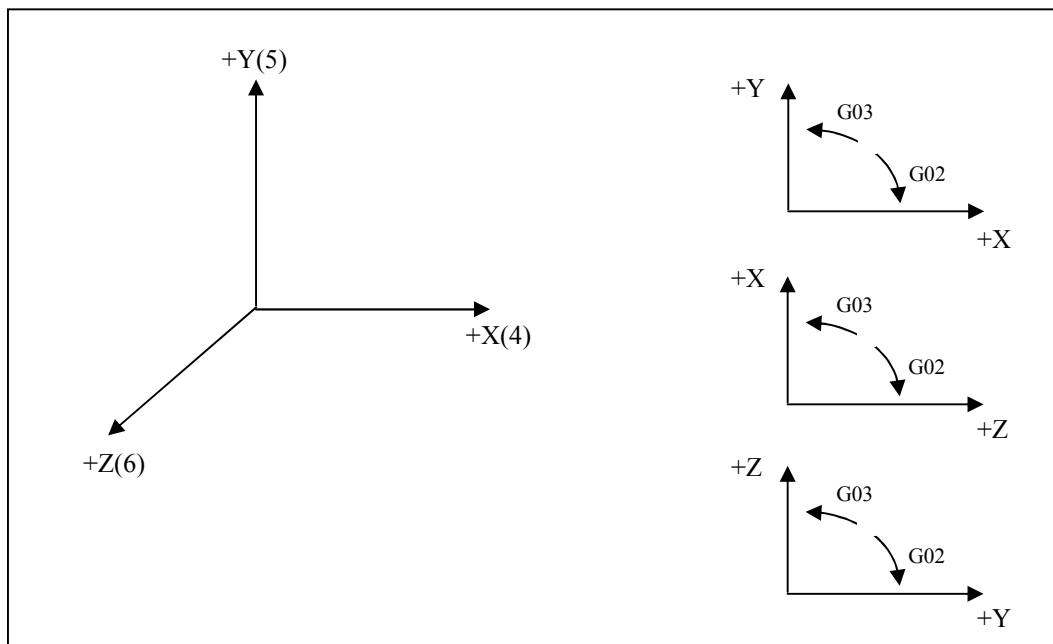


рис. 5

В скобках указаны координаты в системе, принимаемой как параллельная с первоначальными. Круговая интерполяция возможна только в одной плоскости. Плоскостью интерполяции можно выбрать плоскость, определенную двумя из шести координат X - Y, Y - Z, Z - X, 4.- 5. и т. п. Теоретически может круговая интерполяция программироваться также в случаях, когда любая из координат является ротационной, но программирование является явно сложным.

Круговая интерполяция вводится координатами **оконечной** точки окружности или круговой дуги. Координаты конечной точки должны быть введены **обе**, даже в случае, когда оконечная точка сходится с началом. Координаты оконечной точки можно вводить абсолютно или с приростом.

Координаты центра окружности программируются адресами I и J. В отличие от оконечной точки **координаты центра окружности должны программироваться только приростом с учетом начальной точки окружности** (к началу круговой интерполяции). Центр с учетом первой оси программируется адресом I, центр с учетом второй оси адресом J. В связи с этим необходимо отметить, что во время круговой интерполяции на плоскости Z - X, первой осью считается ось Z, и второй осью X (рис.5), т. е. расстояние центра окружности от начала в оси Z программируется адресом I и в оси X адресом J!

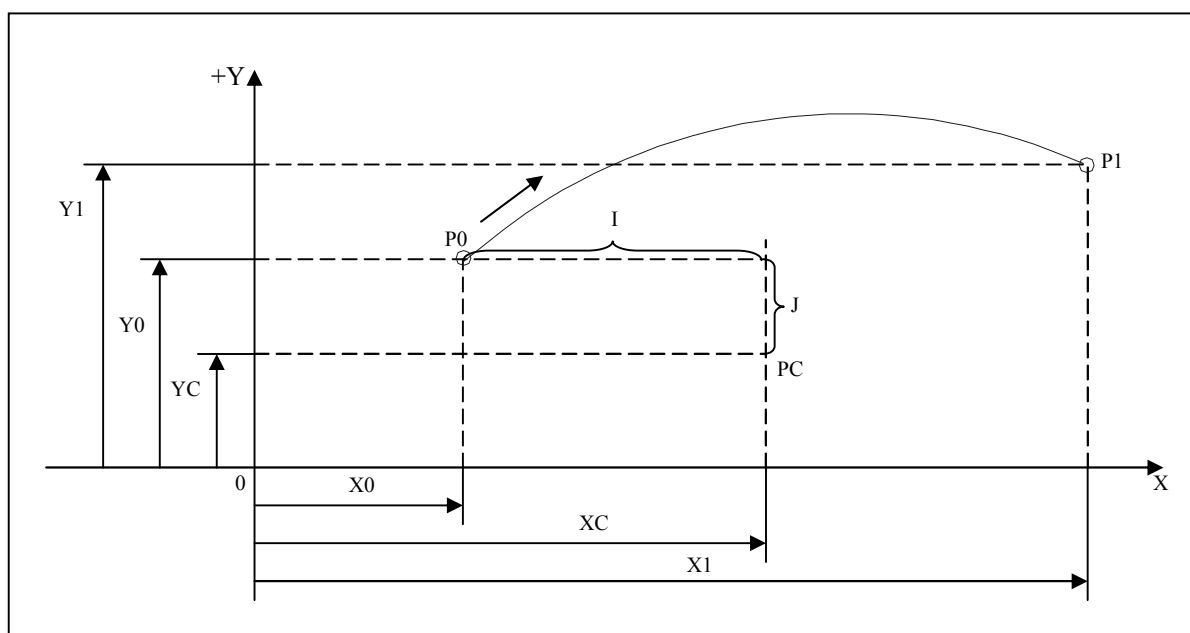


рис. 6

- P0 - начальная точка круговой интерполяции
- X0,Y0 - координаты начальной точки
- P1 - конечная точка круговой интерполяции
- X1,Y1 - координаты конечной точки
- PC - центр окружности (круговой дуги)
- XC,YC - координаты центра окружности

Круговая интерполяция на рис. 6 является G02 (по направлению движения часовых стрелок).
Координаты центра окружности для оси X находятся на расстоянии равном I от начала:

$$I = XC - X0$$

Координаты центра окружности для оси Y находятся на расстоянии равном J с начала:

$$J = YC - Y0$$

Знак перед адресом J будет отрицательным, потому что центр окружности с учетом начала находится в отрицательном направлении. Если адрес I или J являются нулевыми, они не должны быть в блоке запрограммированы.

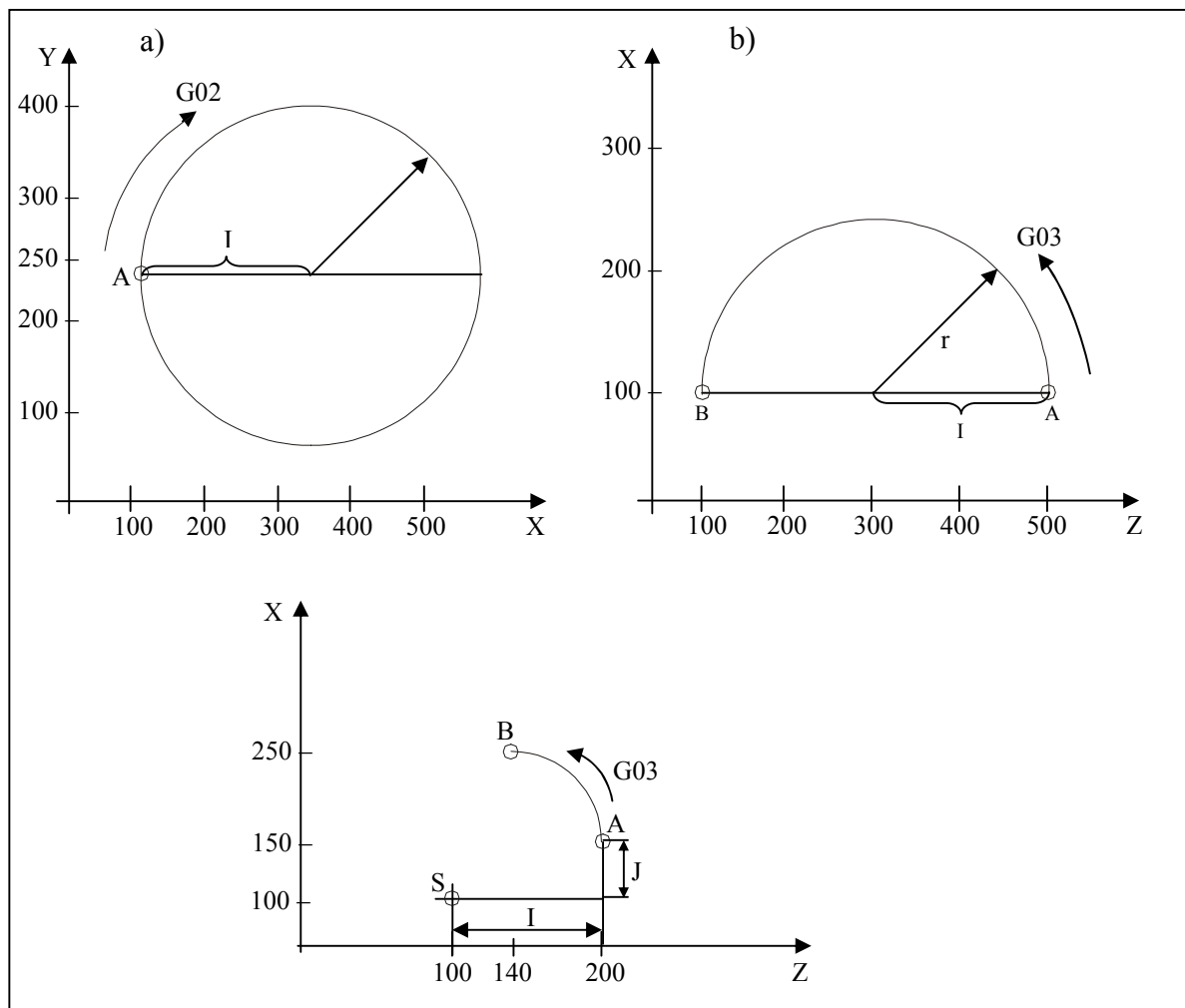


рис. 7

Пример программы для всей окружности в плоскости X - Y (рис. 7a):

Абсолютно:

N19 G02 G90 X200.0 Y250.0 I200.0 F120

С приростом:

N19 G02 G91 X0 Y0 I200.0 F120

Пример программы для полуокружности в плоскости Z - X (рис. 7b):

Абсолютно:

N20 G03 G90 X100.0 Z100.0 I-200.0 F100

С приростом:

N20 G03 G91 X0 Z-400.0 I-200.0 F100

Пример программы для отрезка окружности в плоскости X - Z (рис. 7c):

Абсолютно:

N21 G03 G90 X250.0 Z140.0 I-100.0 J-50.0 F100

С приростом:

N21 G03 G91 X100.0 Z-60.0 I-100.0 J-50.0 F100

Пометка:

Круговая интерполяция не ограничена одним квадрантом. Программируемая окружность может пересекать также большее количество квадрантов.

Если будет в случае круговой интерполяции запрограммирована координата конечной точки, которая

не находится на окружности, система сообщает об ошибке номер 7.56 - "КОНЕЧНАЯ ТОЧКА НЕ НАХОДИТСЯ НА ОКРУЖНОСТИ". Конечные точки, а также координаты центра, необходимо в партпрограмму вводить с точностью на один микрон! Точность ввода можно частично уменьшить установкой машинной константы номер 55 – допуск центра окружности.

5

5. РЕЗКА РЕЗЬБЫ НОЖОМ

5.1 Резка резьбы без выхода из траектории

Во время программирования резки резьбы ножом функцией G33 свяжет система движение в координате, для которой было введено поднимание, с движением шпинделя. Движение остальных интерполяционных координат осуществляется таким способом, чтобы конечное движение находилось в заданных координатах и было выполнено по заданной траектории.

Блок резьбы программируется под следующими адресами:

F	Поднимание резьбы, вводится в мм/оборот.
X,Y,Z,...	Координаты окончательной точки резьбы (можно программировать абсолютно или с приростом), при которой движение связано с вращением шпинделя
I	Угольный сдвиг начала связанного движения от НУЛЕВОГО ИМПУЛЬСА
M03,M04	Направление вращения шпинделя
S	Вращение шпинделя
G33	Резка резьбы

Максимальная величина подъема, вводится прямо адресом F, и составляет 99,99 мм/оборот.

Если I = 0, или если оно вообще запрограммировано, начнется взаимная связь передаваемых импульсов с датчика оборотов шпинделя, работает от передачи нулевого импульса датчиком. Если под нулевым адресом I запрограммирована величина, обозначает сдвиг угла начало, связанное с движением от нульпульса (нулевого импульса). Угольный сдвиг начала резьбы имеет практическое значение только во время программирования многозаходной резьбы.

Резка резьбы аналогична программированию сдвига в мм/оборот функцией G95. Единственной разницей является то, что во время программирования G33 ждет для начала движения НУЛЬПУЛЬСа, хотя в случае G95 для начала сдвига НУЛЬПУЛЬСА не ожидает.

Ожидание нулевого импульса необходимо для обнаружения точной повторной установки в резьбу в случае многократного прохождения.

Пример 1:

Пример указывает часть программы с расписанной резкой резьбы, т. е. **он не использован** как цикл для резки резьбы G84. Резка резьбы в оси Z начинается со значения Z3.0 мм и заканчивается на значении – 9.0 мм в выходной точке, подъем составляет 1,5 мм:

.....	
N325 G00 G95 T9 &1100 D9 S1000 M03 M42	"ГЛАВНОЕ ПРЕДЛОЖЕНИЕ
N330 G00 X26.618 Z3.	"УСТАНОВКА БЫСТРЫМ СДВИГОМ НАЧАЛА РЕЗЬБЫ
	"РЕЗКА РЕЗЬБЫ
N335 G33 Z-9.10. F1.5	"ВЫЕЗД БЫСТРЫМ СДВИГОМ ИЗ ВЫХОДНОЙ ТОЧКИ
N340 G00 X29.	

N345	Z3.	"ВОЗВРАТ G00 В ИСХОДНОЕ ПОЛОЖЕНИЕ
N350	X26.418	"ИСХОДНАЯ ТОЧКА ВТОРОГО ПРОХОЖДЕНИЯ
N355 G33	Z-9. I0. F1.5	"РЕЗКА РЕЗЬБЫ
N360	G00 X29.	
N365	Z3.	
N370	X26.177	"ИСХОДНАЯ ТОЧКА ТРЕТЬЕГО ПРОХОЖДЕНИЯ
N375 G33	Z-9. I0. F1.5	"РЕЗКА РЕЗЬБЫ
N380	G00 X29.	"и т. д.

Пример 2:

Блок резки резьбы в оси Z, начало сдвинуто на 180 градусов от нуля:

N20 M04 S500 Z200.0 F1.5 I180 G33

Пример 3:

Резка резьбы на конусе:

Во время резки резьбы на конусе будет запрограммировано еще вторая ось, в которой осуществляется сдвиг. Для токарных станков это ось X.

Указанный блок создаст по длине 100 мм конусную резьбу из начального диаметра 226 мм до конечного диаметра 246 мм.

N125 G00 G95 T9 & I100 D9 S500 M03 M42	"ГЛАВНОЕ ПРЕДЛОЖЕНИЕ
N130 G00 X226 Z3.	"УСТАНОВКА БЫСТРЫМ СДВИГОМ В НАЧАЛЕ РЕЗЬБЫ
N135 G33 X246 Z-100 I0. F1.5	"РЕЗКА РЕЗЬБЫ

5.2 Программирование резьбы с выходом из траектории

Программирование является принципиально одинаковым как в случае резьбы без выхода из траектории. В блоке дополняется только под **адресом J** длина выхода из траектории (всегда положительная величина, без учета направления сдвига). Программированное окончательное положение резьбы в этом случае включает в себя длину выхода из траектории. За блоком резки резьбы должен быть запрограммирован пустой блок. В следующем блоке должны быть запрограммированы все координаты. Если требуется изгиб, должны повторяться окончательные положения.

Пример 4:

Резка резьбы в оси Z с подъемом 6 мм и выходом из траектории 12 мм:

N10 M03 S50	"НАЧАЛО ВРАЩЕНИЯ ШПИНДЕЛЯ
N20 Z5. X64.	"ИСХОДНОЕ ПОЛОЖЕНИЕ
N30 Z-100 F6.0 J12 G33	"РЕЗКА РЕЗЬБЫ С ВЫХОДОМ ИЗ ТРАЕКТОРИИ 12 мм
N40	"ПУСТОЙ БЛОК
N50 X64. Z-100. G0	"ДОЛЖНЫ БЫТЬ ЗАПРОГРАММИРОВАННЫ ВСЕ ОСИ

5.2.1 Угол выхода из резьбы

Угол выхода из резьбы можно ввести двумя способами:

- A) машинной константой
- B) параметром

А) Введение машинной константой

В машинную константу № 8 (файл TAB0.REK) вводится угол выхода из резьбы (максимально 60 градусов), включая знак.

Отдельные декады обладают следующим значением (zn = знак):

Декады:	8	7	6	5	4	3	2	1
zn.	0	0	1	1	2	5	0	0

В 1 - 5 декады вводится угол выхода с точностью на одну тысячную. В указанном примере установлен угол 12, 500 градусов.

В 6 декаду вводится номер по порядку оси, которая принимает участие на выходе из резьбы. В случае токарных станков это ось X, т. е. вводится 1. (Ось Y составляет 2, ось Z составляет 3 и т. п.).

7 и 8 декада не имеют значение – должны быть указаны нули.

Знак определяет, если угол выхода положительный или отрицательный (напр. для внутренней резьбы может быть отрицательным).

Этот способ введения угла выхода из резьбы (т.е. при помощи машинной константы) на практике используется только в случае, если требуется всегда одинаковый угол выхода и, прежде всего одинаковое направление выхода из резьбы.

В) Введение параметром

Если нам необходимо менять угол выхода и, прежде всего направление выхода, что является на практике необходимым, если запрограммированы внутренняя и внешняя резьба, должен быть угол и направление выхода запрограммированы параметром, номер которого определяется в одинаковой машинной константе (№ 8)

Отдельные декады потом имеют следующее значение (zn. = знак):

Декады:	8	7	6	5	4	3	2	1
zn.	0	4	0	0	0	0	0	0

7 и 8 декады определяют номер параметра, в котором запрограммирован угол и направление выхода, остальные декады не имеют значение в этом случае (резка резьбы с выходом).

В примере в 7 и 8 декадах 04, т. е. угол и направление будут приняты из параметра R04.

Важная пометка:

В общем, можем выбрать любой параметр (**кроме R00 и R80**), если он с резкой резьбы, расписывается в партпрограмме самостоятельно. **Если, однако, используются твердые циклы** (файлы PEVNECY4.NCP, PEVNECY5.NCP или PEVNECY6.NCP), **должен быть использован только параметр R04**, как указано в примере. Рекомендуются тогда выбирать, если это возможно, этот параметр.

Параметр заполнен одинаковым способом как машинная константа, если в ней определен угол. Например, для угла выхода 20 градусов в направлении в минус, причем выход будет лежать в оси X, будет параметр наполнен следующим способом:

N10 R04= -00120.000

Внимание: Не забудьте указать в 6 декаде номер оси, которая «выходит» из резьбы. Для токарных станков это X, т. е. записывается 1.

Пример 5:

Одинаковый как пример 4, только направление выхода введено не машинной константой, а параметром R04. Он может быть установлен в любое время до собственной резки резьбы.

N10 M03 S50

N20 Z5. X64. R04=120.0

N30 Z-100 F6.0 J12 G33

N40

“НАЧАЛО ВРАЩЕНИЯ ШПИНДЕЛЯ

“ИСХОДНОЕ ПОЛОЖЕНИЕ и ВВЕДЕНИЕ УГЛА И НАПРАВЛЕНИЯ ВЫХОДА

“РЕЗКА РЕЗЬБЫ С ВЫХОДОМ 12 мм

“ПУСТОЙ БЛОК

5.3 Резка резьбы с въездом и выходом (выходом из траектории)

Пометка: Действительно от версии пульта 30.29

В некоторых специальных случаях годится, или даже необходимо, иметь возможность запрограммировать резку резьбы с въездом («выходом в начале резьбы»). На практике это можно использовать, если по технологическим или другим причинам необходимо создать резьбу с выходом «от конца». Иногда может возникнуть также требование к резьбе с ходом и выходом, например для создания спиральной смазывающей канавки и т. п.

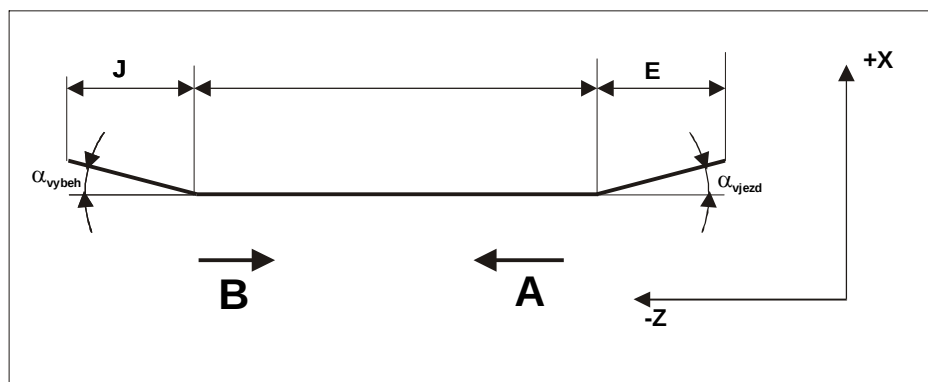
Длина „въезда“ вводится адресом E, что является въездом и что выходом с резьбы определено направлением движения. Если движемся в направлении A (см. рис.) въездом является величина с размерами на рис. под буквой E и выходом величина с размерами под буквой J. Эти адреса должны быть также в программе.

Если будет движение в направлении B (см. рис.), въездом может быть величина с размерами на рис. под буквой J, но в программе должен находиться адрес E (E=въезд), выходом может быть величина с размерами на рис. под буквой E, но в программе должен находиться адрес J (J=выход).

Пометка:

Адрес E необходимо всегда запрограммировать с десятичным знаком, т.е. въезд 10 мм необходимо запрограммировать E10. или E10.0, в случае адреса J это не является необходимым, может также с десятичным знаком программироваться J10, одинаковым способом как J10.0.

Угол въезда и выхода вводится в диапазоне 0 - 60 градусов. **Знак углов программируется одинаковым способом как въезд и выход** (определяющим направлением является выход из резьбы), причем система сама поменяет направление въезда в резьбу. Как в примере, указанном на рисунке, будут введены входной и выходной угол положительно.



Правила для ввода угла входа и выхода являются аналогичными как для резки резьбы с выходом.

А) Введение машинной константой

Используется в случае, если угол въезда и выхода являются одинаковыми, и использует только одно направление въезда и выхода.

Отдельные декады обладают следующим значением (zn = знак):

Декады:	8	7	6	5	4	3	2	1
Zn	0	0	1	1	2	5	0	0

В 1 – 5 декады вводится угол выхода и въезда с точностью на одну тысячную. В указанном примере угол установлен на 12, 500 градусов.

В 6 декаду вводится порядковый номер оси, которая принимает участие на выходе и въезде в резьбу. В случае токарных станков это ось X, т.е. вводится 1. (Остальные случаи не имеют практического значения).

7 и 8 декада не обладают значением – должны быть указаны нули.

Знак определяет, если угол выхода и въезда является положительным или отрицательным (напр. в случае внутренней резьбы он может быть отрицательный).

В) Введение параметром

Если необходимо поменять угол выхода и въезда, а также направление выхода и въезда, что на практике необходимо, если программируется внутренняя и внешняя резьба, должны быть угол и направление выхода и угол и направление въезда запрограммированы параметрами, номера которых определены в одинаковой машинной константе (№ 8).

Отдельные декады потом обладают следующим значением (zn. = знак):

Декады:	8	7	6	5	4	3	2	1
zn.	1	5	1	6	0	0	0	0

7 и 8 декады определяют номер параметра, в котором запрограммирован угол и направление выхода, 5 и 6 декады определяют параметры, в которых запрограммирован угол и направление въезда.

Остальные декады не имеют в этом случае значение.

В примере в 7 и 8 декады введено 15, т.е. угол и направление выхода будут приняты из параметра R15. В 5 и 6 декады 16, т.е. угол и направление въезда будут приняты из параметра R16.

Важная пометка:

В общем можно выбирать любые параметры, **кроме R00 и R80.**

Параметры заполнены одинаковым способом как машинная константа, если в ней определен угол. Например, для угла выхода 20 градусов в направлении минус, причем выход будет находиться в оси X, и для угла въезда 18 градусов в направлении минус, причем въезд будет находиться в оси X, будут параметры заполнены следующим способом:

N10 R15= -00120.000 R16= -00118.000

Внимание: не должны забыть указать в 6 декаде номер оси, которая «выходит» из резьбы. Оба номера для обоих параметров должны быть одинаковыми. В случае токарных станков это ось X, т.е. будет записана 1. Знаки будут указаны оба одинаковые (решающим является выход), хотя на самом деле поедет въезд в противоположном направлении.

Угол для въезда и выхода программируется в пределах от 0 до 60 градусов. Система сама определяет направление угла.

Пример части программ резки резьбы с выходом «до конца», т.е. использован въезд в резьбу, выход не использован. Для введения въезда использован параметр R16, угол въезда составляет 18 градусов и является положительным (знак определяет угол выхода, хотя в этом случае он не использован!):

N10 M04 S50	“НАЧАЛО ВРАЩЕНИЯ ШПИНДЕЛЯ
N20 X71.798 Z-100. R16=118.0	“ИСХОДНОЕ ПОЛОЖЕНИЕ, ВВОД УГЛА И НАПРАВЛЕНИЯ ВЪЕЗДА
N30 Z3 F6.0 E12 G33	“РЕЗКА РЕЗЬБЫ с ВЪЕЗДОМ 12 мм
N40	“ПУСТОЙ БЛОК
N50 X64. Z3 G0	“ДОЛЖНЫ БЫТЬ ЗАПРОГРАММИРОВАНЫ ВСЕ ОСИ

Внимание:

В отличие от резки резьбы с выходом, где перед резкой резьбы устанавливается требуемый диаметр и в котором будет выполнена резьба, для резки резьбы с въездом должна быть величина установлена на такой размер в оси X, чтобы после въезда достичь требуемого диаметра! Величина будет вычислена по углу и длине въезда. В указанном примере должна быть резьба выполнена с диаметром 64 мм.

5.4 Резка резьбы с большим шагом

Если необходимо создать резьбу с шагом больше, чем 99.99 мм, будет использован параметрический ввод шага по адресу F. Например, для шага 250 мм будет введение партпрограмм следующим:

```
.....  
N50 R10=250.0           "введение величины 250.0 в параметр R10  
N60 M03 S100            "начало вращения шпинделя  
N70 Z200.0 FR10 I0 G33   "шаг F вводится параметрически в параметр R10  
.....
```

Пометка: Действительно для всех типов резьбы (без выхода, с выходом и с въездом).

6

6. ВВЕДЕНИЕ СДВИГА

Рабочий сдвиг для обрабатываемой кривой линии программируется в адресе F. Сдвиг вводится способом согласно G-функции группы G6, т.е. эта функция определяет размер адреса F.

Во время программирования функций G94 и G96 отражается величина в адресе F:

а) если он запрограммирован без десятичного знака, то сдвиг в мм/мин.

Пример:

N1 G94 F 1000 обозначает сдвиг 1000 мм/мин.

б) если он запрограммирован с десятичным знаком, то сдвиг в м/мин.

Пример:

N2 G94 F.5 обозначает сдвиг 0.5 м/мин. т.е. 500 мм/мин.

N3 G94 F1.2 обозначает сдвиг 1.2 м/мин. т.е. 1200 мм/мин.

Система допускает макс. ввод размера сдвига 90000 мм/мин. (т.е. 90 м/мин.).

Во время программирования функций G95 и G97 отражается величина в адресе F:

а) если он запрограммирован без десятичного знака, то сдвиг в мкм/об.

Пример:

N4 G95 F 1000 обозначает сдвиг 1000 мкм/об.

б) если он запрограммирован с десятичным знаком, то сдвиг в мм/об.

Пример:

N5 G95 F 1.0 обозначает сдвиг 1 мм/об.

N6 F0.2 обозначает сдвиг 0.2 мм/об.

Система допускает максимальное введение размера сдвига на один оборот 99,999 мм/об. (F99999 или F99.999). Для сдвига на оборот необходимо снимать импульсы от шпинделя и поэтому станок с системой позволяют также резку резьбы. Сдвиг на оборот можем применить также в ручных режимах.

Возможная максимальная программируемая скорость для комплекта системы с данным станком определена свойствами станка. Во время программирования размера сдвига на один оборот зависит максимальная величина программируемого сдвига [мм/об.] от макс. допустимой скорости сдвига $V_{\max}$ и заданных оборотов шпинделя S.

Действительно отношение:

$$V = S \cdot \beta$$

где V - скорость сдвига в мм/об.

S - обороты шпинделя в об./мин.

β - сдвиг в мм/об. (запрограммирован в адресе F при G95)

Максимальный программируемый сдвиг макс. для максимальной скорости сдвига $V_{\max}$ и введенные обороты шпинделя S определены отношением:

$$\beta_{\max} = V_{\max} / S$$

и аналогично максимальные программируемые обороты $S_{\max}$ для данной максимальной скорости сдвига $V_{\max}$ и заданного сдвига определен отношением:

$$S_{\max} = V_{\max} / \beta_{\max}$$

Пример:

$V_{\max} = 2000$ мм/мин.

$S = 1000$ об./мин.

$\beta_{\max} = ?$

$$\beta_{\max} = V_{\max} / S = 2000 / 1000 = 2 \text{ мм/об.}$$

Для ротационных координат, когда скорость вводится в тысячных импульса ротационного датчика в минуту (градусах/мин.), рассчитаем скорость в миллиметрах/мин. на радиус R [мм] от оси вращения в случае, если координаты движутся сами по формуле:

$$F_{sk} = F \cdot 2\pi R / \text{количество тысяч импульсов на оборот.}$$

Во время программирования функции M36 (из группы M4) действительный сдвиг равен запрограммированному. Во время программирования функции M37 действительный сдвиг уменьшен в отношении 1: 100 по сравнению с запрограммированным.

6.1 Постоянная скорость резки (KRR) G96 и G97

В машинной константе (№ 67) определяется, которая координата для постоянной скорости резки является управляющей. В блоках с KRR скоростью должна находиться нулевая точка программы для управляющей координаты в оси вращения. Из величины адреса S и мгновенного радиуса рассчитываются постоянно мгновенные обороты, которые во время движения меняются. Так как во время прохождения блока не может меняться степень перевода, необходимо его выбору уделить большое внимание, или использовать несколько блоков с различной степенью перевода. При G97 по обратной связи прямо пропорционально меняется также сдвиг. Если скорость резки постоянна и в следующем блоке она не запрограммировано S , будут подставлены в последний раз подставленные обороты в адрес S . Во время перехода из оборотного на минутный сдвиг этот адрес F не выполняется. Ручные режимы постоянной скорости резки не выполняются.

Система использует для KRR две функции:

G96 – Постоянная скорость резки со сдвигом мм/мин.

G97 – Постоянная скорость резки со сдвигом мм/об.

Скорость резки вводится функцией S , которая в этом случае обладает значением не оборотов, а скорости резки в десятичных частях метра/мин. Например скорость резки 50 м/мин. вводится величиной S500.

Пример: G97 S500

Так как для большинства машин, прежде всего токарных станков и каруселей, принято **вводить постоянную скорость резки со сдвигом при вращении функцией G96 и скорость резки функцией S в м/мин.**, можно в машинных константах установить поменять эти функции, а также пересчитать S . Эти пересчеты выполнит система автоматически и программист записывает только скорость резки в м/мин.

Рекомендованная установка машинной константы для ввода KRR со сдвигом на оборот G96 и S в м/мин.:

Файл TAB0.REK – машинная константа номер 329, третья декада = 3

Пример:

N10 G0 X300 Z100 M44 M3 S100 „быстрый сдвиг на диаметр 300., обороты 100 об./мин.

N15 Z-1

N20 G96 G1 F200 S90 X50 „включит KRR 90 м/мин., спуск на диаметр 50, сдвиг F = 0.2 мм/об.

N25 G0 Z100

Во время программирования быстрого сдвига меняются обороты также по диаметру, если функция G96 не отозвана программированием G94.

Пометка 1

Если не будет установлена машинная константа № 329 по рекомендации, указанной выше, должен быть блок N20 запрограммирован следующим способом:

N20 G97 G1 F200 S900 X50 „включит KRR 90 м/мин., спуск на диаметр 50, сдвиг F = 0.2 мм/об.

Пометка 2

Для старшей версии была для смены G96/G97 и запрограммирования S в м/мин. использована инверсия через программу KONV836.EXE, вызов которой был установлен в CNC836.KNF, константа \$50.

7

7. КОРРЕКЦИЯ ИНСТРУМЕНТА

Пометка: Действительно от версии системы с 20.10.1997 г. и позднее.

Вложение колец в случае коррекции радиуса возможно от версии пульта 30.27 (10.11.2000 г.).

Коррекция инструмента позволяет создать общую партпрограмму, которую можно использовать для различных диаметров и длины инструментов (фрез, ножей токарных станков и т.п.). Во время смены инструмента с другими размерами урегулируется только соответствующая коррекция и партпрограмма останется без изменения.

Коррекцию имеем двух видов:

Коррекция радиуса инструмента

- определена G-функцией из 3. группы (G41, G42 и G40)

Коррекция длины

- определена адресом &

Этот раздел описывает коррекции радиуса.

7.1 Файл TAB0.KOR и таблица коррекций в запоминающем устройстве

Пометка: Название файла, или номер за TAB не обязательны. В общем можно использовать большее количество таблиц. В этой инструкции будем использовать название TAB0.KOR.

Размер коррекций (диаметра и длины) уложен в файле TAB0.KOR. Система во время своей работы функционирует с копией этого файла, уложенной в таблице коррекций во **внутреннем запоминающем устройстве станка. В эту таблицу в запоминающем устройстве в файл TAB0.KOR переписывается автоматически всегда после включения системы и далее после каждого редактирования файла TAB0.KOR, если будет после окончания редактирования выполнена его запись** (см. „Инструкцию по обслуживанию CNC836“).

Системная таблица коррекций во внутреннем запоминающем устройстве имеет следующую структуру:

Номер коррекции	I. данное	II. данное	III. данное	IV. данное	V. данное
1	Коррекция радиуса	Длина для первой оси	Длина для второй оси	Длина для третьей оси	Длина для четвертой оси
2	Коррекция радиуса	длина X	длина Y	длина Z	Длина 4.
и т.п. до 99					

Таблица коррекций имеет всего 99 статей, или система позволяет использовать 99 различных коррекций радиусов и длины инструмента. Каждая статья таблицы содержит всего 5 данных. Первые данные содержат величину коррекции радиуса. Следующие четыре данные – это коррекция длины первой, второй, третьей и четвертой осей. **Коррекцию длины нельзя использовать для случайной пятой и шестой осей.**

Коррекция, записанная в файле TAB0.KOR обладает одинаковой структурой, находится в файле TAB0.KOR в ниже описанной форме. (Пометка: Файл начинается с ключевого слова \$KOR, до него может быть записан любой комментарий.)

Строка начинается с двухместного номера коррекции (указан случайный не значительный ноль), за которым следует двоеточие. Величина коррекции радиуса указана знаком R =, величины коррекции длины указаны в названиях координат или порядковым номером координат.

Пометка:

Названия координат используются, если по порядку первая ось обозначена X, вторая ось обозначена Y, третья ось обозначена Z и четвертая ось обозначена U. Порядковый номер и название координат определены в машинных константах № 0 - 5 в файле TAB0.REK. Для токарных станков обычно первая ось - X и вторая ось - Z. В этом случае не должен использоваться файл для коррекции оси Z знак Z, а номер 2 (вторая координата). Для оси X можно использовать знак X (или номер 1).

В следующем примере файла TAB0.KOR указано несколько возможных способов записи. Рекомендованный способ записи в файл TAB0.KOR для трехосевой фрезы (X,Y,Z) указан для коррекции номер 1. Рекомендованный способ для токарных станков (X, Z), указан для коррекции номер 2. Для четырехосевой машины X,Y,Z,4 рекомендованный способ указан для коррекции номер 3.

Для коррекции № 25 не указана коррекция радиуса, потому что в таблице для этой коррекции радиусов будет находиться ноль. Для коррекции номер 95 указана запись при помощи порядковых номеров координат.

Если соответствующая коррекция не указана, ее величина будет записана в таблице (в запоминающем устройстве памяти системы) как нули. В указанном примере будут нулевые радиусы и коррекции длины во всех здесь неуказанных статьях, т.е. кроме коррекции № 1, 2, 3, 25 и 95, все остальные. Нулевыми будут также коррекции, неуказанные для отдельных осей.

\$KOR

```
01:    R=10.0  X=20.0  Y=30.0  Z=40.0
02:    R=0.8   X=0.0   2=0.0
03:    R=0.8   X=0.0   Y=120.0  Z=0.0   4=0.0
25:           X=85.0   Y=45.5   Z=0.0
95:    R=0.0   1=12.55  2=0.0   3=0.0   4=0.0
```

\$KOR

```
01:    R=0.0   X=0.0   Y=0.0   Z=0.0   4=0.0
```

Образец файла TAB0.KOR для токарных станков содержит 99 статей с нулевыми коррекциями в форме:

\$KOR

```
01:    R=0.0   X=0.0   2=0.0
```

Пометка 1:

Если коррекция радиуса используется для токарного станка, может быть в каждой статье еще указан тип используемого инструмента в следующей форме:

\$KOR

```
01:    R=0.0   X=0.0   2=0.0   P=3
```

Номер за P = должен находиться в пределах от 1 до 9. Другие величины не разрешены. Описание

использования типа инструмента для токарных станков указано ниже.

Пометка 2:

Файлы можно из-за наглядности сократить, если не использовать все 99 коррекции. Название файла TAB0.KOR не обязательно, для токарных станков он может быть в системе указан под названием TAB2.KOR. Важное то, чтобы одинаковое название не было также указано в конфигурационном файле CNC836.KNF в параметре 21. Хотя система позволяет использовать, в общем, несколько различных файлов с коррекциями, рекомендовано из-за определенности использовать (и размещать в запоминающее устройство) только один.

Способы заполнения таблицы коррекций

Во время записи (заполнения) таблицы коррекций необходимо отличить запись в файл TAB0.KOR и запись в копии таблиц во внутреннее запоминающее устройство системы.

Таблицу коррекций во внутреннем запоминающем устройстве системы можно заполнить следующими способами:

- а) Редактированием файла TAB0.KOR и **последующей записью** – описание редактора см. Инструкцию по обслуживанию. Обработанные величины коррекций останутся в файле постоянно записанными. После каждого включения системы переписуются во внутреннее запоминающее устройство системы.
- б) Интерактивным вводом в ручном режиме нажатием кнопки R, т.е. D, или & - описание см. Инструкцию по обслуживанию. Коррекции, введенные интерактивным способом, записываются как в файл TAB0.KOR, так и во внутреннее запоминающее устройство системы. Каждым включением системы они переписываются во внутреннее запоминающее устройство системы.
- в) Заполнения таблицы коррекций из партпрограммы программированием функции G92 – описание см. Инструкцию по программированию. **ВНИМАНИЕ:** Этим способом заполняется только таблица во внутреннем запоминающем устройстве системы. Величины в файле TAB0.KOR останутся без изменений. Это означает, что коррекции, записанные в системные таблицы в партпрограмме при помощи функции G92, действительны только до момента, пока не будет редактированием записан файл TAB0.KOR или отключена система. После включения действительны величины, указанные в файле TAB0.KOR. Так как запись в таблицы коррекций осуществляется прямо в партпрограмме, всегда гарантировано, что для данной партпрограммы будут всегда действительны величины коррекций, которые здесь будут записаны. Случайное изменение величин коррекций, однако, должно в этом случае осуществляться в партпрограмме, ни в коем случае в файле TAB0.KOR!

Для заполнения таблицы коррекций при помощи функции G92 используются параметры, величины которых приводятся в данную статью таблицы коррекций следующим способом:

величина параметра R0 определена для заполнения коррекций радиуса

величина параметра R1 определена для заполнения коррекций длины в первой оси (обычно X)

величина параметра R2 определена для заполнения коррекций длины во второй оси (обычно Y или Z для токарных станков)

величина параметра R3 определена для заполнения коррекций длины в третьей оси (обычно Z)

величина параметра R4 определена для заполнения коррекций длины в четвертой оси

Функцией G92 определяется, если соответствующий параметр в блоке запрограммирован, его величина будет записана в соответствующие данные той статьи таблицы коррекций, которая указана в адресе D.

Пример:

Хотим заполнить статью 12 таблицы коррекций следующими величинами:

коррекция радиуса 10.0 мм, коррекция длины в оси Y составляет 25,5 мм и коррекция длины в оси Z составляет - 5.0 мм. Блок партпрограммы будет иметь следующую форму:

N10 G92 D12 R0=10.0 R2=25.5 R3=-5.0

Если любой параметр в блоке не указан, размер соответствующей величины в определенной статье таблицы коррекций не меняется. В нашем случае величина коррекции длины в оси X и четвертой оси в статье 12 таблицы коррекций не меняются, т.е. останутся здесь первоначальные величины.

7.2 Коррекция радиуса с эквидистантой

Если установлена машинная константа номер 95, 8 декада на 1, можно использовать новый способ решения коррекции радиуса. Можно (за исключениями, указанными ниже) запрограммировать контуры обрабатываемого продукта или размеры чертежа. Траектория, подверженная коррекции, (траектория середины инструмента для фрез, траектория центра радиуса лезвия для токарных станков) находится на эквидистанте. В случае разрыва двух по себе последующих блоков доезжают до центра инструмента на пересечение эквидистант или вкладывается дуга, которая из этих возможностей использует, зависящие от установки машинные константы (и версии – версии ниже, чем 30.27 не умеют вкладывать в случае разрывов дугу).

Общий перечень установки машинных констант, касающихся коррекции:

95 – 8 декада	0	Старший способ коррекции радиуса, описанный в разделе 7А (уже не рекомендуется использовать – только из-за совместимости со старшими версиями)
95 – 8 декада	1	Рекомендованная установка. Коррекция длины с концевыми точками на пересечении эквидистант или с вкладыванием дуг
339 – 8 декада	0	Коррекция радиуса с окончательными точками на пересечении эквидистант. Необходимо использовать как прямоугольные машины (машины с переключаемыми осями)
339 – 8 декада	1	Коррекция радиуса с вкладыванием дуги в случае разрыва, если угол больше чем 180 градусов плюс допуск, введенный в 1 – 6 декады этой константы (можно использовать от версии 30.27)
339 – 7 декада	0	Отключение коррекции функцией G40 только в блоке, в котором запрограммировано движение хотя бы одной из осей плоскости коррекции (рекомендованный способ)
339 – 7 декада	1	Разрешение программировать функцию G40 и в неподвижных блоках (нерекомендованный способ – можно использовать от версии 30.27) – необходимо учесть то, что коррекция отключается в любом из последующих блоков, в котором будет запрограммировано движение в плоскости коррекции. С этого времени не должна плоскость коррекции меняться.
339 – 1 – 6 дек.		Вводится угол для вкладывания дуги. Дуга вкладывается, если будет угол между запрограммированными траекториями больше, чем здесь установленный.
39 - 1 – 8 дек.		Угол допуска для плавной стыковки – связанная машинная константа. Ее величина (угол) установлена для принятия решений системой о плавном передвижении. Величина в машинной константе 339 (угол) должна быть меньше или максимально равна углу, введенному в этой константе.

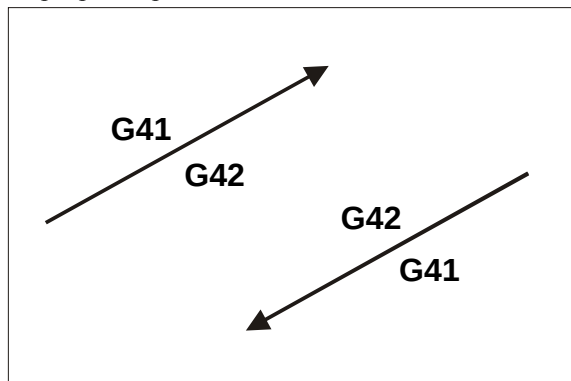
Значение этих констант видно также по следующему тексту и рисункам.

Коррекция радиуса эффективна в конце блока, в котором была вызвана функция G41 или G42, т.е. начнет действовать в следующем блоке. **Вызванная (включение функций G41 или G42) и отключенная (изъятие функций G40) коррекция радиуса может быть только в блоке с линейной интерполяцией.** Нельзя включать и отключать коррекцию радиуса на окружности.

Для программирования коррекции радиуса G41, G42 действительны следующие правила:

Коррекция G41/G42 являются постоянной функцией и действительны до отключения функции G40. Коррекция действительна для выбранной плоскости коррекции (G17,G18,G19). Если плоскость коррекции явно не выбрана, действительна приоритетная плоскость коррекции G17 (плоскость 1 и 2 оси – т.е. обычно XY для фрез и XZ для токарных станков). Если коррекции радиуса используем только в этих осях, не должна быть плоскость коррекции запрограммирована. Случайное изменение плоскости коррекции (функции G17,G18, G19) не должно быть выполнено при включенной коррекции радиуса.

Коррекция может находиться также в блоках без движения в плоскости коррекции, или система перейдет более чем через 200 блоков без движения в плоскость коррекции и движение после этого продолжается, как будто бы неподвижные блоки или движение в другой, чем плоскости коррекции не были запрограммированными.



G41 – это коррекция налево, т.е. центр инструмента движется (если коррекция радиуса положительное число) налево от запрограммированной траектории по направлению движения.

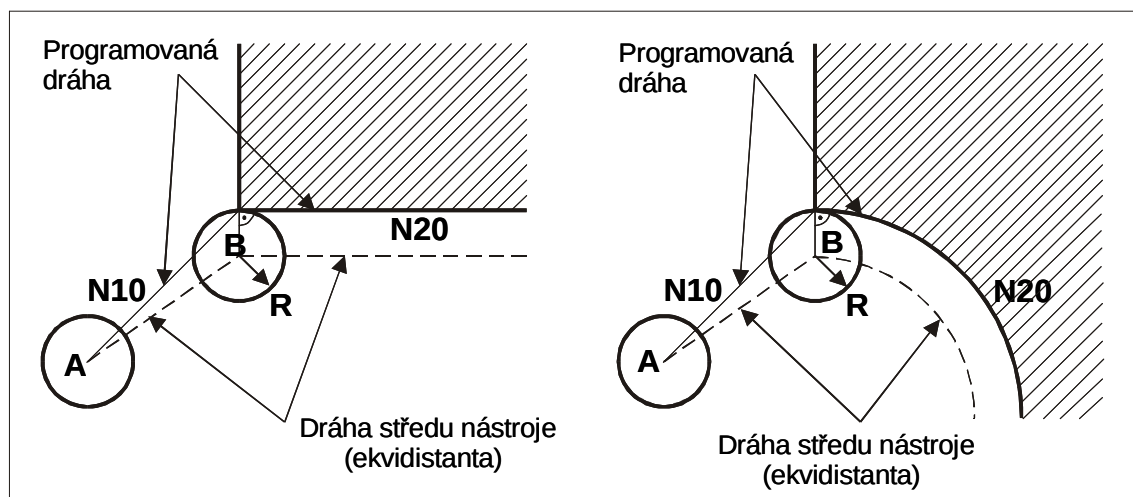
G42 – это коррекция направо, т.е. центр инструмента движется (если коррекция радиуса положительное число) направо от запрограммированной траектории по направлению движения.

Если бы величина коррекция была отрицательной, будут меняться стороны коррекции, т.е. G41 с отрицательной коррекцией совпадает с G42 с положительной коррекцией.

7.2.1 Включение коррекции радиуса и ход коррекции

Во время включения коррекции радиуса действительно однозначно следующее правило:

Коррекцию можно включить только в случае линейной интерполяции. Траектория центра инструмента движется с начальной точки блока N перпендикулярно траектории блока N+1, т.е. блок N+1 весь будет выполнен уже с включенной коррекцией. На рис. 1 в блоке N10 включена коррекция радиуса G42, траектория, подлежащая коррекции, изображена штриховой линией с точки A в точку B. Точка B, которая устанавливается, находится перпендикулярно траектории блока N20. Одинаковым способом будет определена точка для ввода коррекции в случае, если движение продолжается по окружности, как указано на том же рисунке направо. Точка B также находится перпендикулярно касательной запрограммированной окружности.



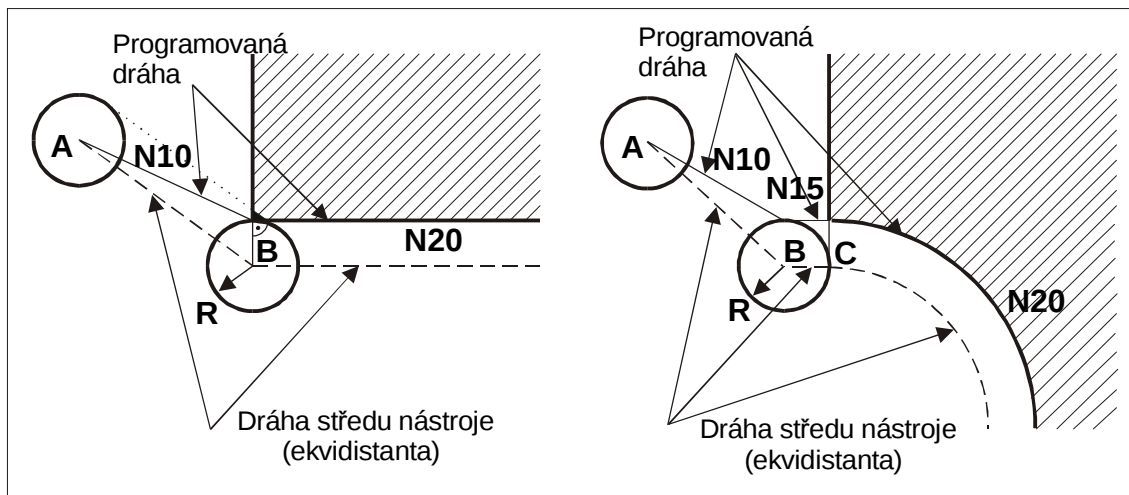
Запрограммированная траектория

Траектория центра инструмента
(эквидистанта)

Рис.1

В случае включения коррекции необходимо выбрать правильное направление установки точки. На рис. 2 налево указан ошибочный угол установки.

При этом направлении установки на коррекцию может произойти ошибочное обрабатывание продукта, потому что контурами инструмента перережете обрабатываемый продукт до того, как он будет установлен перпендикулярно следующего блока.



Запрограммированная траектория

Траектория центра инструмента
(эквидистанта)

Рис.2

Более подходящим способом тогда является установка коррекции перед материалом, как указано на рис. 2 направо. Если будет коррекция установлена перед обрабатываемым продуктом, что является обыкновенным способом, потом угол установки не имеет значение. Установка коррекции заканчивается в точке В, в точку С (блок N15) уже программа движется по включенной коррекции.

Если коррекция включена, движется центр инструмента по эквидистантам (обозначены штрихованной линией), находящимся на расстоянии с радиусом инструмента от запрограммированной траектории. На рис. 3 указаны примеры траектории, подлежащей коррекции, для коррекции налево и направо.

Таким способом будет выполнена коррекция радиуса в версиях ниже, чем 30.27 и в версиях 30.27 и выше в случае, если будет установлена машинная константа 339, 8 декада на 0.

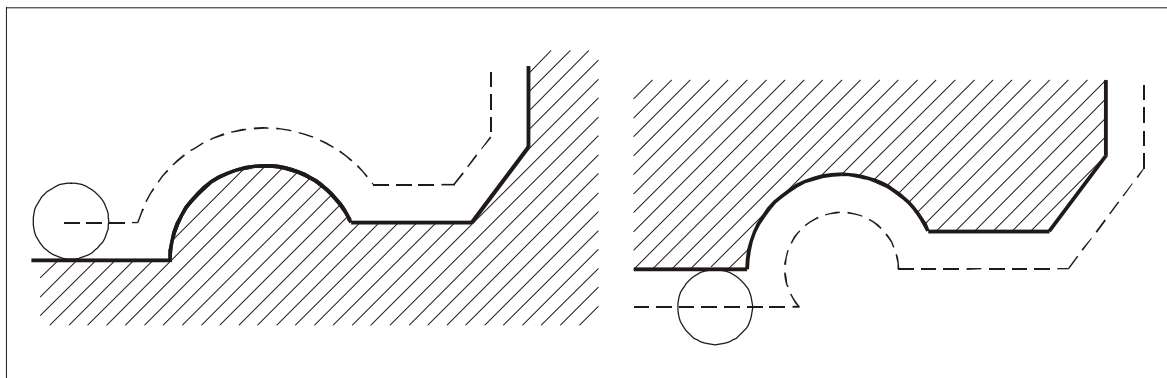
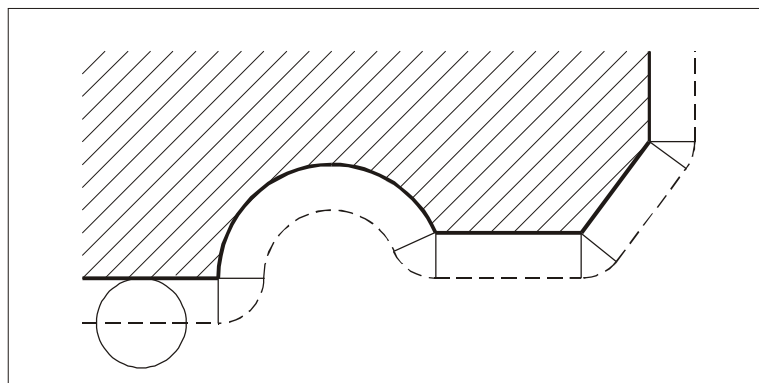


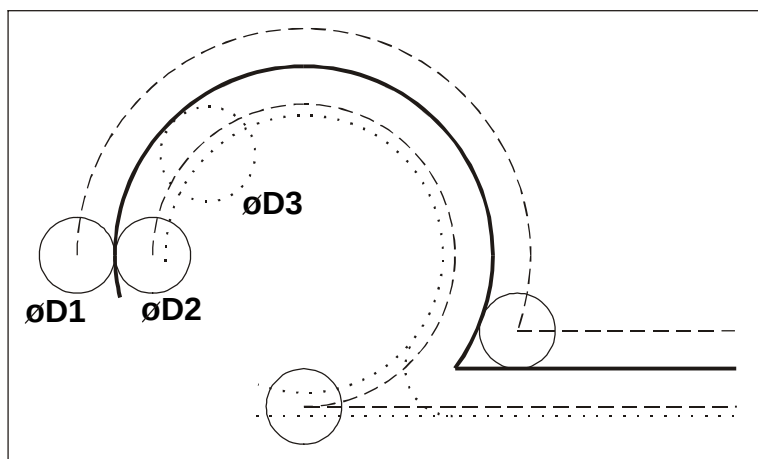
Рис. 3

В версии 30.27 и выше в случае, если будет установлена машинная константа 339, 8 декада на 1, будут вкладываться в случае разрыва дуги, как указано на рис. 4.



Дуги вкладываются, если угол больше, чем 180 градусов (плюс угол допуска, установленный в машинной константе 339 – см. ниже). Для меньших углов будет применена траектория, подлежащая коррекции, на пересечении эквидистант, т.е. одинаковым способом, как указано на рис. 3 налево, т.е. для углов меньше, чем 180 градусов ход коррекции одинаков без учета установки 8 декады машинной константы № 339.

Рис. 4



Если движется центр инструмента на пересечении эквидистант (339, 8дек.=0), необходимо во время программирования заботиться о том, чтобы всегда существовало пересечение эквидистант. Если эквидистанты не пересекутся в двух по себе следующих блоках, система сообщает ошибку «НЕ НАЙДЕНО ПЕРЕСЕЧЕНИЕ ЭКВИДИСТАНТ». Этот случай может произойти в случае неподходящей комбинации коррекции и угла касательных в точке, в которой пересекается траектория двух по себе последующих блоков. На рис. 5

указан пример в случае программирования окружности и прямой линии.

Рис. 5

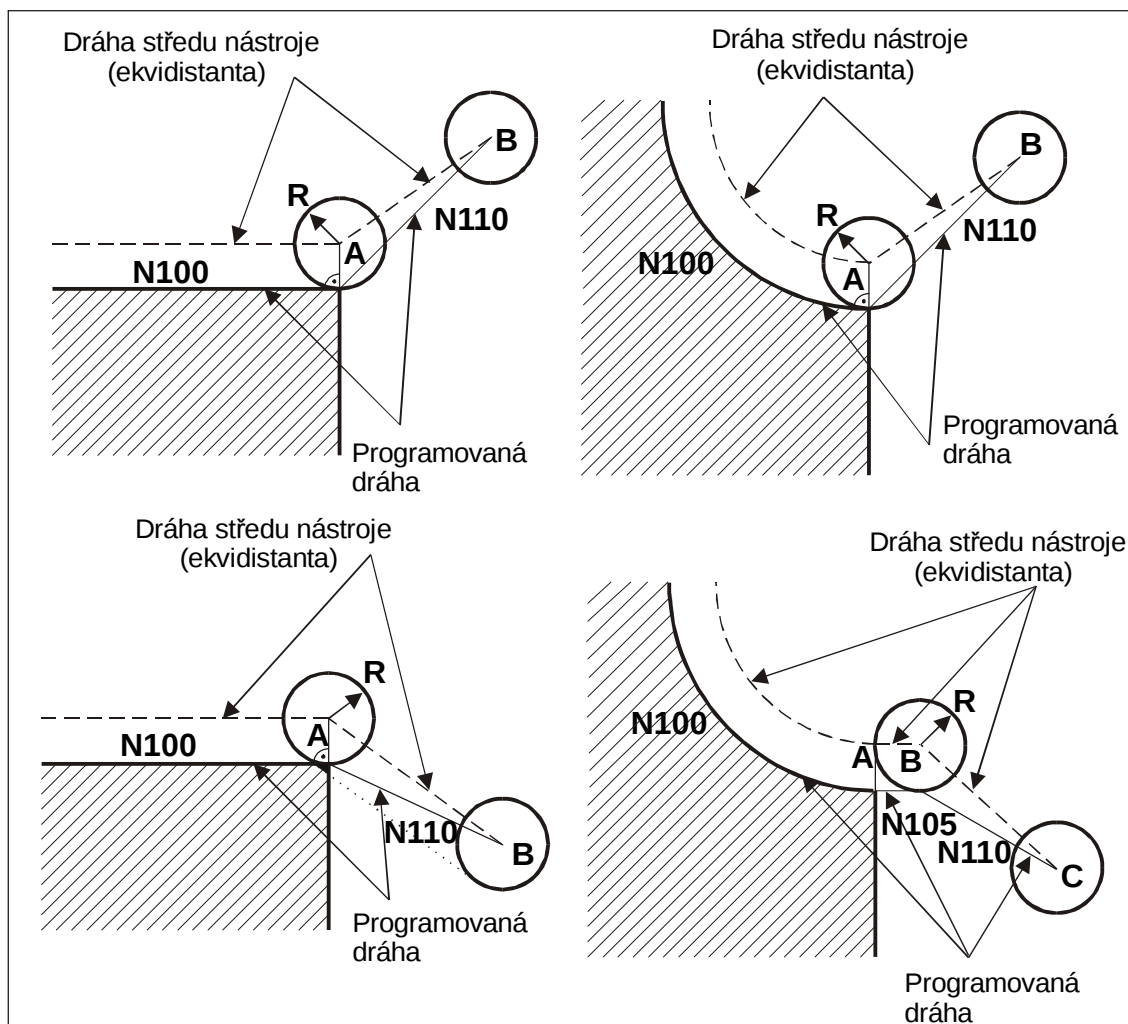
Во время коррекции налево G41 – инструмент диаметром D1 – не возникнут никакие проблемы, эквидистанты всегда пересекутся. Во время коррекции направо G42 и инструмента диаметром D2 эквидистанты также пересекутся, хотя инструмент уже находится относительно далеко. Во время коррекции направо G42 и диаметре инструмента D3, однако, эквидистанты уже не пересекутся (траектория на рис. 5 обозначена точечной линией) и система сообщает указанную ошибку. **Для новых версий и во время установки машинной константы (339, 8дек=1) будет, однако, вложена в указанном случае дуга (также обозначена точечной линией). Вкладывание дуги однозначно решает случаи, кроме того, будет также траектория инструмента короче, чем сокращается время обработки. Для новых версий рекомендуется использовать этот способ.**

Пометка:

В некоторых случаях (если не вкладывать дуги) будет, однако, найдено пересечение эквидистант, но оно находится далеко. Обычно речь идет о технологически неподходящих случаях, которые на практике не применяются. Если они появятся, необходимо выбрать более подходящий способ программирования.

7.2.2 Выражение коррекции радиуса

Отзыв коррекции радиуса выполняется функцией G40 и возможен, также как переключение G41, G42, только в случае линейной интерполяции. Окончательная точка, подверженная коррекции, последнего блока до G40 находится на перпендикуляре к касательной, которая проходит окончательной программированной точкой. Отзыв коррекции указан на рис. 6. Также как в случае включения коррекции должно соблюдаться направление блока, в котором находится программируемая функция G40. Это не действительно для случая, если коррекцию отзываем уже вне обрабатываемого продукта.



Траектория центра инструмента
(эквидистанта)

Программируемая
траектория

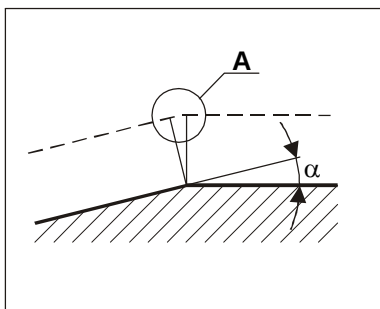
Рис. 6

На рис. 6 находится отзыв коррекции G40, запрограммированный в блоке движения (т.е. движение запрограммировано хотя бы в одной оси плоскости коррекции). Установкой машинной константы 339 (7 декада = 1) можно запрограммировать G40 также в блоке без движения. Этот способ, однако, не рекомендуется использовать.

7.2.3 Предельный угол для вкладывания обрабатываемых продуктов

Действительно для версии 30.27 и выше

В машинную константу 339 можно ввести предельный угол для вкладывания дуг в случае коррекции радиуса. Использование вытекает из примера – траектория центра инструмента обозначена штриховой линией. В машинной константе 339 вводим предельный угол α_{limit} напр. 15 градусов. После этого в случае (см. рис. налево), если угол $\alpha > \alpha_{\text{limit}}$, будет вложена в случае разрыва дуга (деталь A), в случае, если угол $\alpha \leq \alpha_{\text{limit}}$, дуга не будет вложена, но траектория будет проходить через пересечение эквидистант.



Установка машинной константы 339 узко связана с машинной константой 39, в которую вводится угол допуска для плавного продолжения. Если угол траектории, который находится между двумя блоками, меньше чем здесь введенный предел, считается траектория плавной, и она будет пройдена (в случае программирования G23) без изменения скорости. Предельный угол для вкладывания дуг в случае коррекции радиуса, введенный в машинной константе 339, бы поэтому должен был быть меньше или максимально равен углу, введенному в машинной константе 39. Если будет предельный угол для вкладывания дуги (константа 339) больше, чем предельный угол для плавной связи (константа 39), будет существовать угол плавности блока, при котором будет запрещено плавное перемещение (система будет замедлять скорость и опять ее набирать).

7.3 Коррекция радиуса для токарных станков

В случае использования коррекции радиуса может быть запрограммирован контур части (чертежные величины). В случае изменения инструмента (радиуса лезвия) меняется только величина коррекции радиуса в таблице.

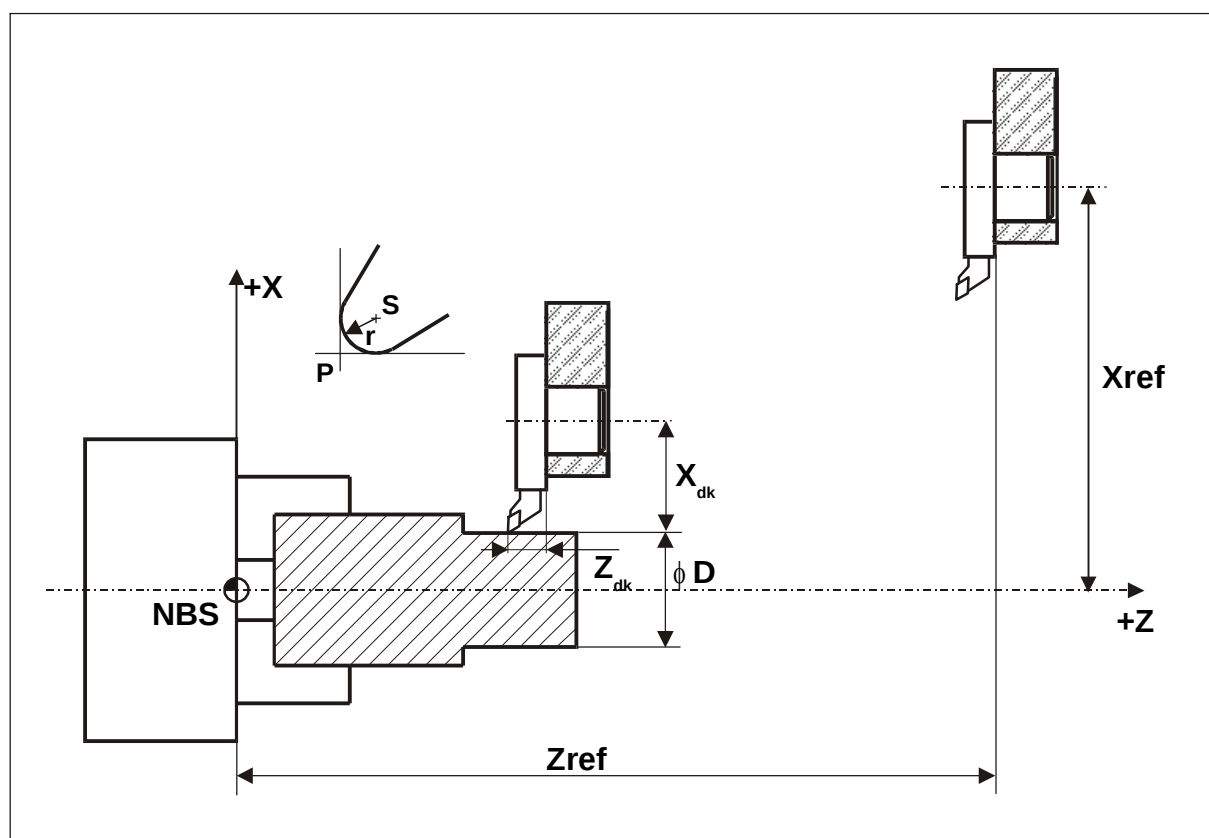


Рис. 7

Во время введения коррекции длины для токарных станков эта коррекция относится к теоретической точке лезвия Р. Эта точка не находится (обычно) в центре лезвия, но на пересечении касательных к радиусу лезвия г, как указано на рис. 7.

Чтобы система могла правильно рассчитать пройденную траекторию по эквидистанте, должно быть кроме радиуса лезвия введено также его положение. Для расчета коррекции радиуса потом система использует кроме данных о радиусе инструмента (лезвия) также данные о том, как нож укреплен в держателе. Возможные положения ножа указаны на рис. 8.

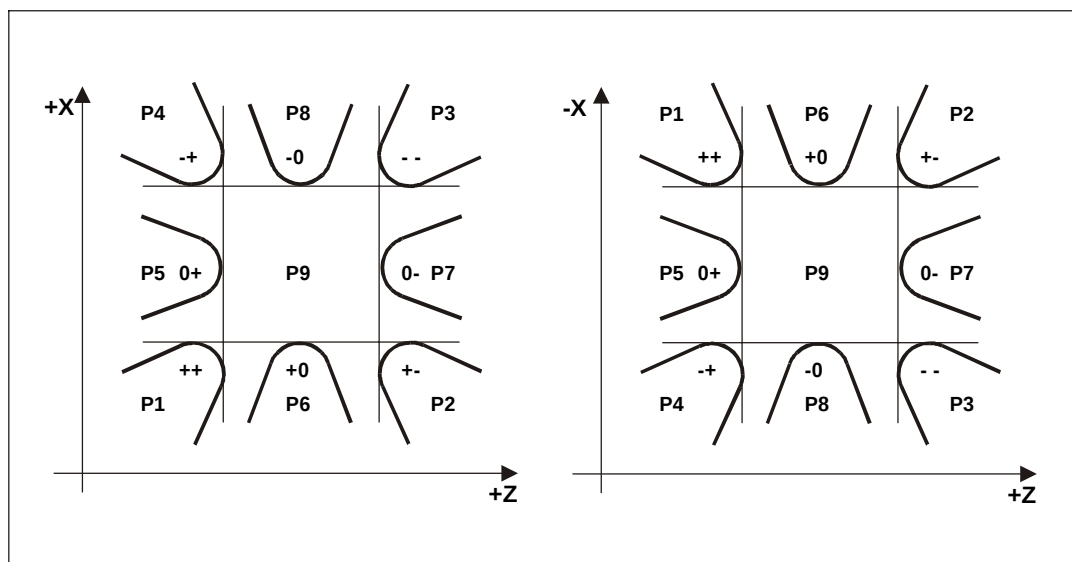


Рис.8

Пометка:

Оба рисунка являются схожими, только они являются зеркальным отражением по оси Z. Для определения положения инструмента будет выбран тот, у которого согласованы направления оси X с конкретным токарным станком. Знаки, указанные для отдельных положений лезвия, записаны в конфигурационном файле CNC836.KNF в параметрах \$51 (для первой головки инструмента) и \$52 (для случайной второй головки инструмента). Если токарный станок обладает только одной головкой инструмента, установка параметра \$52 не имеет значения (будут указаны напр. одинаковые величины как для первой головки инструмента). Знаки определяют направление аддитивного сдвига теоретической вершины лезвия инструмента относительно центра радиуса лезвия.

Влияние аддитивного сдвига, которое учитывает использованное положение инструмента, указано на рис. 9. На рис. 9А указана траектория без использования коррекции радиуса. Теоретическая вершина лезвия инструмента движется по траектории, которую запрограммирует технолог – в этом случае траектория запрограммирована на грань обрабатываемого продукта. Если будут запрограммированы траектории только параллельных с системой координат, т.е. параллельных только с осями X и Z, не должна коррекция программироваться, т.е. траектория с коррекцией и без коррекции должны совпадать.

На рис. 9В использована коррекция радиуса налево G41, но в таблицу коррекций не введен тип инструмента (не указано Р) или не введено Р = 9. Инструмент типа Р9 имеет теоретическую вершину в середине лезвия, поэтому обладает аддитивным сдвигом в обоих направлениях равным нулю. Как видно по рисунку, коррекция G41 будет включена (вершина движется с точки V1 в точку V3), но потому что окончательная точка V3 не подлежит коррекции с аддитивным сдвигом, инструмент будет двигаться мимо обрабатываемого продукта.

На рис. 9С указано правильное использование коррекции радиуса. В таблице коррекций должен быть для данной коррекции радиуса введен тип инструмента – в нашем случае Р = 3, который имеет по рис. 8 аддитивный сдвиг в оси X минус и в оси Z также минус. Для размера радиуса с учетом указанных знаков

сдвинется теоретическая вершушка V3 с рисунка 9В в положение V3 на рис. 9С, что является правильным положением лезвия с учетом обрабатываемого продукта. Траектория теоретической вершушки указана на рис. 9С штриховой линией с точки V1 в точку V3.

Использование коррекции радиуса проявится, прежде всего, в случае траектории, которая не является параллельной ни одной оси.

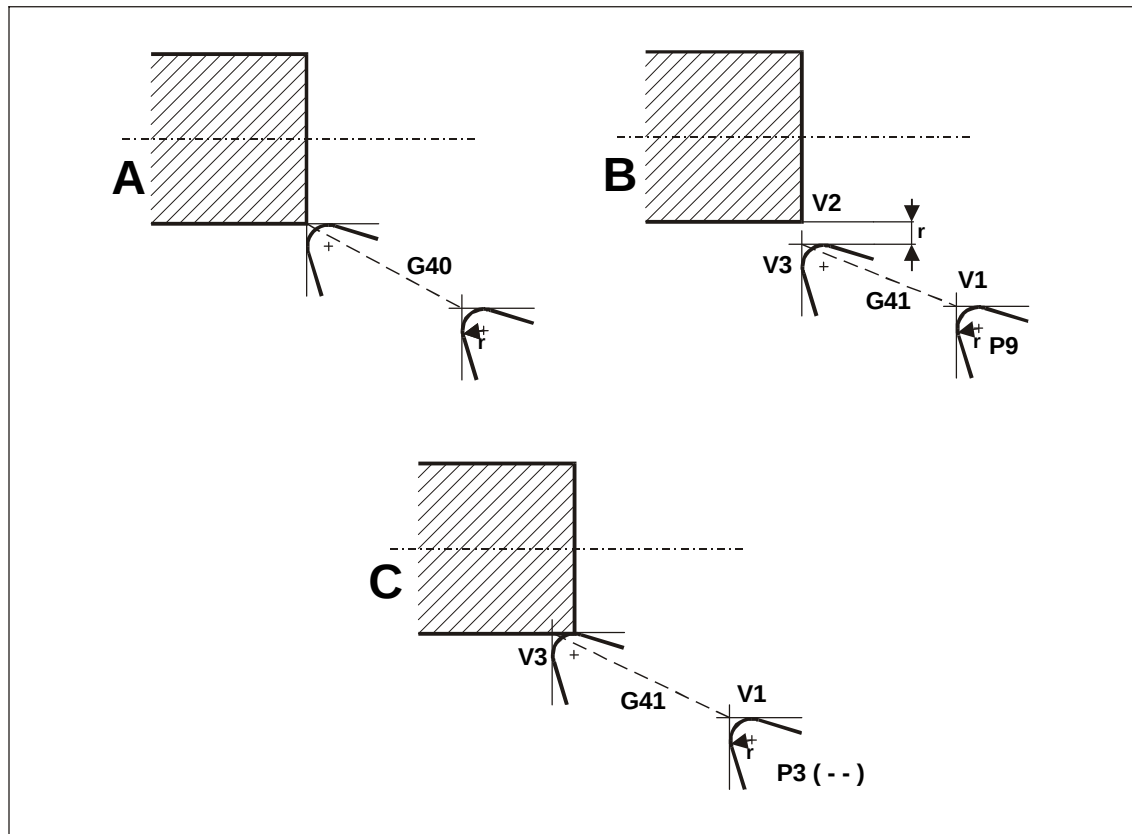


Рис. 9

7.4 Индикация для токарных станков при использовании коррекции радиуса

Индикация положения в системе включает аддитивный сдвиг. Какие будет система показывать величины в отдельных блоках, мы укажем на примере части партпрограммы.

Содержание файла с коррекциями TAB0.KOR будет следующим:

```
$KOR
01: R=0.8 X=120.0 Z=340.5 P=3
```

```
....
```

```
....
```

В файле для коррекции указан номер D1 и записана коррекция радиуса 0.8 мм. Предполагаем, что коррекция длины установлена для инструмента номер T1, положение лезвия согласно рис. 8 - это P = 3. Система координат, принятая для токарных станков:

Направо Z +

Наверх X -

Часть партпрограммы будет иметь следующую форму (предполагаем программирование диаметра в оси X):

%1	
N5 G90 G54 G40 &1100 D1 T1	„ВЫБОР ИНСТРУМЕНТА И ТАБЛИЦЫ КОРРЕКЦИЙ
N10 X 20 Z50	„УСТАНОВКА ИСХОДНОГО ПОЛОЖЕНИЯ
N20 X0 Z0 G41	„ВКЛЮЧЕНИЕ КОРРЕКЦИИ
N30 X20	
N40 Z-50	
N50 X 40 Z-75	
N60 Z-100	
N70 X50	„ПОСЛЕДНИЙ БЛОК С КОРРЕКЦИЕЙ
N80 X60 G40	„ОТЗЫВ КОРРЕКЦИИ

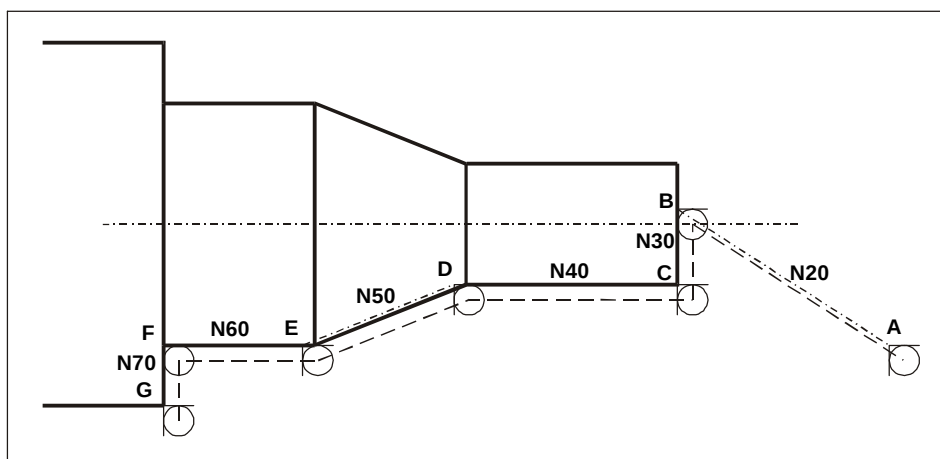


Рис. 10

На рисунке 10 нарисована траектория лезвия по указанной партпрограмме (радиус лезвия для наглядности в несколько раз увеличен).

Индикация на концах отдельных блоков указана в таблице:

КОНЕЦ БЛОКА	ТЕОРЕТИЧЕСКАЯ ВЕРШИНА В ТОЧКЕ	ИНДИКАЦИЯ X (ДИАМЕТРА)	ИНДИКАЦИЯ Z
N20	B	-1.6	0.0
N30	C	+20.0	0.0
N40	D	+20.0	-50.646
N50	E	+40.0	-75.646
N60	F	+40.0	-100.0
N70	G	+48.4	-100.0
N80		+60.0	-100.0

В блоке N70, который является последним блоком до отзыва коррекции, находится середина лезвия на перпендикуляре к конечной точке и теоретическая вершушка находится на радиус ближе к оси. Величина 48.4 является диаметром 50 минус 2 раза радиус лезвия 0.8. (Индикация является по диаметру!)

Пометка:

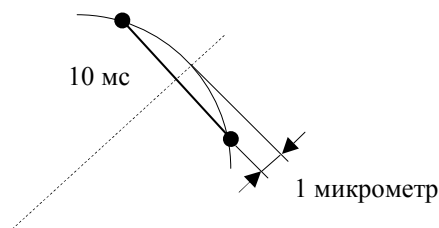
В случае графического изображения траектории нарисовать теоретическую вершушку лезвия. Кроме блоков N20 и N50 (на рис. 10 обозначены штриховой линией между точками A-B и D-E) будет траектория, подверженная коррекции, совпадать с запрограммированной.

7.5 Управление скоростью в случае коррекций диаметра

Система управляет скоростью отдельных координат так, что запрограммированная скорость в блоке определяет вектор в пространстве, направление которого всегда является касательным к данной траектории, и его составные части в системе координат определяют скорость в отдельных осях.

Программируемая скорость в блоке равна окончательной касательной скорости, чем обеспечены постоянные условия резки для обрабатываемого материала. Изменения действительной касательной скорости от программируемой скорости могут быть причинены следующими факторами:

- Изменение скорости с учетом введенного процента от скорости (%F).
- Возможность снизить скорость во время круговой интерполяции с учетом точности от формы идеальной округлости. Ограничение скорости проявляется на малых кругах. Расчетный такт интерполятора системы составляет 10 мс и поэтому это на самом деле круговая интерполяция, состоящая из линейных отрезков движения, которые длятся 10 мс. Система стандартно ограничивает скорость во время круговой интерполяции таким способом, чтобы отклонение обладало идеальной округлостью меньше, чем 1 микрометр. Если это разрешено динамикой станка и не надо соблюдать отклонение от идеальной округлости 1 микрометр, можно управлять ограничением скорости на круговой интерполяции при помощи машинной константы R232 (см. приложение F).



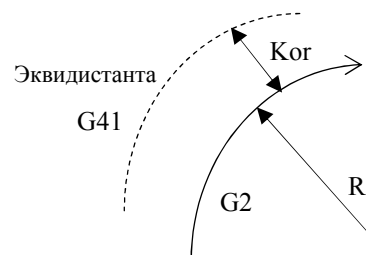
- При использовании коррекции радиуса система стандартно влияет на действительную касательную скорость таким способом, чтобы скорость в точке соприкосновения инструмента и материала была той, которая запрограммирована в блоке. Этим способом обеспечены константные условия резки. Влияние на скорость происходит только в случае круговой интерполяции. Действительная окончательная скорость больше или меньше в отношении актуальной коррекции радиуса и радиуса окружности. Система на самом деле соблюдает константную угловую скорость.

радиус окружности без коррекций радиуса составляет:	$R = \text{SQR}(I^2 + J^2)$
программируемая скорость в блоке составляет:	F
актуальная коррекция радиуса составляет:	Kor
действительная касательная скорость составляет:	Fsku

Действительно для отдельных случаев:

Круглая интерполяция G2 и коррекция радиуса налево G41:

Действительная касательная скорость будет **больше**, чтобы была сохранена программируемая скорость по месту соприкосновения инструмента и материала. Если установлена коррекция налево, материал должен обрабатываться всегда с правой стороны инструмента, в противоположном случае будет эффект влияния скорости проявляется наоборот и не будут соблюдаться условия резки.

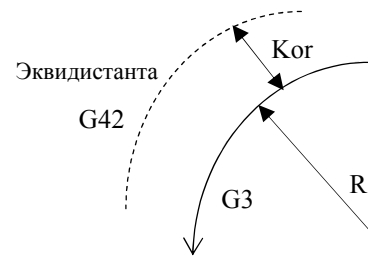


Для действительной скорости будет действительно:

$$Fsku = F * [(R + Kor) / R]$$

Круговая интерполяция G3 и коррекция радиуса направо G42:

Действительная касательная скорость будет **больше**, чтобы была сохранена программируемая скорость по месту соприкосновения инструмента и материала. Если установлена коррекция направо, материал должен обрабатываться всегда с левой стороны инструмента, в противном случае будет эффект влияния на скорость проявляться наоборот и не будут соблюдаться условия резки.

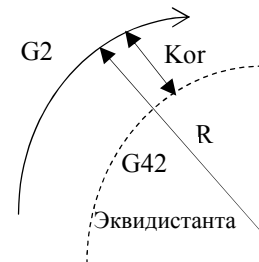


Для действительной скорости будет действительно:

$$F_{sku} = F * [(R + Kor) / R]$$

Круговая интерполяция G2 и коррекция радиуса направо G42:

Действительная касательная скорость будет **меньше**, чтобы была сохранена программируемая скорость по месту соприкосновения инструмента и материала. Если установлена коррекция направо, материал должен обрабатываться всегда с левой стороны инструмента, в противном случае будут эффект влияния на скорость проявляться наоборот и не будут соблюдаться условия резки.

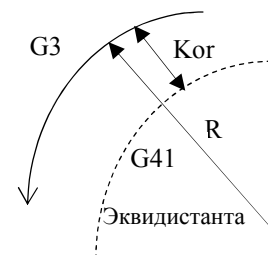


Для действительной скорости будет действительно:

$$F_{sku} = F * [(R - Kor) / R]$$

Круговая интерполяция G3 и коррекция радиуса налево G41:

Действительная касательная скорость будет **меньше**, чтобы была сохранена программируемая скорость по месту соприкосновения инструмента и материала. Если установлена коррекция направо, материал должен обрабатываться всегда с левой стороны инструмента, в противном случае будет эффект влияния на скорость проявляться наоборот и не будут соблюдаться условия резки.



Для действительной скорости будет действительно:

$$F_{sku} = F * [(R - Kor) / R]$$

Важная пометка:

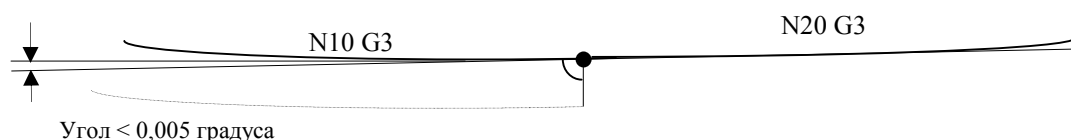
Если технологические причины не разрешают использовать коррекцию по выше описанному способу, можно влияние скорости в случае коррекции радиуса выключить при помощи пятой декады машинной константы R168. Если пятая декада машинной константы R168 установлена на величину 1, система не соблюдает константную угловую скорость на окружности, т.е. действительная касательная скорость не меняется, и если не произойдет ограничение скорости с учетом требуемой точности (R232), будет эта скорость равна запрограммированной скорости.

7.6 Тест неразрывности для коррекции радиуса

Система во время расчета коррекций радиуса вычисляет пересечение эквидистант. Во время расчета пересечения должна внутри выполняться подготовка следующих блоков (максимально 200 блоков вперед), чтобы обнаружить эквидистанту в блоках движения в плоскости коррекции. Существуют случаи, когда расчет пересечения является проблематичным. Такой проблемный расчет может получиться например, тогда, если запрограммированы 2 круговые дуги в следующих блоках, которые имеют огромные радиусы, центры которых находятся на расстоянии от окончательных мер блоков, но находятся близко друг другу. Директивы блоков в траектории движения в точке их соединения являются почти

одинаковыми. Введение таких блоков очень требовательное с точки зрения точности, потому что уже малая неточность (приблизительно микроны) обуславливает большое отклонение во время расчета пересечения эквидистант. Эта действительность возлагает большие требования на CAD системы для генерации NC программ. Если не возможно достичь требуемой точности, можно при помощи машинной константы **R234** ввести т. наз. „предел для теста непрерывности для коррекции радиуса“.

В константе **R234** вводится предел для теста непрерывности коррекций радиуса. Если угол между направляющими касательными в точке соединения соседних блоков меньше, чем установленный предел в константе R234, система не рассчитывает пересечение эквидистант, но рассчитает точку на перпендикуляре в точке соединения блоков. В этом случае может быть окончательная траектория точнее, чем пересечение эквидистант. Величина в константу R234 вводится в тысячных частях градуса. Если величина является нулевой, система рассчитывает в коррекции радиуса пересечение эквидистант всегда, кроме случая, когда соседние блоки плавно переходят друг в друга абсолютно по касательной. Константу рекомендуется установить на величине приблизительно 0.005 тысячных частей градуса. Если бы величина в константе R234 была несоразмерно большой, могло бы это негативно повлиять на точность эквидистанты.



7.7 Уточнение круговой интерполяции

С версии системы 30.16 и версии кассеты 5.023 (3.1.2000 г.) введена возможность автоматического уточнения введения круговой интерполяции.

Проблематика точности введения блока для коррекции радиуса с эквидистантой

Система CNC836 в случае запрограммирования движения по окружности использует стандартно определенный способ ввода. Это означает, что окружность вводится большим числом параметров, чем минимально необходимо (кроме координат начала и конца необходимо запрограммировать обе координаты центра I, J). Заранее определенный способ ввода позволяет системе например проверить правильность ввода круговой интерполяции и в случае ошибки сообщить, что окончательная точка не находится на окружности.

В расчете эквидистанты для коррекции радиуса используются отдельные параметры из введения блока, что требует в некоторых случаях **большие требования к точности ввода** всех параметров блока. Например, во время плавного перехода двух отрезков окружности (о больших радиусах) система внутри рассчитает систему квадратных уравнений, и правильный корень всех решений определяет по расстоянию до окончательной точки из введенного блока. Если этот размер не был введен с достаточной точностью, система может определить неправильный корень, что, например, может проявиться таким образом, что будет для эквидистанты выбран дополнительный круговой отрезок.

На практике иногда появляются проблемы, когда предлагаемая CAD система не работает с требуемой точностью во время генерации NC программы. В этом случае должны быть соответствующим способом установлены машинные константы **R234** для теста связи и **R55** для допуска середины окружности.

В CNC систему вводится т. наз. **препроцессор** для обработки партпрограммы (который кроме другого позволяет прямое использование функций диалоговой графики в NC программе). Одна из функций введенного препроцессора также **автоматически пересчитывать и уточнять введенную траекторию**

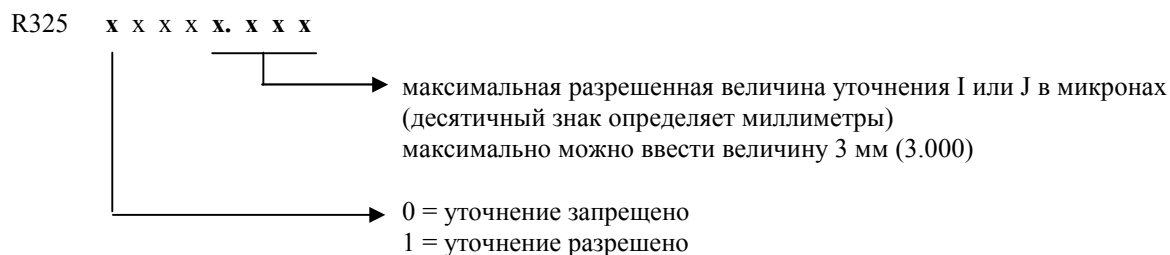
круговой интерполяции. Это новое свойство должно удалить проблемы с точностью ввода блоков в случае менее точных предлагаемых CAD систем.

Активация и принцип уточнения

Необходимые условия для активации уточнения являются следующими:

- 1.) В файле CNC836.KNF должен быть разрешен препроцессор **\$63 = 1** (первоначально установлено)
- 2.) Должна быть установлена машинная константа **R325**:

Установка машинной константы **R325**:



Препроцессор уточнит величину **I** или **J** до одной тысячной микрометра. В уточняемом блоке автоматически появится новый адрес **E**, который уточнение выполнит по отношению:

$$I_z = I + E/1000 \quad \text{или} \quad J_z = J + E/1000 \quad \dots I_z \text{ и } J_z - \text{это уточненные величины } I \text{ и } J$$

Восьмая декада адреса **E** – это признак, если должен быть уточнен адрес **I** или **J**. Если восьмая декада установлена на величину **0**, уточняется величина **I**, хотя она установлена на величину **1**, уточняется статья **J**. Величина уточнения в букве **E** – установлена в **тысячных долях микрометра**.

Примеры:

E2.054	$I_z = I + 2.054/1000$	(например для $I = 12.345$ будет $I_z = 12.347054$)
E-10006.992	$J_z = J - 6.992/1000$	(например для $J = 5.088$ будет $J_z = 5.081008$)

Величины для уточнения, рассчитанного препроцессором, можно для информации контролировать в форме листинга блока.

7.8 Выбор плоскостей коррекции и интерполяции

Действительно от версии 30.29

Для блока с круговой интерполяции определена плоскость интерполяции запрограммированными координатами. В некоторых интерполяционных плоскостях в зависимости от типа станка необходимо заменить очередность осей в круговой интерполяции, что влияет на программирование, или причисление **I** и **J** к осям интерполяции. Каждый станок по своему типу обладает однозначно определенным множеством плоскостей интерполяции.

В случае коррекций радиуса при помощи функций **G17**, **G18**, **G19** и **G14**, **G15** и **G16** выбирается актуальная плоскость коррекции во всех возможных плоскостях интерполяции станка, которых может быть больше, чем плоскостей коррекций. Технолог имеет возможность в случае коррекций радиуса использовать максимально 6 плоскостей коррекций.

Для определения плоскостей интерполяции и коррекции определены в машинной константе номера 340 и 341.

Константы разделены на 8 двухдекадных частей. В каждой части (двухместный номер) определены порядковые номера осей плоскости интерполяции.

Первая цифра из пары (большая декада) определяет порядковый номер первой координаты в плоскости

интерполяции (диапазон 1 – 6)

Вторая цифра из пары (нижняя декада) определяет порядковый номер второй координаты в плоскости интерполяции (диапазон 1 – 6) .

Одновременно действительно:

Первая пара из константы 340 определяет одновременно плоскость коррекции, вызовом функции G17 (первоначально 12).

Вторая пара из константы 340 определяет одновременно плоскость коррекции, вызовом функции G18 (первоначально 31).

Третья пара из константы 340 определяет одновременно плоскость коррекции, вызовом функции G19 (первоначально 23).

Четвертая пара из константы 340 определяет только плоскость интерполяции – нельзя ее использовать для коррекции радиуса.

Первая пара из константы 341 определяет одновременно плоскость коррекции, вызовом функции G14 (первоначально 34).

Вторая пара из константы 341 определяет одновременно плоскость коррекции, вызовом функции G15 (первоначально 24).

Третья пара из константы 341 определяет одновременно плоскость, вызовом функции G16 (первоначально 41).

Четвертая пара из константы 341 определяет только плоскость интерполяции – нельзя ее использовать для коррекции радиуса.

Первоначальное состояние констант 340 и 341 является двойным.

Если здесь записаны одни нули, действительны плоскости интерполяции и коррекции по выше указанному перечню, т.е. как будто было бы записано следующее:

R340: 00233.112

R341: 00412.434

Пометка: Десятичный знак не обладает никаким значением.

7.9 Коррекция длины

7.9.1 Коррекция на длину инструмента

От версии пульта 30.29 может быть использована коррекция на длину инструмента для всех шести координат (до этой версии это были только 4 координаты).

От указанной величины порядок использования коррекций длины указан машинной константой 329, т.е. 7 декадой. Ее значение является следующим:

A	329	[7дек.] = 0	Старший способ, для расположения и отзыва коррекции длины, использует управляющую функцию &, коррекция только для первых 4 осей
B	329	[7дек.] = 1	Новый способ, не использует управляющую функцию &, коррекцию длины можно выбрать для всех осей, расположение осуществляется запрограммированием адреса Dxx, где xx это номер таблицы коррекций, отзыв выполним запрограммированием D0
C	329	[7дек.] = 0	Новый способ, используется порядочная функция & только для включения и отзыва коррекции, только величины 0 = отзыв коррекции, 1 = включение коррекции по Dxx, где xx это номер таблицы коррекций.

7.9.2 Коррекция длины – способ управления A

Соответствующая коррекция на длину инструмента выбирается управлением функцией & и адресом D. Величины отдельных коррекций длины уложены в таблицу коррекций и выбираются адресом D.

Сопряжением выбранных величин коррекций длины с соответствующей величиной таблицы коррекций, выбранных адресом D, в отдельные координаты, управляется т. наз. "управляющей функцией для сопряжения коррекций длины", запрограммированной под адресом &. Функция является четырехдекадной (k1, k2, k3, k4), отдельные декады k1, k2, k3, k4 могут принимать значение только как цифровые величины 0,1,2.

Во время программирования &k1k2k3k4 обладают отдельные декады следующим значением:

k1=0 – в координату X вводится нулевое значение коррекции длины
 k1=1 – к координате X причисляется величина коррекции длины, т.е. II. данное соответствующей таблицы коррекции, выбранное под адресом D.
 k1=2 – от координаты X будет вычитана величины длины коррекции (II данное соответствующей таблицы коррекций).
 k2=0 – в координату Y будет записана нулевая величина коррекции длины.
 k2=1 – к координате Y причисляется величина коррекции длины (III данное соответствующей величины таблицы коррекции).
 k2=2 – от координаты Y вычитается величина коррекции длины. (III данное соответствующей величины таблицы коррекций).
 k3=0 – в координату Z будет введена нулевая величина таблицы коррекций
 k3=1 – к координате Z прибавляется величина коррекции длины. (IV данное соответствующей величины таблицы коррекций)
 k3=2 – от координаты Z вычитается величина коррекции длины. (IV данное соответствующей величины таблицы коррекций).
 k4=0 – в координату 4 будет записана нулевая величина коррекции длины.
 k4=1 – к координате 4 прибавляется величина коррекции длины. (V данное соответствующей величины таблицы коррекций).
 k4=2 – от координаты 4 вычитается величина коррекции длины. (V данное соответствующей величины таблицы коррекций).

Управляющая функция & обслуживает только выбор величины коррекции длины из соответствующей величины таблицы коррекций. Номер величины определен адресом D. Управляющая функция & и функция D – являются постоянными, т.е. действительны до времени, пока они не изменены.

Пример:

N10 &1201 D12 ... К координате X прибавляется величина второй статьи таблицы коррекций номер 12, от координаты Y вычитается величина из третьей статьи той же таблицы, к координате Z не прибавляется и ни к какой величине, к четвертой координате прибавляется величина из пятой статьи таблицы № 12.

Коррекция длины относится исключительно к пространству XYZ и 4 координатами, и предназначена исключительно для простого сдвига в этом пространстве, ни в коем случае для компенсации радиуса обрабатываемого инструмента. Она может принимать макс. величину +/- 999,999 мм.

В случае включения коррекции длины происходит движение по траектории, которая только линейно сдвигается на величину коррекции длины в отдельных осях и ее форма сходится с запрограммированной формой или формой случайно находящейся под влиянием включенной коррекции радиуса. Коррекция длины включается только в блоке, в котором запрограммирована линейная интерполяция, или быстрый сдвиг. Инструмент в следующей программе придерживается траектории подлежащей коррекции, если коррекция не будет отключена запрограммированием функции &0, или пока не запрограммирована новая величина коррекции длины.

На рисунке указан пример движения инструмента во время коррекции длины. Траектория партпрограммы без коррекции длины находится на рис. жирной линией. Тонкой линией обозначено прохождение партпрограммы во время включения коррекции длины. Партпрограмма для всех рисунков является одинаковой.

Для прохождения коррекции длины согласно а) и б) заполним таблицу коррекций напр. следующим способом:

N1 G92 D1 R1=125.0 R2=80.0

Для прохождения коррекции длины согласно с) заполним таблицу коррекций напр. следующим способом:

N1 G92 D1 R1=80.0 R2=50.0

Остальные блоки партпрограммы могут быть записаны напр. следующим способом (предполагаем, что координаты X и Y находятся в нулевом положении):

N10 X100.0 Y120.0 G1 F1000 D1 &1000 " согласно рис. а) "

N10 X100.0 Y120.0 G1 F1000 D1 &100 " согласно рис. б) "

N10 X100.0 Y120.0 G1 F1000 D1 &2200 " согласно рис. в) "

N20 X350.0

N30 Y250.0

N40 X100.0

N50 Y120.0

N60 X0 Y0 &0

" отмен коррекции длины"

N70 M30

Надо отметить, что направление коррекций длины в этом случае определено управляющей функцией &. Того же результата можно достичь, если во время заполнения таблицы (блок N1) будут введены отрицательные величины, и в величину управляющей константы & мы вместо 1 запишем 2 и вместо 2 запишем 1. Блок N10 будет потом выглядеть следующим способом:

N10 X100.0 Y120.0 G1 F1000 D1 &2000 " согласно рис. а) "

N10 X100.0 Y120.0 G1 F1000 D1 &200 " согласно рис. б) "

N10 X100.0 Y120.0 G1 F1000 D1 &1100 " согласно рис. в) "

Во время обработки отдельных блоков указанной программы деятельность системы является следующей:

Программирование коррекции радиуса и длины в одном блоке партпрограммы.

В одном блоке партпрограммы могут быть одновременно запрограммированы коррекции радиуса и длины даже в одинаковой плоскости. Должны соблюдаться правила для программирования, указанные выше. Система во время обработки программы регистрирует самостоятельно коррекцию радиуса и ее разложение в отдельных осях и самостоятельно коррекции длины в отдельных осях. Окончательные точки тогда определяются как сумма коррекции радиуса и длины.

7.9.3 Коррекция длины – способ управления В

От версии пульта 30.29 может использоваться коррекция на длину инструмента для всех шести координат, которая не должна быть запрограммирована, т.е. можно использовать управляющую функцию &.

Коррекция длины управляется только запрограммированным адресом Dxx, где xx это номер таблицы коррекций. Коррекция включается во всех осях, которые не обладают в таблице нулевой величиной. Отзыв коррекции запрограммирован в D0.

Пометка:

В случае управления при помощи & могут находиться в таблице не нулевые значения для данной оси, но если коррекция будет включена, это повлияет только на управляющую функцию &. Это является разницей по сравнению с управлением только адресом D, где принимает решение содержание таблицы для данной оси.

7.9.4 Коррекция длины – способ управления С

От версии пульта 30.29 можно использовать коррекцию на длину инструмента для всех шести координат.

Этот способ является комбинированным способом управления согласно А и В. Для включения и отзыва используется управляющая функция &, но она принимает значение только двух величин:

- 0 – отзывает коррекцию на длину
- 1 – включает коррекцию на длину

Номер таблицы коррекций определен под адресом D.

8

8. ПРОГРАММИРОВАНИЕ ТЕХНОЛОГИЧЕСКИХ ФУНКЦИЙ

Значение некоторых функций, указанных в этом разделе, может быть модифицировано автором интерфейса для данного станка. Точное значение и перечень использованных М-функций, или Р и Н-функций должен сообщить автор интерфейса станка.

8.1 Временная задержка

Временная задержка программируется функцией G04 из группы G10. Эта функция обладает ограниченным действием только в блоке, в котором она была запрограммирована. Длительность временной задержки определена величиной функции Q, которая занимает четыре декады и может тогда принимать значение 0 - 9999. Одна единица выражает временную задержку 10 мс, т.е. можно запрограммировать временную задержку в пределах от 10мс до 99,99 сек. по 10мс.

Пример:

Q10 10 x 10мс = 100мс
Q100 100 x 10мс = 1000мс = 1сек.
Q1000 1000 x 10мс = 10000мс = 10сек.

8.2 Обороты шпинделя

Обороты шпинделя запрограммированы адресом S. Величину оборотов можно выбрать в 4 декадах, т.е. в пределах 0 - 9999 об./мин. Включение диапазона оборотов степени перевода запрограммировано функциями M41, M42, M43, M44.

Старт вращения в направлении CW вводится функцией M03, старт в направлении CCW вводится функцией M04. После старта движения этими функциями длится вращение так долго, пока не запрограммирована функция M05 (СТОП шпинделя) или M19 (СТОП шпинделя в ориентировочной точке).

Система может передавать на своем выходе функцию S или BCD код, или чаще передает на своем выходе аналоговое напряжение в пределах +/- 10В для возбуждения регулятора привода шпинделя.

8.2.1 Стоп шпинделя в ориентировочной точке и управление в связи с положением

Стоп шпинделя в ориентировочной точке будет запрограммирован функцией M19. После выполнения стопа в ориентировочной точке произойдет остановка шпинделя в определенной точке и закрытие связи

с положением. После этого можно управлять шпинделем в связи с положением.

Управление шпинделем в связи с положением служит для поворота шпинделя в любое положение с точностью $1/n$ оборота, где n это количество передаваемых импульсов от импульсного датчика на один оборот шпинделя. Управление в связи с положением практически обозначает переключение шпинделя на ротационную ось, скорость вводится в градусах за минуту. (F100 = обозначает скорость 100 градусов за минуту).

В одном блоке можно запрограммировать поворот шпинделя максимально на 69999, 999 градусов. Величина поворота будет запрограммировано в большинстве случаев под адресом C, A или B. Разъединение связи положений и переход на управление шпинделем в связи со скоростями произойдет при программировании функции M03, M04 или M05.

Программирование функции M19 можно только в случае, если она совместима с интерфейсом для соответствующего станка.

8.3 Охлаждение инструмента

Для управления охлаждением в системе предназначены 2 группы M функций [M5,M6], которые могут управлять 4 самостоятельными охлаждающими округами. Значение M функции указано в совокупном перечне адресов, используемых системами CNC8х6.

Уточнение функции для конкретного станка должен предоставить автор интерфейса данного станка (напр. если используются два округа охлаждения и т.п.).

8.4 Прерывание и конец партпрограммы

В системе можно запрограммировать четыре M-функции (M00,M01,M02,M30), которыми можно завершить или прервать обрабатываемую партпрограмму:

- | | |
|-----|--|
| M00 | - Безусловный стоп
Обрабатываемая партпрограмма программы будет прервана после выполнения всех операций в блоке, в котором запрограммирована эта функций. Одновременно в группах M2, M5 и M6 временно установится M-функция на величину, обозначенную + (см. таблицу значений адресов) т.е. M05 (стоп шпинделя), M00 (выключение охлаждения 1 и 2), M52 (выключение охлаждения 3 и 4). После старта партпрограммы кнопкой СТАРТ величины M-функции в группах M2, M5 и M6 вернутся в свое первоначальное значение, если не запрограммировано другое. |
| M01 | - Условный стоп
Система ведет себя аналогично как в случае M00, если работает в модификации режима AUT - M01. |
| M02 | - Конец партпрограммы
Конец партпрограммы с обратным возвращением в начало партпрограммы. Будут отправлены заключительные функции M05, M09, M52. |
| M30 | - Система ведет себя аналогично как при M02. |

Пометка:

Автор интерфейса для конкретного станка может поведение этих функций до определенного уровня модифицировать.

8.5 Укрепление и освобождение обрабатываемого продукта

Функцией M10 выдан приказ для укрепления обрабатываемого продукта и функцией M11 выдан приказ для освобождения обрабатываемого продукта. Самым циклом укрепления и освобождения управляет программируемый интерфейс системы. Функции M10 и M11 принадлежат к седьмой группе.

8.6 Смена инструмента и обрабатываемого продукта

Функцией M06 выдан приказ на смену инструмента. Самым циклом смены руководит программируемый интерфейс системы.

Функцией M60 выдан приказ на смену обрабатываемого продукта. Самым циклом смены управляет программируемый интерфейс системы. Функции M06 и M60 принадлежат к восьмой группе.

8.7 Вспомогательные М-функции группы M14

В этой группе можно запрограммировать любую одну величину М функции из величин, которые не были указаны в остальных группах (M1 - M9). Система передает эту запрограммированную величину из этой группы в BCD коде.

8.8 Вспомогательные М-функции групп M10, M11, M12, M13

Пользователем определены группы М-функций. Каждая определенная группа может содержать четыре М-функции. Коды М-функций вводятся как машинные константы.

8.9 Номер инструмента - функция Т

Функцией Т, которая является 8-ми декадной, вводится номер инструмента, который на основании передаваемой функции M06 включается в положение подходящее для обработки (станок содержит автоматическое изменение инструментов).

8.10 Вспомогательные функции Н, Р

Эти передаваемые функции являются двухдекадными, величины которых передаются в BCD коде и предназначены для управления манипулятором обрабатываемого продукта и инструмента, или для другого использования, требуемого пользователем.

9

9. ПАРАМЕТРИЧЕСКОЕ ПРОГРАММИРОВАНИЕ - ФУНКЦИЯ R

Система позволяет запрограммировать вместо конкретной величины любого адреса соответствующий параметр R. В момент обработки данных адреса, программируемых при помощи параметра R, будет вместо этого параметра установлена величина, которая была в последний раз в некотором предшествующем блоке партпрограммы установлена (декларирована). Параметрическое программирование обладает главным значением при использовании твердых циклов, макроциклов и подпрограмм, если обычно в программируемых блоках собственной партпрограммы величины отдельных параметров декларированы и с твердого цикла, партпрограммы или макроцикла вызываются. Однако не исключается возможность декларации и вызова параметров также в остальных блоках партпрограммы.

Следующее главное использование параметров это в случае арифметических операций. Эта проблематика описана в самостоятельном разделе.

Работа с параметрами разделяется на две фазы:

- а) декларация параметров
- б) вызов параметров

9.1 Декларация параметров

Во время декларации параметров устанавливаем для отдельных параметров конкретное значение. Декларация параметров в блоке партпрограммы имеет следующую форму:

N10 R0=aaaa R1=bbbb R2=cccc R95=xxxx

где aaaa,bbbb,cccc,.....,xxxx декларированные величины.

Заполнение параметров R величинами можно описать в блоке совместно с остальными адресами.

Пример:

N620 G54 G0 X0 Y0 R1=864 R2=-0.864 R3=100000 R20=250 R31=1

После выполнения этого блока будут в таблице параметров переписаны параметры № 1, 2, 3, 20 и 31 новыми величинами. Величины параметров не меняются. Отметим, что первый и второй параметр будут в таблицу записаны как 0.864 (второй со знаком минус), величина 100000 как 100.000, величина 250 как 0.250 и величина 1 как 0.001

Максимальное количество параметров составляет 96 (0 - 95). Максимальная величина параметра может находиться в пределах +69999.999 - -69999.999. Декларированная величина, однако, должна быть в соответствии с допустимой величиной адреса, в котором будет параметр вызван. Параметр 95 не меняется (он определен как управляющий), параметры 90 - 94 не рекомендовано использовать, потому что они зарезервированы для случайного расширения параметров управления.

ПРЕДУПРЕЖДЕНИЕ: в случае ввода партпрограммы не возможен контроль на допустимую величину

отдельных параметров и за правильную декларацию величин параметров несет полностью ответственность программист. Правильность параметров оценивается во время выполнения партпрограммы.

Пример:

N10 R10=123

N20 GR10

Эта запись бы во время выполнения партпрограммы сообщала ошибку „В двухдекадный адрес нельзя записать R в больше чем 2 декады“, потому что величина в G может быть макс. двухдекадная, но величина использованного параметра обладает тремя декадами (см. ниже).

9.2 Вызов параметров

Во время вызова параметров вместо этого параметра будет сначала установлена его декларированная величина. В партпрограмме вместо величины в соответствующем адресе будет указан адрес Rxx, где xx это номер параметра.

Пример:

Блок, в котором вызваны параметры, декларированные в предшествующем примере (блок N620):

N630 GR31 XR1 YR2 M03 SR20 TR3

Если бы не использовались параметры, будет блок записан следующим способом:

N630 G01 X0.864 Y-0.864 M03 S250 T100000

Величины всех адресов, кроме номера блока N, черты дроби (/), адреса R и функции G79, можно параметризовать.

Пример:

Партпрограмма с параметрическим программированием:

```
.  
. .  
. .  
N2 R30=350. R31=250. R29=1000  
N3 R1=100. R2 =200.  
N4 R3=15 M03 M42 M08  
N5 G90 G00 XR1 TR3  
N6 G82  
N7 G91 XR1 XR2  
N8 G90 G82  
N9 Z500. M05  
. .
```

На указанном примере выполнена, во-первых, декларация параметров R в самостоятельных блоках программы (N02, N03), а также в блоке совместно с программированием остальных адресов (N04). В блоках N5 и N7 выполнен вызов параметров.

9.3 Десятичная точка в декларации параметров

Во время декларации параметров важно выяснить влияние десятичной точки. Параметры, в общем, номера без размеров, размер приобретают после сопряжения с адресом. Параметр может быть записан с десятичной точкой, или без нее, и в обоих случаях будет иметь одинаковое значение, если они будут записаны правильно. Если параметр записан без десятичной точки, имеет в случае адресов с размерами

(напр. X,Y,Z, I,J) размер в микрометрах!

Разницу лучше объясним на примере:

```
N10 G0 X100
N20 R10=100
N30 R11=100.
N40 R12=0.100
N50 XR10 YR11 ZR12
```

В блоке N10 запрограммирован адрес X прямо на величину 100 (без дес. точки). Если речь идет о прямом программировании для адресов с размерами, не должна (но может) быть десятичная точка записана и размер будет в миллиметрах, т.е. 100мм. В случае параметрического программирования будет результат следующим:

В блоке N50 сдвинется X на величину 100 микронов (0.1мм), Y на величину 100мм и Z на 100микронов (0.1мм).

Во время записывания партпрограмм тогда рекомендовано записать декларацию параметров, определенных для адресов, которые обладают характером длины или расстояния с десятичной точкой, что намного разборчивее, чем запись без десятичной точки в микронах. Наоборот данные без размеров записываются (для разборчивости) без десятичной точки. Касается это напр. декларации параметра для оборотов S, G-функции, M функции, T-функции и т.п.

Пример:

```
N10 R20=120 R21=40 R22=12345678
N20 SR20 GR21 TR22
```

В блоке N20 будет выполнена передача 120 оборотов, отмена коррекции G40 и передача T12345678.

Одинаковое значение должна иметь также запись с десятичной точкой, записанная следующим способом, так как эта запись является для указанных функций менее разборчивой:

```
N10 R20=0.120 R21=0.040 R22=12345.678
N20 SR20 GR21 TR22
```

Во время программирования скорости сдвига (см. раздел 6) можно ввести F без точки, т.е. в мм/мин. или с точкой, т.е. в м/мин. Скорость F вводится параметрически и будет иметь следующий размер:

```
N10 R10=100
N20 R10=0.100
N30 R10=100.
N40 FR10
```

Правильно декларированные величины находятся в блоках N10 и N20. Можно сказать, что скорость в блок N10 записана в мм/мин., скорость в блок N20 записана в м/мин. (обе скорости являются одинаковыми).

Скорость, введенная в блок N30, записана в м/мин. и обозначает скорость 100м/мин. – т.е. скорость больше, чем система позволяет.

10

10. ВИНТОВАЯ ЛИНИЯ

Система управления CNC8x6 позволяет буксировку координат, т.е. управление одной координатой в зависимости от движения другой координаты. Это свойство использовано в случае винтовой линии, что является случаем буксировки координат в зависимости от движения по окружности.

Программирование винтовой линии позволено связанным движением в круговой и линейной интерполяции.

Введение винтовой линии будет запрограммировано введением подъема для координат, которые буксированы (ось, которая движется по линейной интерполяции), и функции G10. Под подъемом понимается расстояние, на которое должна сдвинуться буксированная координата, в течение всего времени всей окружности в круговой интерполяции управляющих координат. Таким образом, введенный подъем не зависит от радиуса круговой интерполяции (а даже на коррекции радиуса). Угол отклонения, однако, меняется в зависимости от радиуса круговой интерполяции.

Блок, в котором запрограммирован подъем, не движется и программируется обычно самостоятельно в любом блоке до собственной винтовой линии. Заданный подъем действителен постоянно до дальнейшего изменения. Например, в блоке N10 Z1 G10 запрограммирован подъем 1.05 мм и буксировочной осью является координата Z.

Программирование винтовой линии осуществляется функцией G09, отмена винтовой линии программируется функцией G98. Функция G98 должна быть запрограммирована в самостоятельном блоке. Во время программирования винтовой линии должно быть введено количество "витков". Количество витков программируется т.наз. скачком с повтором на одинаковый блок.

Пример:

```
N5 X0 Y0 Z0 G00
N10 Z1.05 G10          "введение подъема 1.05 мм
....
....
....
N50 G09 G02 X0 Y0 I100 G73 L50 Q20 "начало винтовой линии
N60 X200. Y0 I100
N70 G98               "конец винтовой линии
N80 X... Y... Z...    "движение всех координат
```

В блок N50 запрограммирована окружность с радиусом 100мм (исходная и конечная точка находятся на координатах 0,0). Окружность повторяется всего 21раз. (на 1 больше, чем запрограммированный под адресом Q). Функцией G73 программируется скачок на блок, программируемый в функции L. В блок N60 запрограммирован круговой сектор (полукруг) и в блок N70 запрограммировано отменение функции винтовой линии (G09).

11

11. АРИФМЕТИКА ПАРАМЕТРОВ

(Действительно от версии программного обеспечения пульта 30.19 - 14.2.2000 г.)

11.1 Общие принципы

Арифметику параметров выполняют функции **G26** **G27** **G28** и **G29**. В блоке, в котором запрограммирована одна из этих функций, должны быть запрограммированы соответствующие параметры управления R5 - R8.

Функция **G26** выполнит операцию по параметру управления **R5**.

Функция **G27** выполнит 2 операции за собой по параметрам управления **R5** и **R6**. (Сначала R5 и потом R6).

Функция **G28** выполнит 3 операции за собой по параметрам управления **R5** **R6** и **R7**. (Сначала R5 потом R6 и, наконец, R7).

Функция **G29** выполнит 4 операции за собой по параметрам управления **R5** **R6** **R7** и **R8**. (Сначала R5, потом R6, потом R7 и, наконец, R8).

Параметр **R95** определен для функции управления.

11.2 Управление двойной точностью

Некоторые операции могут быть введены в свои операнды как величины с двойной точностью – четыре слова. В этом случае в операнд будут записаны величины из двух по себе следующих параметров **ор1** и **ор1+1**, или **ор2** и **ор2+1**. Если разрешена двойная точность параметров, в параметре управления указывается номер параметра увеличенный на 50.

От программной версии пульта 30.08 (10.9.1999 г.) имеет система в распоряжении **96 параметров**. Поэтому мы должны в случае параметрических операций выбрать, если будем использовать двойную точность, и используем только 50 параметров или не будем использовать двойную точность, и используем всех 96 параметров.

1. Двойная точность запрещена:

Номер параметра: 0, 1, 2, ..., 95 Основная точность: 32 битов

2. Двойная точность разрешена:

Номер параметра: 0, 1, 2, ..., 49 Основная точность: 32 битов
50, 52, ..., 99 Двойная точность для параметров 0, 1, ..., 49: 64 битов

Управление двойной точностью параметров разрешается при помощи машинных констант или его можно использовать как динамическое управление при помощи параметра **R95**.

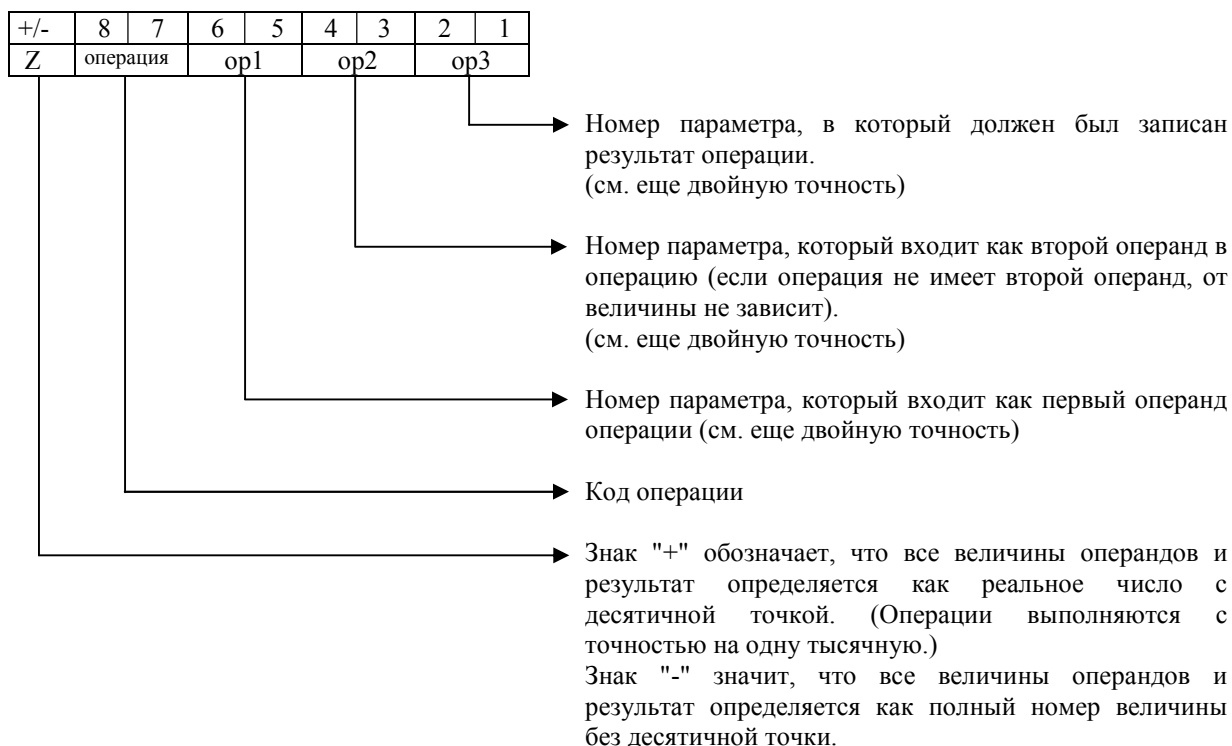
4 декада машинной константы 283: 0 арифметика параметров использует параметры 0-49 с

- возможностью двойной точности операндов
- 1 арифметика параметров использует параметры 0-95 без возможности двойной точности операндов
- 2 управление двойной точностью операндов определяет 1 декада параметра R95: 1.R95=0 диапазон 0-49, двойная точность разрешена 2.R85=1 диапазон 0-95, двойная точность запрещена

Если 4 декада машинной константы 283 установлена на величину 2, можно управлять активацией двойной точности в расчетах динамически. В случае установки 1 декады параметра R95 рекомендуем использовать инструкцию:

R5=48110095 G26 ;функция SET 1.R95 (установит 1 декаду параметра R95 на величину 1)
R5=48210095 G26 ;функция CLR 1.R95 (установит 1 декаду параметра R95 на величину 0)

11.3 Структура параметров управления



В следующем описании операций будут звездочкой обозначены операнды и результаты, которые могут иметь двойную точность.

ОПЕРАЦИЯ	КОД	ОПИСАНИЕ
Копирование $op1^* \rightarrow op2^*$	0	Скопирует величину параметра согласно op1* в параметр согласно ci1* . Величина параметра op1 останется без изменений. Величина op2 и Z не влияет.
Суммирование $op1^* + op2^* \rightarrow ci1^*$	1	Суммирование величины параметра согласно op1* с величиной параметра согласно op2* и запись результата в параметр согласно ci1* .
Вычитание $op1^* - op2^* \rightarrow ci1^*$	2	Вычитание величины параметра согласно op2* от величины параметра согласно op1* и запись результата в параметр согласно ci1* .
Умножение $op1^* \cdot op2^* \rightarrow ci1^*$	3	Умножение величины параметра согласно op1* на величину параметра согласно op2* и запись результата в параметр согласно ci1* .
Разделение $op1^* / op2^* \rightarrow ci1^*$	4	Разделение величины параметра согласно op1* на величину параметра согласно op2* и запись результата в параметр согласно ci1* .
Возведение во вторую степень $op1^2 \rightarrow ci1^*$	5	Возведение во вторую степень величины параметра согласно op1 и уложение результата в параметр согласно ci1* .
Второй корень $\sqrt{(op1^*)} \rightarrow ci1^*$	6	Квадратный корень величины параметра согласно op1* и запись результата в параметр согласно ci1* .
Абсолютная величина $ op1^* \rightarrow ci1^*$	7	Абсолютная величина параметра согласно op1* и запись результата в параметр согласно ci1* .
Инверсная величина $-op1^* \rightarrow ci1^*$	8	Инверсная величина параметра согласно op1* и запись результата в параметр согласно ci1* .
Бинарный сдвиг $op1^* \cdot 2^{op2} \rightarrow ci1^*$	9	Бинарный сдвиг величины параметра согласно op1 . Сдвиг будет выполнен согласно параметру op2 . Величина op2 может быть ± 32 (± 64 для двойной точности)
Обратная величина $1/op1^* \rightarrow ci1^*$	10	Обратная величина параметра согласно op1* и запись результата в параметр согласно ci1* .
Функция 2^X $2^{op1} \rightarrow ci1^*$	11	Функция 2^X согласно op1 и запись результата в ci1 (цель). Операнд op1 не может иметь двойную точность и должен быть реальным числом. Результатом является реальное число и может иметь двойную точность.
Функция $\log_2 x$ $\log_2(op1) \rightarrow ci1^*$	12	Функция $\log_2 x$ согласно op1 и запись результата в ci1 . Операнд op1 может иметь двойную точность и должен быть реальное число. Результат является реальным числом и не может иметь двойную точность.
Функция x^y $(op1)^{op2} \rightarrow ci1^*$	13	Возведение величины согласно op1 в величину согласно размеру в op2 и запись результата в ci1 . Операнды op1 и op2 не могут иметь двойную точность и должны быть реальным числом. Результатом является реальное число и может иметь двойную точность.
Функция $\sqrt{(x^2 + y^2)}$ $\sqrt{(op1^2 + op2^2)} \rightarrow ci1^*$	14	Функцию $\sqrt{(op1^2 + op2^2)}$ можно использовать для реальной и целой величины.
Больше $op1^* > op2^* \rightarrow ci1^*$	15	Если величина параметра согласно op1 больше, чем величина параметра согласно op2 , будет записана в параметре согласно ci1* величина 73 , в противном случае 78 .
Больше или равно $op1^* \geq op2^* \rightarrow ci1^*$	16	Если будет величина параметра согласно op1 больше или равна величине согласно op2 , будет записана в

		параметре согласно cil* величина 73 , в противоположном случае 78 .
Меньше $op1^* < op2^* \rightarrow cil^*$	17	Если величина параметра согласно op1 меньше чем величина согласно op2 , будет в параметр согласно cil* уложена величина 73 , в противоположном случае 78 .
Меньше или равно $op1^* \leq op2^* \rightarrow cil^*$	18	Если величина параметра согласно op1 меньше или равна величине согласно op2 , будет в параметр согласно cil* записана величина 73 , в противоположном случае 78 .
Сравнение $op1^* = op2^* \rightarrow cil^*$	19	Если величина параметра согласно op1 равна величине согласно op2 , будет в параметр согласно cil* записана величина 73 , в противоположном случае 78 .
Условие "равно" $op1^* = op2^* \rightarrow \text{продолжай}$	20	Если величина параметра согласно op1 равна величине согласно op2 , продолжается расчет следующей параметрической операции.
Условие "неравно" $op1^* \neq op2^* \rightarrow \text{продолжай}$	21	Если величина параметра согласно op1 не равна величине согласно op2 , продолжается расчет следующей параметрической операции.
Конец условия $\rightarrow \text{продолжай}$	22	Окончание условий для расчета параметрических операций.
Перевод int $\rightarrow$ real $real(op1) \rightarrow cil$	23	Перевод величин целых чисел согласно параметру op1 на реальную величину согласно cil
Перевод real $\rightarrow$ int $int(op1) \rightarrow cil$	24	Перевод реальной величины согласно op1 на целые числа величин согласно cil
Перемещение измерение $ODM \rightarrow cil^*$	25	Перемещение измерением координат согласно op1* (и op2*) и запись в параметр согласно cil* . В операнды op1 (op2) вводится порядковый номер координат (1,2,...,6).
Перемещение программированной величины координат $PROG \rightarrow cil^*$	26	Перемещение программированной координаты согласно величине координат op1* (и op2*) и запись в параметр согласно cil* . В операнд op1 (op2) вводится требуемый номер координат (1,2,...,6).
Функция $\tan(x)$ $\tan(op1) \rightarrow cil^*$	27	Функция $\tan(x)$ согласно op1 и запись результата в cil . Операнд op1 не может иметь двойную точность и должен принимать реальное значение. Результат является реальным номером и может иметь двойную точность.
Сумма геометрической последовательности $1 + k + k^2 + \dots + k^{n-1}$ $(k^n + 1) / (k - 1) \rightarrow cil^*$ $k...op1^*$ $n...op2^*$	28	Сумма геометрической последовательности $1+k+k^2+\dots+k^{n-1} = (k^n-1)/(k-1)$ Частное k определено параметром op1 . Количество членов n определено параметром op2 . Операнды op1 и op2 не могут иметь двойную точность и должны быть реальными числами. Результат принимает реальное значение и может иметь двойную точность.
Условие "больше" $op1^* > op2^* \rightarrow \text{продолжай}$	29	Если величина параметра согласно op1 больше, чем величина согласно op2 , продолжается расчет последующей параметрической операции.
Условие "больше или равно" $op1^* \geq op2^* \rightarrow \text{продолжай}$	30	Если величина параметра согласно op1 больше или равна величине согласно op2 , продолжается расчет последующей параметрической операции.
Условие "меньше" $op1^* < op2^* \rightarrow \text{продолжай}$	31	Если величина параметра согласно op1 меньше величины согласно op2 , продолжается расчет последующей параметрической операции.
Условие "меньше или равно" $op1^* \leq op2^* \rightarrow \text{продолжай}$	32	Если величина параметра согласно op1 меньше или равна величине согласно op2 , продолжается расчет последующей параметрической операции.
ПРОГРАММА запуск программы от op1 по cil	33	Запуск программы от параметра op1 до параметра cil

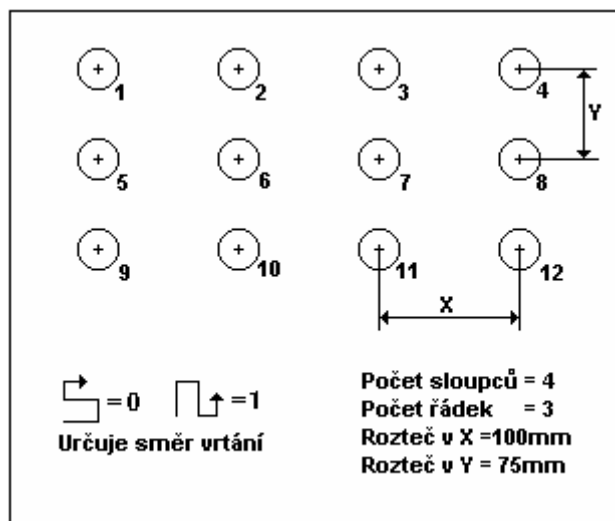
		(включительно). Параметры могут быть заполнены в одинаковом блоке. За функцией 33 не должна быть запрограммирована следующая параметрическая операция.
Перемещение параметров перемещение от op1 до op2 до cil	34	Перемещение параметров от op1 (включительно) до op2 (включительно) до области согласно параметру cil Для op1=99 аннулируется область от cil по op2
Смена параметров смена из op1 по op2 в cil	35	Смена параметров из op1 (включительно) по op2 (включительно) в область согласно параметру cil
Вписанная дуга	36	Функция использует большую область параметров
$\sin(\text{op1}) \rightarrow \text{cil}$	37	Функция $\sin(x)$ согласно op1 и запись результата умноженного на *10000000 в cil .
$\cos(\text{op1}) \rightarrow \text{cil}$	38	Функция $\cos(x)$ согласно op1 и запись результата умноженного на *10000000 в cil .
$\arcsin(\text{op1/op2})$	39	Функция $\arccos(x/y)$ согласно op1/op2 и запись результата в cil .
$R \cdot \cos((I \cdot B/N) + A)$	40	Функция для расчета дыр на окружности
$\text{op1} + 1 \rightarrow \text{cil}$	41	для op2=0 инкрементация величины op1 и запись в cil для op2 < 0 больше величины согласно op1 на номер в op2 и запишется в cil
$\text{op1} - 1 \rightarrow \text{cil}$	42	для op2=0 декрементация величины op1 и запись в cil для op2 < 0 вычитается величина согласно op1 от номера в op2 и запишется в cil
SAVE	44	Сохранение всех параметров
RESTORE	45	Возобновление параметра от op1 по op2 включительно в cil Для op1=op2=cil=0 возобновление всех 96 параметров
LICHY(op1) -> cil (НЕЧЕТНОЕ)	46	Если величина параметра согласно op1 нечетная, запишется в параметр согласно cil величина 73 , в противоположном случае 78 .
SUDY(op1) -> cil (ЧЕТНОЕ)	47	Если величина параметра согласно op1 четная, запишется в параметр согласно cil величина 73 , в противоположном случае 78 .
LOGIC(dek) -> dek(cil)	48	Логические операции для декады параметров 8. и 7. декада является кодом для логики = 48 6. декада – это код операции: 1 = SET, 2=CLR, 3=NOT, 4 = OR, 5= AND, 6= XOR, 7=TEST 5. декада определяет порядковый номер декады для логической операции 1,2,...,8 4. и 3. декада определяет 1. операнд для операции OR, AND, XOR и TEST 2. и 1. декада определяют 2. операнд и одновременно cil для операций OR, AND, XOR. Операции SET, CLR и NOT работают только с параметром согласно cil . Для операции TEST в параметре согласно cil будет записана величина 73 или 78 .
READ(TAB) -> cil	49	Перемещение из таблиц в параметры 8. и 7. декада – это код для перемещения = 49 6. декада – это код таблицы: 1=таб. коррекций, 2=таб. сдвига, 3=таб. машинных констант, 4=регистр блока BWRK, 5=регистр блока AWRK 5. декада определяет столбец в отдельных таблицах 1,2... для регистров блока это: 1=функция M, 2=функция G 3=функция P, 4= H, 5= S, 6= T, 7= Q, 8=D 4. и 3. декада определяют источник в таблицах. Для таб. коррекций это 1-99, для таб. сдвига это 53-59, для машинных констант это 0-99 + столбец*100, для регистров блока вводится группа функций M и G

		2. и 1. декада определяют параметр цели для записи величины из таблицы
WRITE(op) -> TAB	50	Перемещение из параметров в таблицы 8. и 7. декада – это код для перемещения = 50 6. декада это код таблицы: 1=таб. коррекций, 2=таб. сдвига 5. декада определяет столбец в отдельных таблицах 1,2... 4. и 3. декада определяют параметр источника для записи величины в таблицу 2. и 1. декада определяют цель в таблицах. Для таб. коррекций это 1-99, для таб. сдвига это 53-59
MESSAGE	51	Изображение технолог. информационного сообщения 8. и 7. декада – это код для сообщений = 51 6. и 5. декада – это код сообщения 1,2,...,99 4. и 3. декада – это номер параметра, который может быть изображен в сообщении (в тексте – это знак самого \$). Величина изображенного параметра прочитана в момент выполнения этой инструкции и во время изображения она не актуализирована. (Кроме этого параметра может быть изображен также параметр текущим способом актуализированный – в тексте необходимо указать его номер \$xy) 2. декада – это время изображения (если это требуется) в секундах 1,2,...,9 1. декада – это управляющий способ изображения: 0= отмена сообщения 1= постоянное сообщение (если не будет отменено) 2= сообщение изображено в течение времени действия одного блока 3= сообщение изображено в течение определенного времени изображения, определенного во 2. декаде

11.4 Пример использования арифметики параметров

Для объяснений использования арифметики параметров используем стандартно поставляемый макроцикл L9001.NCP для сверления дыр на регулярном расстоянии в столбцах. В системах CNC8x6 в настоящее время можно использовать 51 математическую операцию. В примере нельзя подробно все описать, но, не смотря на это, принцип использования является одинаковым, и пример точно послужит как проект для создания собственных программ с использованием арифметики параметров.

Макроцикл для сверления дыр использует эти вступительные параметры:



R09 = номер твердого цикла 81 - 89
 R10 = Расстояние дыр в оси X (= расстояние столбцов)
 R11 = Расстояние дыр в оси Y (= расстояние строки)
 R12 = Количество столбцов
 R13 = Количество строк
 R14 =
 0 ... сверление по строкам (начало в оси X)
 1 ... сверление по столбцам (начало в оси Y)

Количество столбцов = 4
 Количество строк = 3
 Расстояние по X = 100 мм
 Расстояние по Y = 75 мм

Определяет направление сверления

Макроцикл предполагает исходную точку над первой дырой, т.е. программист должен обеспечить в партпрограмме установку над первой дырой.

Отметим еще, что знак в параметрах R10 и R11 определяет направление движения от первой дырки, поэтому может начинаться в любом угле задуманного квадрата или прямоугольника. Макроцикл возвращается после окончания опять в исходную точку.

Далее предполагаем, что минимальное количество введенных строк составляет 2, а также минимальное количество введенных столбцов составляет 2.

Для нашего примера (см. рис) будут параметры установлены следующим способом:

R09 = 81 „твердый цикл G81
R10 = 100.0
R11 = 75.0
R12 = 4 „количество столбцов
R13 = 3 „количество строк
R14 = 0 „сверление по строкам (см. номера дыр)

Предполагаем систему координат X,Y, положительное направление направо и вверх.

Пометка:

Если посмотрим на параметры напр. по выбору индикации, будут параметры, введенные без точки, аннулированы в форме с десятичным знаком, следующим способом:

R09=0.081, R12=0.004, R13=0.003

Величина, однако, одинаковая – но, не смотря на это с этими числами должны в некоторых арифметических операциях работать как с реальными числами (до кода операции этот знак минус).

В следующем тексте сделаем подробный комментарий к арифметическим операциям.

N1 G79 L9001

В блок N10 будут декларированы различные вспомогательные параметры, которые будут необходимы для дальнейших расчетов, здесь декларируются также различные константы, необходимые для дальнейших расчетов.

N10 **R15=0** " Вспомогательный рабочий параметр
 R16=0 " Вспомогательный рабочий параметр
 R17=0 " Вспомогательная декларация константы 0
 R18=31 " Вспомогательный параметр для расчета ротации
 R19=0 " Вспомогательный рабочий параметр
 R20=0 " Счетчик (рабочий параметр)

В блоке N20 будет принято решение по входному параметру R14, если начнется сверление по строкам или по столбцам, т.е. в которой оси начнется первое движение. Будет использован код операции 19 – сравнение. Сравнивается входной параметр R14 (op1) с константой 0, которую мы декларировали в параметр R17 (op2). Результат арифметической операции сравнения будет записан в параметр R06 (cil). В случае равенства будет записана в R06 величина 73, в случае не равенства величина 78.

N20 G26 R5=19141706 " если R14=0 потом сверление происходит по строкам (X)

В блоке N30 будет заменен формальный параметр R06 действительной величиной, рассчитанной в предшествующем блоке, т.е. будет выполнено G73 L1105 или G78 L1105. Функция G73 выполнит шаг на блок N1105, функция G78 не сделает ничего, программа будет продолжаться на следующей строке. В нашем примере будет выполнен шаг на блок N1105.

N30 GR06 L1105 " G73 L1105 (= X) или G78 L1105 (= Y)
 СВЕРЛЕНИЕ ПО СТОЛБЦАМ (Y)

N105

.....

.....

.....

N900 G70 „Конец сверления по столбцам

N1160 G27 R5=08100010
R6=-41200020
G73 L1110 QR13 G80

"шаг на следующую строку

Следующие блоки определены только для расчета возврата в исходное положение первой дыры.

Если количество строк LICHÝ (НЕЧЕТНОЕ) – возвращается в осях X и Y.

Если количество строк SUDÝ (ЧЕТНОЕ) – возвращается только в оси Y.

Количество строк находится в параметре в R19.

Для принятия решения, если количество строк четное или нечетное использует код операцию 47, которая в параметре R17 установит величину 73 (шаг), если количество строк четное или 78 (продолжается), если количество строк нечетное. В нашем примере количество строк нечетное (3) и должен с дыры 12 вернуться на 1 по обеим осям. Продолжаем тогда в блоке N1185.

N1170 G26 R5=47190017 G80

**" если (R19 = четное) потом R17=73 в
 противоположном случае R17=78**

N1185 GR17 L1200 R15=0

" шаг на L1200 в случае четного количества строк

В блоке N1190 будет рассчитан возврат в оси X, т.е. перепишет ноль в R15 рассчитанную величину. Будет использован код умножения (опять как реальный номер – знак минус).

N1190 G28 R5=-41120012

**" Из количества повторов будет рассчитано
 количество расстояния столбцов**

R6=-03121015

**" R12*R10=R15 (количество расстояния)*(расстояние
 X) = траектория**

R7=08150015

" Изменение знака траектории

Подробным способом будет рассчитан возврат в оси Y.

$R13 \cdot R11 = R14$... (количество строк-1)*(расстояние Y) = прирост возврата в Y.

N1200 G27 R5=-03131114 R6=08140014

Рассчитанные траектории записаны в R14 и R15.

N1210 YR14 XR15

" возврат в исходную точку

В блок N1230 вернуться назад параметры, которые хранились в блоке N1105. Если op1, op2 и cil являются 0, будет выполнен возврат всех 96 параметров. Параметры op1, op2 и cil (целью) можно управлять, которые параметры (и куда) должны вернуться.

N1230 G90 G26 R05=45000000
N1300 G70

**" Вернуться назад сохраненные параметры
 " конец макроцикла**

*

12

12. ПРОГРАММИРОВАНИЕ И ОПРЕДЕЛЕНИЕ ТВЕРДЫХ ЦИКЛОВ

12.1 Создание твердых циклов

Твердые циклы – это норма, рекомендованная макроциклами, которые выполняют стандартные циклы для сверления, фрезеровки или токарных работ. На прохождение твердого цикла влияет только установка параметров, согласно которым твердый цикл выполняет свою деятельность. Принципы для создания твердых циклов одинаковы как для фрезеровки, так и для токарных станков.

Твердые циклы в запоминающем устройстве системы CNC8x6 записываются как нормальные партпрограммы и можно их создавать и редактировать. В резервном запоминающем устройстве (на диске) может находиться также больше файлов с твердыми циклами, но система во время выбора партпрограммы прочитает тот файл, имя которого указано в параметре **\$17 конфигурационного файла CNC836.KNF**. Следующим условием автоматического чтения твердых циклов является установка **3. декады машинной константы 99 на величину 1**. (см. Инструкцию по обслуживанию, разделы: "Машинные константы" и "Конфигурация").

Создание и редактирование твердых циклов можно выполнять из системы CNC8x6. Рекомендуется использовать редактор системы (вход из основного меню через кнопки Система и Система-редактирование), или во время редактирования и записи файла с твердыми циклами в среде партпрограмм (вход из основного меню через кнопку Программы и Редактирование) можно – если был после включения системы выполнен выбор любой партпрограммы – сообщить ошибку «Партпрограмма уже находится в запоминающем устройстве», потому что твердый цикл вводится в запоминающее устройство автоматически во время выбора программы. Во время редактирования твердых циклов через редактор системы не будет после записи выполнен синтаксический контроль. Случайные ошибки поэтому проявятся уже во время выбора программы.

Файл с твердыми циклами начинается с ключевого слова: " \$PC и заканчивается знаком *.

Во время создания твердых циклов часто используют арифметику параметров. Параметрические операции подробно объяснены в разделе «Арифметика параметров».

Для программирования твердых циклов действительны следующие правила:

Движения, включенные в твердые циклы, выполняются в блоке, где в первый раз запрограммирована одна из функций G81 – G89 за интерполяциями, но до заключительных M-функций. Цикл сверления будет потом выполнен в каждом следующем блоке (функции G81-G89 уже не должны быть запрограммированы) до блока, в котором запрограммирована функция отмены твердого цикла G80. В блоке с G80 уже цикл сверления не выполняется.

Твердые циклы не меняют запрограммированные технологические M-функции, G-функции, и величины остальных адресов (напр. номер таблицы коррекций D, скорость F и т.п.) и параметров, или после выполнения твердых циклов вернут величины функций, адресов и параметров, которые были запрограммированы до отмены твердого цикла, хотя их твердый цикл поменяет.

Например, в твердом цикле G84 использован реверс шпинделя. До вызова твердого цикла будет запрограммировано напр. M3, твердый цикл после перемещения на дно дыры вернет шпиндель, т.е. передаст M4, но после окончания твердого цикла (точнее после старта блока следующего после твердого цикла) опять начнет вращаться шпиндель в направлении M3. Аналогично, если в твердом цикле напр. поменяет скорость F, после окончания твердого цикла будет скорость установлена на F, которое было запрограммировано до вызова твердого цикла.

Пометка:

Это есть главная и практически единственная разница между твердыми циклами и макроциклами. После выполнения макроциклов останутся величины G и M функций и адресов такими, как их случайно установит или поменяет макроцикл.

Пример:

```
N10 ...
N20 X50 G0
N30 R26=400 R27=200 R30=5.0 R31=-85.0 R32=25.0 „декларация параметров для твердого
цикла
N40 M3 M41
N50 X100 G00 G81
N60 X200
N70 X300
N80 X400
N90 X500
N100 X600 G80
```

Твердый цикл сверления G81 будет впервые выполнен в конце блока N50 (после перемещения на расстояние X100) и потом в каждом следующем блоке. Последний цикл сверления будет выполнен в блоке N90 после перемещения на расстояние X500, потому что в блоке N100 уже запрограммирована отмена твердого цикла функцией G80. Обратите внимание на размещение функции вызова твердого цикла (здесь G81). Он не может находиться уже напр. в блоке N30 (выполнение параметров) или N40 (включение M3). Если бы была функция G81 напр. в блоке N30, будет выполнено сверление уже в этом блоке (на в последний раз установленном расстоянии X50) и следующее сверление в одинаковом положении X в блоке N40.

12.2 Примеры твердых циклов сверления для фрезеровки

Твердые циклы для фрезеровки поставляются производителем, но, не смотря на это, пользователь их может обработать, или создать следующие собственные. С выгодой можно использовать арифметику параметров, описанную в самостоятельном разделе.

В системах CNC8x6 (от версии 30.29) поставляются стандартные твердые циклы в файле **PCYKLYDG.NCP**. Старшая версия использует файл PEVNECYK.NCP или другой обработанный. Файл. Какие твердые циклы будет система использовать определено установкой параметра **\$17 в файле CNC836.KNF**, где введено **название файла с твердыми циклами**.

Если используется диалоговое создание партпрограммы и в нем выбор твердых циклов, рекомендовано использовать PCYKLYDG.NCP.

Описание твердых циклов из файла PCYKLYDG.NCP

Твердый цикл начинается в референционной плоскости RA. Глубина сверления выполняется в плоскости RB. Твердый цикл заканчивается в плоскости RC. Координаты референционных плоскостей RA, RB и RC должны быть запрограммированы в **абсолютных координатах**. Прирост глубины q для сверления глубоких дыр запрограммирован с приростом. Собственная программа выполнения твердых циклов в системе создана в параметрической форме. В партпрограмме, с которой соответствующий твердый цикл вызывается, должны быть декларированы координаты плоскостей, прирост для сверления глубоких дыр, временная задержка, размер сдвига и оборотов отдельными параметрами следующим способом:

Параметр	Описание
R26	Декларирует скорость сдвига (адрес F), который в рамках компьютера выполняется.
R27	Декларирует размер оборотов (адрес S) для данного компьютера (в соответствии с раньше заданной функцией M41 - M44)
R28	Декларирует прирост q для сверления с ополаскиванием (G83) и сверление с отламыванием стружек (G86)
R29	Декларирует размер временной задержки (функция Q)
R30	Декларирует координаты плоскости RA („откуда сверлить“)
R31	Декларирует координаты плоскости RB („куда сверлить“)
R32	Декларирует координаты плоскости RC („куда выехать“)

Рекомендации для введения параметров:

Во время введения параметров важно обратить внимание на влияние десятичного знака на величину параметра. Величина параметра, в общем, является номером без единицы. Размер приобретает после соединения с конкретным адресом. Если не указан десятичный знак, приобретает параметр Rxx=200 величину 200. Если будет величина 200 соединена с параметром, который определяет обороты, будет размер 200об./мин. Если будет величина 200 соединена с параметром, который определяет скорость сдвига, будет размер 200мм/мин., если будет величина 200 соединена с параметром, который определяет размер длины или положение, будет размер 200 микронов (внимание – ни в коем случае миллиметров!).

Запись самого десятичного знака можно себе представить как дополнение трех нулей к величине параметру. Rxx=200 бы, таким образом, обозначало 200000 оборотов, 200000 мм/мин. (=200 м/мин.) и 200000 микрометров (=200 мм).

С практической точки зрения тогда рекомендуется записать величину параметра для оборотов без десятичного знака, величину параметра для скорости или без этого знака, если хотим ввести величину в мм/мин. или с этим знаком, если хотим ввести величину в м/мин. и величину параметра для длины и положения с десятичным знаком, если более подходящим является вводить размеры в мм, чем в микронах.

Возможности введения параметров для твердых циклов приведены в следующей таблице.

Параметрический ввод	Возможности записи величины в параметр	Размер введенной величины	Пометка
FR26 (скорость)	R26=200	200 мм/мин.	Для миллиметрового сдвига (G94)
	R26=0.200	0,2 м/мин. (=200 мм/мин.)	Для миллиметрового сдвига (G94)
	R26=200	200 микронов/об.	Для сдвига оборотов (G95)
	R26=0.200	0,2 мм/об.	Для сдвига оборотов (G95)
SR27 (обороты)	R27=300	300 об./мин.	
	R27=0.300	300 об./мин.	
ZR28 (прирост)	R28=25.	25 мм	
	R28=25.0	25 мм	
	R28=25000	25000 микронов (=25 мм)	
QR29 (врем. длительность)	R29=250	250 * 10 мс = 2,5 с	Одна единица = 10 мс
	R29=0.250	250 * 10 мс = 2,5 с	
ZR30 (Плоскость)	R30=5. 5	5,5 мм	
	R30=5500	5500 микронов (=5,5 мм)	
ZR31 (Плоскость)	дт-то		
ZR32 (Плоскость)	дт-то		

Пометка:

Диалоговое введение твердых циклов устанавливает форматы ввода всех параметров в форму с десятичным знаком. Более подробно см. в разделе Создание диалога.

Пример использования твердого цикла:

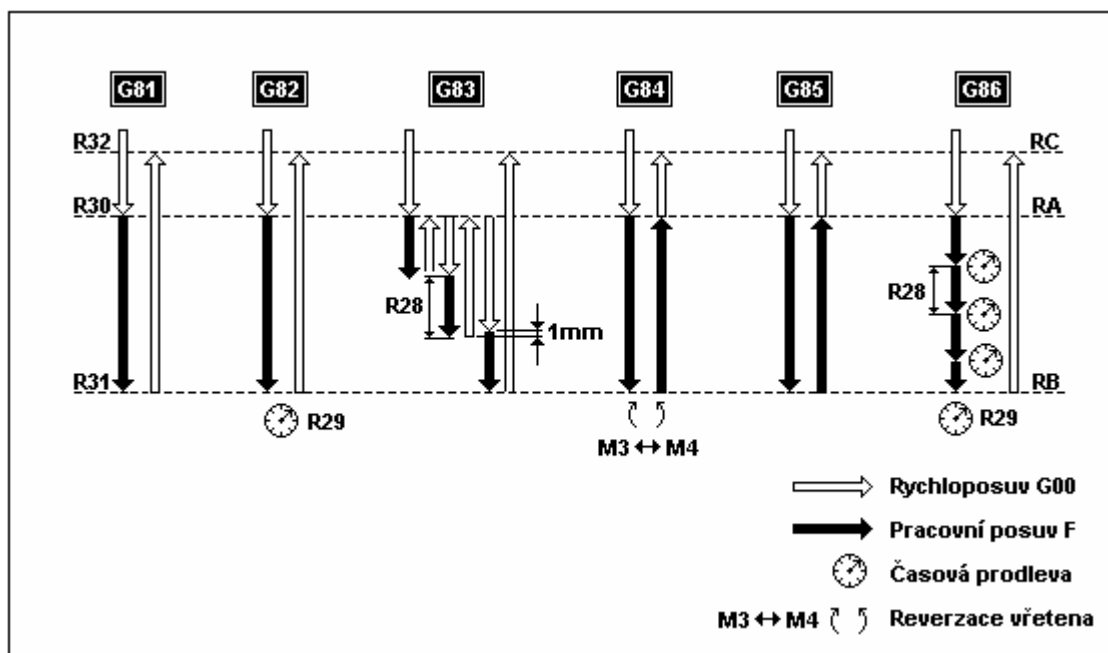
(для твердого цикла G82 вводится сдвиг 460 мм/мин., 1000 об./мин., временная задержка 3 с. Сверление начинается с положения 5.0 мм до положения – 50.0 мм и после окончания выедет на расстояние 30.0. Всего сверлятся 3 дыры. Первая дыра в блоке N4 в положении X100. Это положение было установлено уже в блоке N1. Вторая дыра сверлится в положении X200 (перемещение в блоке N5 запрограммировано приростом G91 на 100 мм). Третья дыра сверлится в положении X300 (опять приростом на 100 далее). Программирование прироста G91 относится только к перемещению. В твердый цикл вводятся параметры плоскостей всегда абсолютно! До твердого цикла вводится S100 об./мин. Твердый цикл меняет обороты параметром R27) на 1000 об./мин. После окончания твердого цикла обороты возвращаются на S100 об./мин., т.е. перемещение между отдельными циклами сверления в блоках N5 и N6 будет выполнено при 100 об./мин.

```

N1    G00    G90    X100.
N2    Z50.    S100    M43    M04
N3    R26=460 R27=1000 R29=300 R30=5. R31=-50. R32=30.
N4    G82
N5    G91    X100.
N6    X100.
N7    G90    Z500.    M05    G80

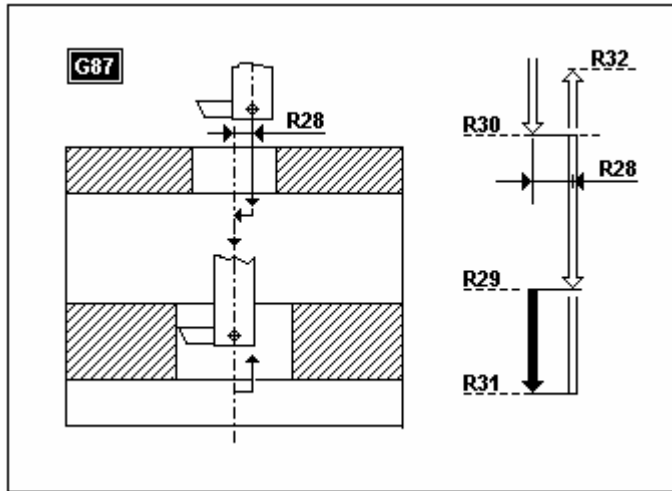
```

Твердые циклы - файл PCYKLYDG.NCP



Быстрый сдвиг G00
 Рабочий сдвиг F
 Временная задержка
 Возвращение шпинделя

Твердый цикл G87 с „отскоком“ в оси Y



Извлечение твердых циклов сверления из файла PCYKLYDG.NCP, который использует также диалоговое создание партпрограмм.

Пометка 1: В диалоговое создание партпрограмм не включен твердый цикл G87.

Пометка 2: В системах может находиться актуализированное состояние файла с твердыми циклами.

```
" $PC - ТВЕРДЫЕ ЦИКЛЫ СВЕРЛЕНИЯ ФРЕЗОЙ В ОСИ Z
" (Совместимая с диалоговой графикой от версии пульта 30.29)
" Дата ревизии: 20.2.2001 г.
"
" Пометка 1:
" В твердом цикле G84 используем для резки резьбы резьбовой
" головкой первоначально G94 (миллиметровый сдвиг)
" Если хотим использовать сдвиг оборотов, поменяем G94 на G95
"
" Пометка 2:
" Если хотим ввести скорость сдвига F и обороты S
" параметрически (R26 и R27) до вызова твердого цикла,
" отменим во всех твердых циклах FR26 и SR27 !
" Скорость F и обороты S потом вводятся до вызова цикла.
"
" Используемые параметры:
"
" R26 = Скорость сверления [мм/мин.]
" R27 = Обороты [об./мин.]
" R28 = Прирост сверления (или отскок в Y для G87)
" R29 = Временная задержка (1=10 мс) (или промежуточная плоскость RD для
G87)
" R30 = Плоскость RA (абсолютное положение откуда сверлить)
" R31 = Плоскость RB (абсолютное положение куда сверлить)
" R32 = Плоскость RC (абсолютное положение куда выехать после окончания
PC)
"
"-----
" Сверление без временной задержки
"-----
N1 G79 L81
N2 G00 G90 ZR30 FR26 SR27
N3 G01 ZR31
```

```

N4    G00 G70 ZR32
"-----
" Сверление с временной задержкой в конце дыры
"-----
N1    G79 L82
N2    G00 G90 ZR30 FR26 SR27
N3    G01      ZR31 G04 Q929
N4    G00      ZR32
N5    G70
"-----
" Сверление с ополаскиванием
"-----
N1    G79 L83
N10   FR26 SR27
      G27 R5=15313024      "if R31 > R30 then R24=73
      R6=07280028        "абс. величина прироста сверления
N20   GR24 L240
" Сверление в направлении минус
N40   G27 R5=00300029
      R6=00300025
      R24=1.000
N50   G26 R5=16312923      " if R31 >= R29 then R23 = 73 else 78
N60   GR23 L140
N70   ZR25 G00 G26 R5=02292829 " R29=R29-R28
N80   G26 R5=15293123      " if R31 > R29 then R23 = 73 else 78
N90   GR23 L110 Q9998      " Отскочит блок 100
N100  G27 R5=00310029      " R29=R31
      R6=00320030        " последнее сверл. - выедет по R32
N110  ZR29 G01
N120  ZR30 G00
N130  G26 R5=01292425 G73 L50 Q9998 " R25=R29+R24 (R24 = 1 мм)
N140  G70
" Сверление в направлении плюс
N240  G27 R5=00300029 R6=00300025 R24=1.000 "R30 -> R29
N250  G26 R5=18312923      " if R31 <= R29 then R23 = 73 else 78
N260  GR23 L340
N270  ZR25 G00 G26 R5=01292829 " R29 = R29 + R28
N280  G26 R5=17293123      " if R31 < R29 then R23 = 73 else 78
N290  GR23 L310 Q9998      " Отскочит блок 300
N300  G27 R5=00310029      " R29 = R31
      R6=00320030        " последнее сверл.- выедет по R32
N310  ZR29 G01
N320  ZR30 G00
N330  G26 R5=02292425 G73 L250 Q9998 " R25 = R29 - R24 (R24 = 1 мм)
N340  G70
"-----
" Сверление с возвращением шпинделя (резьбовой головкой)
"-----
N1    G79 L84
N2    G00 G90 ZR30 FR26 SR27
      R08=4              " Вспомогательное для сравнения на M4
      R09=3              " Вспомогательное для сравнения на M3
      G26 R05=33100016    " Программа 10 - 16
      R10=49510220        " Запишет 2. группы M-fci в R20
      R11=20200900        " if R20=R09 (R20=M3?) then продолжать
      R12=00080007        " В R7 da M4
      R13=22000000        " Конец условия
      R14=20200800        " if R20=R08 (R20=M4?) then продолжай
      R15=00090007        " В R7 da M3
      R16=22000000        " Конец условия
N3    G01      ZR31 G94 M49 " G95 для сдвига оборотов

```

```

N4   MR07      ZR30          " возврат шпинделя (R07)
N5   G70 G00 ZR32
"-----
" Сверление и выезд рабочим сдвигом
"-----
N1   G79 L85
N2   G00 G90 ZR30 FR26 SR27
N3   G01       ZR31
N4       ZR30
N5   G70 G00 ZR32
"-----
" Сверление с отламливанием стружек с временной задержкой
"-----
N1   G79 L86
" Принимает решение о направлении сверления - if R30 > R31 - сверление в
минус
"                               - if R30 < R31 - сверление в плюс
" Прирост сверления вводится всегда положительно
"
N3   FR26 SR27
      G27 R5=07280028      "абс. величина прироста сверления
      R6=16313034          "if R31<R30 then скок
N4   GR34 L100
" направление в минус
N10  G00 G90 ZR30          "установит быстрым сдвигом плоскость RA
      G26 R5=00300024      "R30 -> R24
N20  G27 R5=02242824      "R24=R24-R28 (в цикле вычитает прирост)
      R6=15312423          "if R31>R24 then R23=73 else R23=78
N30  GR23 L60              "скок G73 или пустая инструкция G78
N40  G01 ZR24 G04 QR29     "сверлит по R24, потом временная задержка
N50  G73 L20 Q9998         "скок на блок N20
N60  ZR31
N70  G70 G00 ZR32
" Направление в плюс
"
N100 G00 G90 ZR30          "установит быстрым сдвигом плоскость RA
      G26 R5=00300024      "R30 -> R24
N200 G27 R5=01242824      "R24=R24+R28 (в цикле прибавляет прирост)
      R6=17312423          "if R31<R24 then R23=73 else R23=78
N300 GR23 L60              "скок G73 или пустая инструкция G78
N400 G01 ZR24 G04 QR29     "сверлит по R24, потом временная задержка
N500 G73 L200 Q9998        "скок на блок N200
N600 ZR31
N700 G70 G00 ZR32
"
"-----
" Сверление с отскоком в оси Y
"-----
" R28 = отскок в оси Y
" R29 = промежуточная плоскость RD
" остальные параметры одинаковы с остальными PC
"
N1   G79 L87
N2   G26 R5=49510224      "R24=запишет направление вращения шпинделя
      G00 G90 ZR30          "Быстрым сдвигом на плоскость RA
N3   G26 R5=08280025      "R25=инверсная величина отскока R28
      G91 YR28              "Отскок в оси Y вне оси сверления
      M19                  "Ориентированный стоп
N4   G90 ZR29              "Абсолютно на промежуточную плоскость RD
N5   G91 YR25              "Возврат в Y на ось сверления
N6   MR24 SR27            "Разгон оборотов шпинделя

```

```

N7 G01 G90 ZR31 FR26      "Сверление в плоскости RB
N8 M19                   "Ориентированный стоп
N9 G00 G91 YR28          "Отскок в оси Y
N10 G90 ZR32             "Выход в плоскости RC
N11 G70 G91 YR25         "Возврат в Y в ось сверления
"-----
" Пустой твердый цикл - для пользователя
"-----
N1 G79 L88
N2
N3 G70
"-----
" Пустой твердый цикл - для пользователя
"-----
N1 G79 L89
N2
N3 G70
*
```

12.3 Возможности обработки твердых циклов

Пользователь может поставленные твердые циклы обработать по необходимости или обычаем. Укажем некоторые возможности, с которыми мы столкнулись на практике. Некоторые пользователи хотят вводить скорость F и обороты S параметрически в твердом цикле (параметры R26 и R27), но хотят их ввести до вызова твердого цикла классическим программированием S и F. В этом случае достаточно из твердых циклов удалить все записи FR26 и SR27, как указано на примере твердого цикла G81:

Первоначальное состояние G81:

```

N1 G79 L81
N2 G00 G90 ZR30 FR26 SR27
N3 G01 ZR31
N4 G00 G70 ZR32
```

Обработанное состояние:

```

N1 G79 L81
N2 G00 G90 ZR30
N3 G01 ZR31
N4 G00 G70 ZR32
```

Следующие возможности можно использовать в твердом цикле G84 сдвиг оборотов (G95) вместо миллиметрового (G94), который обладает той выгодой, что можно менять обороты, если они не удовлетворяют условиям резки, хотя бы они меняли геометрические отношения во время резки резьбы. Эту выгоду, однако, до определенного уровня при G94 исключено использовать с головками для резки резьбы, которые могут своим отпружиниванием перенести разницу скорости сдвига и оборотов во время изменения оборотов. Изменение скорости %F заблокировано функцией M49.

Старшие типы твердых циклов не использовали плоскость RC для выезда из твердого цикла и заканчивали цикл в одинаковой плоскости, в которой цикл начался, т.е. в плоскости RA. Если бы у нас были готовые партпрограммы, которые плоскость RC не учитывают (не используют параметр R32), достаточно в твердых циклах заменить все ZR32 на ZR30.

Следующей возможностью является собственное создание твердого цикла, который не должен быть связан со сверлением, но возможно создать напр. твердый цикл для установки в положение для смены инструмента и его замены. В файле PCYKLYDG.NCP находятся для пользователя свободные циклы G88 и G89. Вы можете найти инспирацию в следующем примере:

```

N10 G79 L88
N20 G54 G0 G40 Z200. M5    „ уедет в безопасное положение и остановит шпиндель
N30 G59 Y0                „ уедет в положение для смены, положение введено в сдвиге G59
N40 TR12                  „ поиск инструмента, номер которого находится в параметре R12
N50 M06                   „ выполнит смену инструмента
N60 G70
```

12.4 Твердые циклы для токарных станков

Для создания и использования твердых циклов для токарных станков действительны одинаковые принципы как для твердого цикла фрезы. Подробнее опишем некоторую разницу.

Для токарных станков поставляются три файла с твердыми циклами: PEVNECY4.NCP, PEVNECY5.NCP и PEVNECY6.NCP. Файлы PEVNECY5.NCP и PEVNECY6.NCP содержат более этого по сравнению с файлом PEVNECY4.NCP возможность переменной резьбы, причем PEVNECY5.NCP предназначен для использования среднего программирования и файл PEVNECY6.NCP для программирования радиуса.

Пометка:

Файл PEVNECY4.NCP должен быть редактирован „вручную“ по тому, если используется программирование радиуса или программирование диаметра. Первоначально установлен на программирование диаметра.

Строки, которые обрабатываются, имеют следующую форму:

R14=+01333333 " Для прогр. диаметра: R33=R33+R33 (01333333) !

Если хотим программирование радиуса, должны обработаться следующим способом:

R14=+00000000 " Для прогр. диаметра: R33=R33+R33 (01333333) !

Для ввода параметров действительны одинаковые принципы, касающиеся использования десятичного знака, как в случае циклов сверления для фрез. Это, более этого, касается напр. отламливанием стружек, которое вводится без десятичного знака (или с десятичным знаком, но вместо сотых и тысячных частей – см. ниже). Для токарных инструментов твердых циклов важен также знак для некоторых параметров, который показывает направление сдвига. В случае программирования диаметра параметры, определяющие размеры в оси X, будут запрограммированы в среднем (напр. также толщина стружек!).

В случаях эти возможности обозначены следующим способом:

(+/-) Знак определяет направление

(Ø) Величина диаметра

12.5 Примеры твердых циклов для токарных инструментов

12.5.1 Продольная черновая обработка - G81

Использованные параметры:

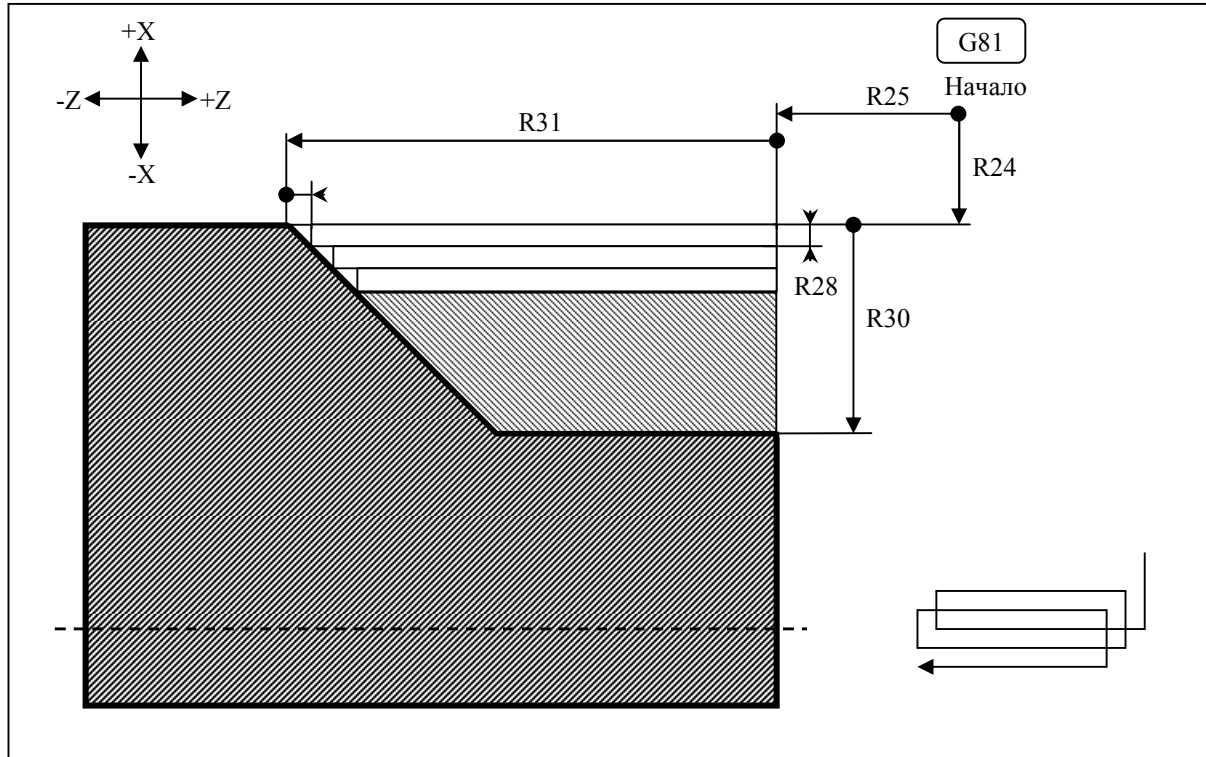
R24	(+/-)	(Ø)	Сколько мм до обрабатываемого полуфабриката в оси X, где начинается твердый цикл
R25	(+/-)		Сколько мм до торцевой части в оси Z, где начинается твердый цикл
R26			Скорость сдвига
R27			Количество повторяемых переездов по окончательной траектории (очистка)
R28		(Ø)	Толщина стружки в мм
R29	(+/-)		Сокращение расстояния Z на одну стружку
R30	(+/-)	(Ø)	Размер области черновой обработки в оси X
R31	(+/-)		Размер области черновой обработки в оси Z

Символ (+/-) в таблице обозначает, что указанный размер знаком определяет направление. Символ (Ø) в таблице обозначает, что указанный размер во время программирования диаметра программируется величиной диаметра и во время программирования радиуса радиусом.

Черновую обработку можно использовать во всех четырех зеркальных плоскостях. Плоскости черновой

обработки определены знаками параметров R30 и R31. Знаки параметров R24, R25 и R29 также зависят от использованной плоскости черновой обработки. Можно сказать, что все эти параметры (R24, R25, R29, R30, R31) запрограммированы приростом с учетом начальной точки, аналогично как это было для адресов I и J во время программирования окружности.

На рисунке указан пример черновой обработки, в котором предполагается, что параметры R24, R25, R30, R31 обладают отрицательными величинами.



Продольная черновая обработка G81 с очисткой (программирование диаметра)

Извлечение твердого цикла G81 из файла PEVNECY5.NCP

```
" ПРОДОЛЬНАЯ ЧЕРНОВАЯ ОБРАБОТКА - G81
N1 G79 L81
N2 G90 G24 " Абсолютно
      R00=+0
      R01=+1
      R02=+15113010 " Модификация инструкции сравнения
      R03=+17113010

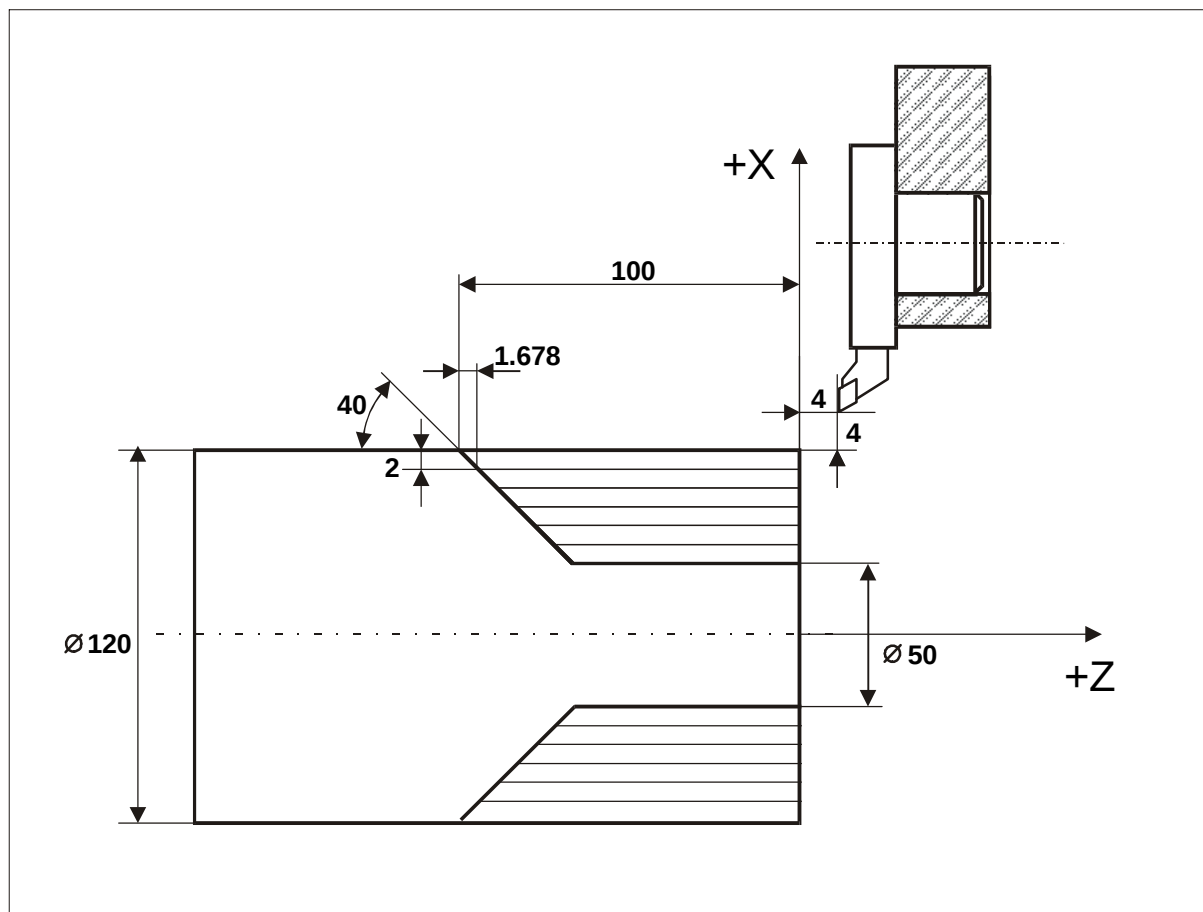
      G26 R05=+33070021 " PROGRAM 7 - 21

      R07=+00280039 " Запись R39=R28
      R08=+00800085 " Запись R35,R36=R30,R31
      R09=+31300000 " IF R30<R00 THEN
      R10=+00020003 " R03=R02
      R11=+08280028 " R28=-R28
      R12=+22000000 " ENDIF
      R13=+26010283 " Наполнение X прогр. до R33, Z прогр. до R34
      R14=+01333333 " Для среднего прогр: R33=R33+R33 (01333333)
      R15=+01332445 " R45=R33+R24
      R16=+01453030 " R30=R45+R30
      R17=+01452811 " R11=R45+R28 (первая стружка)
      R18=+01342538 " R38=R34+R25
      R19=+01383131 " R31=R38+R31
```

		R20=+00310044		" R44=R31 (запись для очистки)
		R21=+00330037		" R37=R33
N50	G00	FR26	XR11	" Перемещается в X на размер
	G26	R05=+01312931		" R31=R31+R29 Сокращение
N6	G01		ZR31	" Стружка
	G27	R05=+00370009		" R09=R37
		R06=+00030007		" R07=R03 (модифицированная инструкция)
N7			XR09	" Выезд в X
	G29	R05=+00110037		" Копирование R11 в R37
		R06=+01112811		" R11=R11+R28 (следующая стружка)
				" Если R11 > R30 потом R10=73 иначе R10 = 78
		R08=+00340009		" R09=R34
N8	G00		ZR09	" Возврат в Z
	GR10	L50 Q9998		" Условный скачок на блок N50
N10	G29	R05=+02464646		" R46=0
		R06=+19274610		" Если R27=R46(0) потом R10=73 иначе R10=78
		R07=+07350040		" R40=abs(R35) (R30)
		R08=+00270047		" R47=R27
N11	GR10	L30 Q9998		" Скачок, если нет очистки
	G28	R05=+03294090		" R40,41=R29.R40
		R06=+04903940		" R40=R40,41/39 (28)
		R07=+01444009		" R09=R44+R40
N12	G00	FR26	XR30	" Перемещается в X до последнего размера
	G26	R05=+00440011		" R11=R44
N13	G01		ZR09	" Очистка стружки
	G26	R05=+00450010		" R10=R45
N14		XR10	ZR11	" Цело
	G26	R05=+00330010		" R10=R33
N15			XR10	
	G28	R05=+00340010		" R10=R34
		R06=+02470147		" R47=R47-R01 декремент
		R07=+15474611		" Если R47>R46(0) потом R11=73 иначе R11=78
N16	G00		ZR10	" Возврат в Z
	GR11	L12 Q9998		
N17	G70			
N30	G00	FR26	XR30	" Перемещается в X до последнего размера
	G26	R05=+01312931		" R31=R31+R29 Сокращение
N31	G01		ZR31	" Последняя стружка
	G26	R05=+00330009		" R09=R33
N32			XR09	" Возврат в X
	G26	R05=+00340009		" R09=R34
N33	G00		ZR09	" Возврат в Z
N35	G70			
"				

12.5.2 Пример использования твердого цикла G81 для продольной черновой обработки

На следующем рисунке находится чертеж с размерами частей. Указанный пример программы предполагает программирование диаметром, система координат по рисунку (положительное направление оси X наверх, Z направо). Исходная точка твердого цикла изображает координаты ножа токарного станка.



Программа для черновой обработки была бы записана следующим способом:

```

N10 G54 &1100 D1 T1 M3 S500      "Главное предложение
N20 X0 Z300
N30 X128 Z4      " Установка в исходное положение
N40 R24= -8.000   " Сколько мм до обраб. полуфабриката в X, (знак = направление)
    R25= -4.000   " Сколько мм до обраб. полуфабриката в Z, (знак = направление)
    R26=200       " Скорость сдвига
    R27=0.002     " Количество повторяемых переездов по окончательной траектории
    R28=4.000     " Толщина стружки в мм
    R29=1.678     " Сокращение расстояния Z на одну стружку (знак = направление)
    R30= -70.000  " Размер черновой обработки в оси X, (знак = направление)
    R31= -100.000 " Размер черновой обработки в оси Z, (знак = направление)
G81
N50 G80
N60 M30
    
```

Пометка:

В случае среднего программирования размеры в X программируются значением диаметра. Толщина стружки 2 мм, поэтому программируется R28=4.0, аналогично еще параметрически R24 и R30.

В случае программирования радиуса указанные параметры программируются радиусом, т.е. были бы в два раза меньше, чем в указанном примере (R28=2.0).

Указанным способом программируются также остальные твердые циклы для токарных станков.

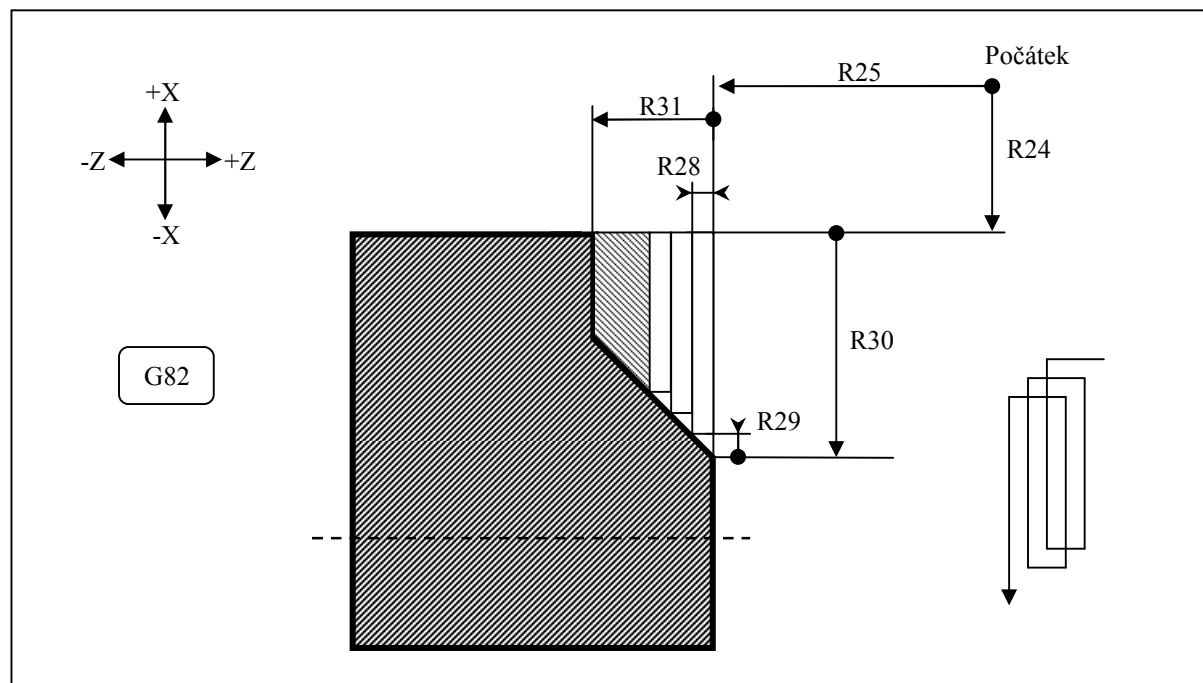
12.5.3 Поперечная черновая обработка - G82

Использованные параметры:

R24	(+/-)	(Ø)	Сколько мм до обрабатываемого полуфабриката в оси X, где начинается твердый цикл
R25	(+/-)		Сколько мм до торцевой части в оси Z, где начинается твердый цикл
R26			Скорость сдвига
R27			Количество повторяемых переездов по окончательной траектории (очистка)
R28		(Ø)	Толщина стружки в мм
R29	(+/-)		Сокращение расстояния X на одну стружку
R30	(+/-)	(Ø)	Размер области черновой обработки в оси X
R31	(+/-)		Размер области черновой обработки в оси Z

Черновую обработку можно использовать во всех четырех зеркальных плоскостях. Плоскости черновой обработки обозначены знаками параметров R30 и R31. Знаки параметров R24, R25 и R29 также зависят от использованной плоскости черновой обработки. Можно сказать, что все эти параметры (R24, R25, R29, R30, R31) запрограммированы приростом с учетом начальной точки, аналогично как это для адресов I и J во время программирования окружности.

На рисунке указан пример черновой обработки, в котором предполагается, что параметры R24, R25, R30, R31 обладают отрицательными величинами.



Поперечная черновая обработка G82 с очисткой (программирование диаметра)

Извлечение твердого цикла G82 из файла PEVNESY5.NCP

```

"
N1      G79    L82
N2      G90    G24          " Абсолютно
                R00=+0
                R01=+1
                R02=+15113110      " Модификация инструкции сравнения
                R03=+17113110

                G26    R05=+33070021      " PROGARM 7 - 21

                R07=+00280039      " Запись R39=R28
                R08=+00800085      " Запись R35,R36=R30,R31
                R09=+31310000      " IF R31<R00 THEN
                R10=+00020003      "      R03=R02
                R11=+08280028      "      R28=-R28
                R12=+22000000      " ENDIF
                R13=+26010283      " Запишет X прогр. в R33 и Z прогр. в R34
                R14=+01333333      " Для прогр. диаметра: R33=R33+R33 (01333333) !
                R15=+01342545      " R45=R34+R25
                R16=+01453131      " R31=R45+R31
                R17=+01452811      " R11=R45+R28 (первая заноза)
                R18=+01332438      " R38=R33+R24
                R19=+01383030      " R30=R38+R30
                R20=+00300044      " R44=R30 запись для очистки
                R21=+00340037      " R37=R34

N50     G01    FR26      ZR11  " Перемещается в Z на размер
        G26    R05=+01302930      " Сокращение R30=R30+R29
N6      XR30   " Стружка
        G27    R05=+00370009      " R09=R37
                R06=+00030007      " R07=R03 (модифицированная инструкция)
N7      ZR09   " Выезд в Z
        G29    R05=+00110037      " R37=R11
                R06=+01112811      " R11=R11+R28 (следующая стружка)
                R08=+00330009      " Если R11 > R31 потом R10=73 иначе R10 = 78
                R09=R33
N8      G00    XR09      " Возврат в X
        GR10   L50      Q9999      " Условный скачок на блок N50

N10     G29    R05=+02464646      " R46=0
                R06=+19274610      " Если R27 = R46(0) потом R10=73 иначе R10 = 78
                R07=+07360040      " R40=abs(R36) (R31)
                R08=+00270047      " R47=R27
N11     GR10   L30      Q9999      " Скачок, если нет очистки
        G28    R05=+03294090      " R40,41=R29.R40
                R06=+04903940      " R40=R40,41/39 (28)
                R07=+01444009      " R09=R44+R40
N12     G01    FR26      ZR31  " Перемещается в Z на последнее расстояние
        G26    R05=+00440011      " R11=R44
N13     XR09   " Очистка стружки
        G26    R05=+00450010      " R10=R45
N14     ZR10   XR11      " Торцевая часть
        G26    R05=+00330010      " R10=R33
N15     ZR10   "
        G28    R05=+00340010      " R10=R34

```

```

R06=+02470147      " R47=R47-R01 декремент
R07=+15474611      " Если R47 > R46(0) потом R11=73 иначе R11 = 78
N16  G00            XR10  " Возврат в X
      GR11  L12      Q9999  " Условное повторение
N17  G70            " Конец

N30  G01  FR26      ZR31  " Перемещается в Z на последнее расстояние
      G26  R05=+01302930  " R30=R30+R29 Сокращение
N31  XR30  " Последняя стружка
      G26  R05=+00340009  " R09=R34
N32  ZR09  " Возврат в Z
      G26  R05=+00340009  " R09=R34
N33  G00            XR09  " Возврат в X
N35  G70            " Конец
"
"
"

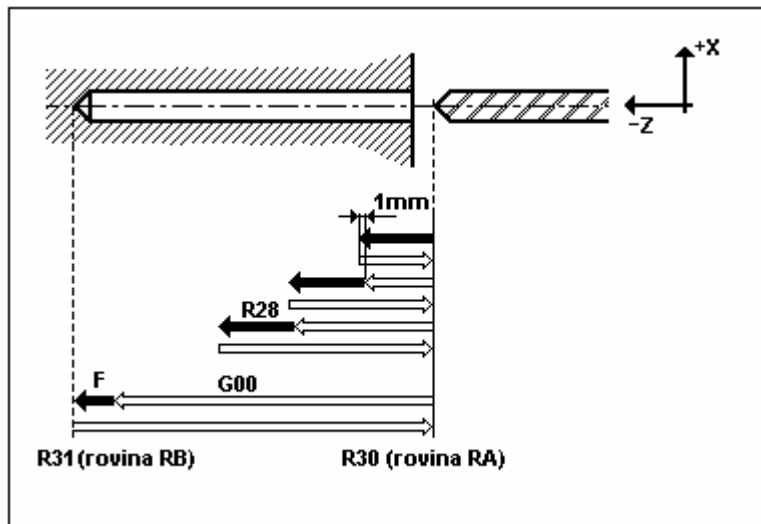
```

12.5.4 Сверление глубоких дыр - G83

Использованные параметры:

R26		Скорость сдвига
R27		Величина оборотов (адрес S) в соответствии со степенью переноса
R28		Прирост сверления - q
R29		Не использован
R30	(+/-)	Декларирует координату плоскости RA
R31	(+/-)	Декларирует координату плоскости RB

На рисунке указан пример цикла G83:



R31 (плоскость RB) R30 (плоскость RA)

```

Извлечение из твердого цикла G83 из файла PEVNESY5.NCP
"
" СВЕРЛЕНИЕ ГЛУБОКИХ ДЫР С ОПОЛАСКИВАНИЕМ - G83
"
"
N1 G79 L83
" СВЕРЛЕНИЕ В ОСИ Z В НАПРАВЛЕНИИ МИНУС
"
N210 G26 R5=15313024 "IF R31 > R30 THEN R24=73 (СВЕРЛЕНИЕ В
НАПРАВЛЕНИИ ПЛЮС)
N220 GR24 L240
"
N40 G27 R5=00300029 R6=00300025 R24=1.000
N50 G26 R5=16312923 " IF R31>R29 then R23=73 else 78
N60 GR23 L140
N70 ZR25 G00 G26 R5=02292829 " R29 = R29 - R28
N80 G26 R5=15293123 " IF R31>R29 then R23=73 else 78
N90 GR23 L110 Q9998 " Перескочит блок 10

N100 G26 R5=00310029 " R29 = R31
N110 ZR29 G01 FR26 " сверлит рабочей скоростью
N120 ZR30 G00 " быстрым сдвигом уедет в плоскость
RA
N130 G26 R5=01292425 G73 L50 Q9998 " R25 = R29 + R24 (R24 = 1 мм)
N140 G70
"
" СВЕРЛЕНИЕ В ОСИ Z В НАПРАВЛЕНИИ ПЛЮС
"
N240 G27 R5=00300029 R6=00300025 R24=1.000 "R30 -> R29
N250 G26 R5=18312923 " IF R31 <= R29 then R23 = 73 else
78
N260 GR23 L340
N270 ZR25 G00 G26 R5=01292829 " R29 = R29 + R28
N280 G26 R5=17293123 " IF R31 < R29 then R23 = 73 else 78
N290 GR23 L310 Q9998 " Перескочит блок 300

N300 G26 R5=00310029 " R29 = R31
N310 ZR29 G01 FR26 " сверлит рабочей скоростью
N320 ZR30 G00 " быстрым сдвигом уедет в плоскость
RA
N330 G26 R5=02292425 G73 L250 Q9998 " R25 = R29 - R24 (R24 = 1 мм)
N340 G70
"
"

```

12.5.5 Резка резьбы на цилиндрической плоскости - G84

Использованные параметры:

R19			Сколько материала оставить в случае переменной черновой обработки резьбы на ее стене (напр. 0.05 мм). Этот прирост отнимется во время последнего n-прохождения согласно параметру R21
R20			Имеет значение только для переменной резки резьбы: Для метрической резьбы вводится R20=0 (или 60 градусов). Для резьбы Витворта вводится R20=55 (градусов) Для трапециoidalной резьбы вводится R20=30 (градусов) Другие величины, чем здесь указанные, также выполняют метрическую резьбу. Пометка: Пока установлена только метрическая резьба !!!

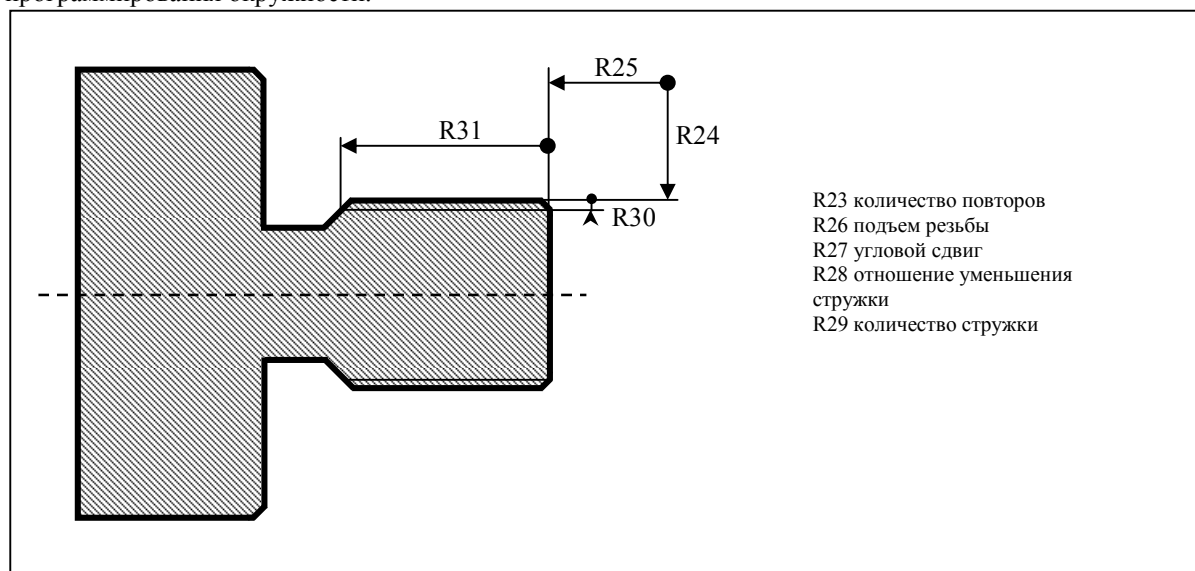
R21			Способ резки резьбы (0=в оси угла резьбы, другие чем ноль=переменная, номер указывает сколько стружек до конца перескочить с переменной)
R22			Для резьбы без выезда не использован
R23			Количество повторяемых переездов по окончательной траектории
R24	(+/-)	(Ø)	Сколько мм до обрабатываемого полуфабриката в оси X начинает РС – знак определяет направление
R25	(+/-)		Сколько мм до торцевой части в оси Z начинает РС – знак определяет направление
R26			Шаг резьбы
R27			Угловой сдвиг начала резки резьбы
R28			Во сколько раз следующая стружка меньше, чем предыдущая (коэффициент < 1), рекомендованная величина коэффициента 0.8
R29			Количество стружки, которыми должна быть резьба вырезана Пометка: в случае переменной резки резьбы под одной стружкой подразумеваются два прохождения на одинаковой глубине резьбы !!!
R30	(+/-)	(Ø)	Глубина резьбы в оси X – знак определяет направление
R31	(+/-)		Длина резьбы в оси Z – знак определяет направление

Пометка: Параметры R19 и R20 необходимы только для переменной резки резьбы (R21 не = 0).

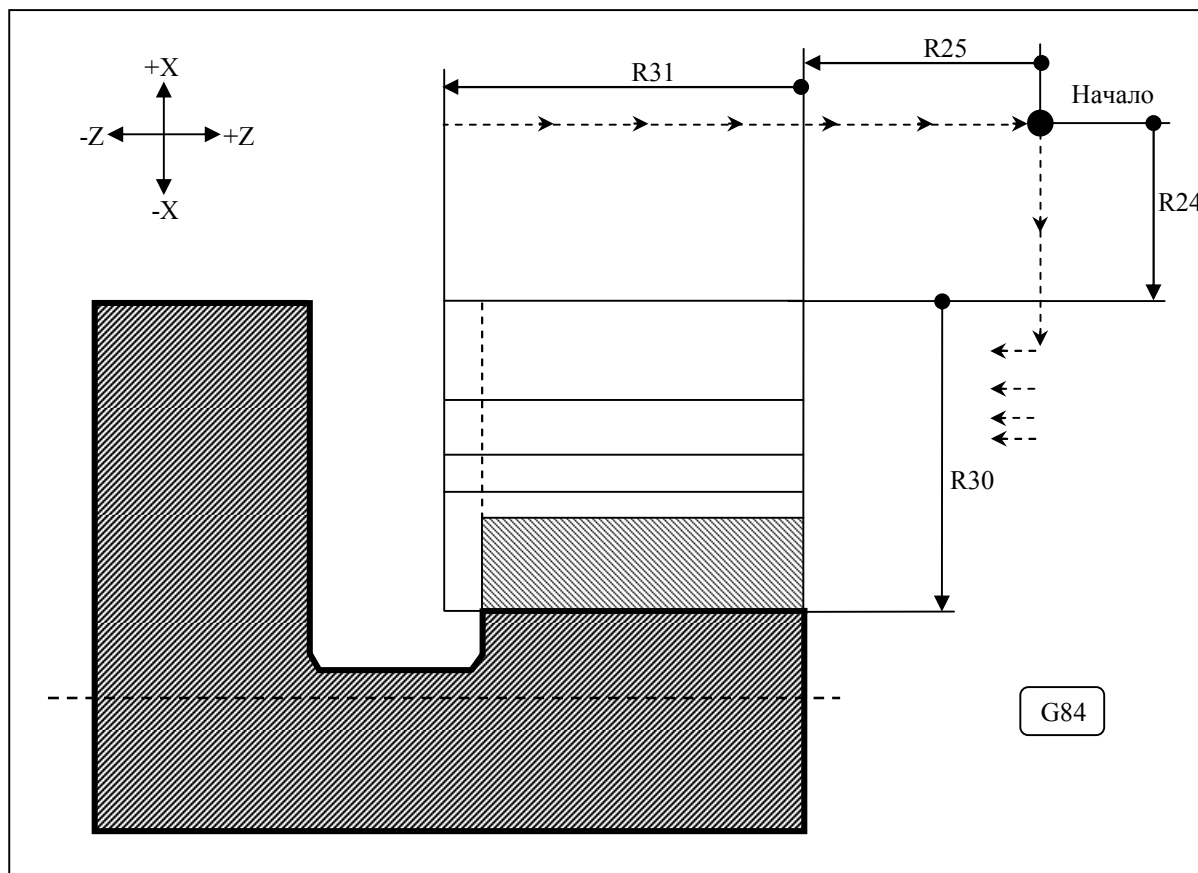
Параметр R28 (коэффициент) определяет на сколько раз следующая стружка меньше предыдущей стружки во время резки резьбы. Стружки во время резки резьбы постепенно уменьшаются по этой константе и создают геометрический ряд с коэффициентом меньше, чем один. Например, в случае толщины стружки 0,5 мм и коэффициенте 0,8 будут следующей толщины:

$$\begin{aligned}
 &0,5 \text{ мм} \\
 0,50 \cdot 0,8 &= 0,4 \text{ мм} \\
 0,40 \cdot 0,8 &= 0,32 \text{ мм} \\
 0,32 \cdot 0,8 &= 0,256 \text{ мм}
 \end{aligned}$$

Резку резьбы можно использовать во всех четырех зеркальных плоскостях. Плоскости резки резьбы определены знаками параметров R30 и R31. Знаки параметров R24, R25 также зависят от использованной плоскости резки резьбы. Можно сказать, что все эти параметры (R24, R25, R30, R31) запрограммированы приростом с учетом начальной точки аналогично как это указано в адресах I и J во время программирования окружности.



На рисунке указан пример резки резьбы, в котором предполагается, что параметры R24, R25, R30, R31 обладают отрицательным значением:



Резка резьбы на цилиндрической плоскости (программирование диаметра)

" РЕЗКА РЕЗЬБЫ НА ЦИЛИНДРИЧЕСКОЙ ПЛОСКОСТИ - G84

"

N1 G79 L84

N2 R00=0 R22=0

" Для резьбы без выезда обеспечит 0

G28 R05=+19210002

" IF R21=0 THEN R02=73 else R02=78

R06=00190047

" R19 -> R47 (прирост нужен только для переменного)

R07=00260048

" в R48 ненулевая величина как признак перемены

N3 GR02 L5

" если резка в оси, потом скачок на N5

" -----

" ПЕРЕМЕННАЯ РЕЗКА

" -----

N28 G90 G24 FR26

" Абсолютно

R00=+0

" R00=0

R01=+1

" R01=1

R02=-2

" R02=-2

R03=+2.000

" Вспомогательная во время разделения на 2

G26 R05=+33060020

" Программа 06-20

R06=+00300044

" Запись глубины резьбы в R44

R07=+23290039

" R39=реал(R29)

R08=+28283940

" R40=сумма геометрического ряда, коэффициент R28

R09=+07300039

" R39=абс(R30)

R10=+04394040

" R40=R39/R40 = 1. шип

R11=+31300000

" IF R30<R00 THEN

R12=+08400040

" R40=-R40

```

R13=+22000000 " ENDIF
R14=+26010283 " Запись X прогр. в R33 и Z прогр. в R34
R15=+01333333 " Для среднего прогр.: R33=R33+R33 (01333333) !
R16=+00800085 " Запись R35,36=R30,31
R17=+01332411 " R11=R33+R24
R18=+01113030 " R30=R11+R30 окончательный размер
R19=+01114011 " R11=R11+R40 (первая стружка)
R20=+01342538 " R38=R34+R25
N30 G29 R05=+01383131 " R31=R38+R31
R06=+01022910 " R10=R02+R29 (количество повторов)
R07=+04260337 " R37=R26/R03: подъем/2
R08=+00400046 " R40 -> R46 стружку будет считать в R46

N40 R18=34.000 " Константа для образца метрическая резьба.
R20=69999.999 " Только для сравнения, чтобы впервые рассчитать
                сдвиг
G29 R05=+02102121 " R21=R10-R21, R21 это счетчик переменного хода
R06=+01210121 " R21=R21+R01 увеличение на 1
R07=+01210121 " R21=R21+R01 больше еще на 1
R08=+03182641 " (34 x S) в параметре R41 (для образца)
"
"-----
"----- Первый проход резьбы -----
"-----
" Рассчитает сдвиг P в оси Z для метрической резьбы согласно формуле:
" P = сдвиг, S = подъем, T = глубина стружки
" Пометка: 34 x S не меняется и подсчитано заранее в R41
"  $P = ((34 \times S) - (55.426 \times T)) / 96$ 

N55 R17=96.000 " Константа для формулы метрическая резьба.
R18=34.000 " Константа для формулы метрическая резьба.
R19=55.426 " Константа для формулы метрическая резьба.
R32=-1 " Прогр. радиуса R32= 0 !!!
" Прогр. диаметра R32=-1 (бинарный сдвиг направо)

G29 R05=09463242 " Во время среднего прогр. разделяет стружку на 2
                (бинарный сдвиг)
R06=03194245 " (55.426 x T) в параметр R45
R07=07450045 " Абсолютная величина R45 (всегда положительный)
R08=02414543 "  $((34 \times S) - (55.426 \times T))$  в R43

N56 G29 R05=21002018 " R20 другое, чем 0? (сдвиг не 0 ?)
R06=04431720 " ДА:  $((34 \times S) - (55.426 \times T)) / 96$  в R20
R07=22000000 " Конец условия

" В R20 находится рассчитанный сдвиг, от сдвига вычитается еще случайный
прирост
" Или в R20 уже находится 0.

N57 G28 R05=+02204703 " Вычитает еще черновую обработку прирост и
                запишет в R03
R06=+00030044 " R03 записано еще в R44
R07=+02340303 " от запрограммированной величины вычитается
                сдвиг

N60 G00 XR11 ZR03 " Перемещается в X и Z на размер
G27 R05=+00330009 " R33 -> R09
R06=+00230039 " R23 -> R39 количество по одинаковой траектории
" N61/ G73 L70 Q9998

```

```

" N62 M0
N70 G33 IR27 ZR31 " G33 резка резьбы
N80 G00 XR09 " Выезд в X
      G26 R05=+00340009 " R34 -> R09, Начальный размер в Z
N90 G00 ZR09 " Возврат в Z
      G28 R05=+00440003 " Выберет записанное R44 (сдвиг) и запишет в R03
      R06=+01340303 " R03 = R34 + R03 ... прогр. величина + сдвиг
      R07=+19480008 " if R48 = 0 then G73 else G78 в R08

N110 GR08 L160 Q9998 " (если признак = 0, уже не делает второй проход)
"-----
" Второй проход на одинаковой глубине X, сдвинуты во вторую сторону резьбы
"-----
N120 G00 XR11 ZR03 " Перемещается в X и Z на расстояние
N130 G33 IR27 ZR31 " G33 - резка резьбы
      G26 R05=00330009 " R09=R33
N150 G00 XR09 " Выедет в оси X
N160 G26 R05=+33120017 " ПРОГРАММА ОТ R12 ДО R17
      R12=+03402840 " R40=R40xR28 - умножение на коэффициент
      R13=+01404646 " стружка записывается в R46
      R14=+01114011 " R11=R11+R40 (следующая стружка)
      R15=+00340009 " R09=R34
      R16=+02210121 " R21=R21-R01, уменьшит счетчик ходов на 1
      R17=+15210008 " if R21 > 0 then G73 else G78 do R08
N180 GR08 L200 Q9998 " счетчик > 0 потом скачок на N200
N185 G26 R05=19480008 " признак перемены в R48 уже = 0 ?
N186 GR08 L200 Q9998 " ДА, потом скачок на N200

N190 G28 R05=+00000048 " Обнулит признак перемены в R48
      R06=+00000020 " Обнулит R20 (рассчитанный сдвиг)
      R07=+00000047 " Обнулит R47 (прирост черновой обработки)
N200 G00 ZR09 " Возврат в Z
      G73 L55 QR10 " Условный скачок на блок N55
N210 G00 FR26 XR30 " Перемещается в X на последнее расстояние
      G27 R05=+00330009 " R09=R33
      R06=+02390139 " R39=R39-R01
N220 G33 IR27 ZR31 " G33 Резка резьбы на расстояние
N230 G00 XR09 " Возврат в X
      G27 R05=+00340009 " R09=R34
      R06=+16390010 " IF R39 >= R00 THEN R10=73 ELSE R10=78
N240 G00 ZR09 " Возврат в Z
      GR10 L210 Q9998 " Условный скачок
N250 G70 " Конец
"-----
" РЕЗКА В ОСИ
"-----
N5 G90 G24 FR26 " Абсолютно
      R00=+0 " R00=0
      R01=+1 " R01=1
      R02=-2 " R02=-2

      G26 R05=+33060021 " Программа 06-21
      R06=+23290039 " R39=реал(R29)
      R07=+28283940 " R40=сумма геометрического ряда, коэффициент
                       R28
      R08=+07300039 " R39=abs(R30)
      R09=+04394040 " R40=R39/R40 = 1. шип
      R10=+31300000 " IF R30<R00 THEN
      R11=+08400040 " R40=-R40
      R12=+22000000 " ENDIF
      R13=+26010283 " Запишет X прогр. в R33 и Z прогр. в R34

```

```

R14=+01333333      " Для среднего прогн: R33=R33+R33 (01333333)
!
R15=+00800085      " Запись R35,36=R30,31
R16=+01332411      " R11=R33+R24
R17=+01113030      " R30=R11+R30 окончательный размер
R18=+01114011      " R11=R11+R40 (первая стружка)
R19=+01342538      " R38=R34+R25
R20=+01383131      " R31=R38+R31
R21=+01022910      " R10=R02+R29 (количество повторов)

N50 G00 XR11        " Перемещение в X на расстояние
    G27 R05=+00330009 " R09=R33
    R06=+00230039    " R39=R23 количество по одинаковой траектории
"N51/G73 L6 Q9998
"N52 M0
N6 G33 IR27         ZR31 " G33 Резка резьбы
N7 G00              XR09 " Выезд в X
    G28 R05=+03402840 " R40=R40.R28 - умножение на коэффициент
    R06=+01114011    " R11=R11+R40 (следующая стружка)
    R07=+00340009    " R09=R34
N8 G00              ZR09 " Возврат в Z
    G73 L50 QR10      " Условный скачок на блок N50
N9 G00 FR26         XR30 " Перемещается в X на последнее расстояние
    G27 R05=+00330009 " R09=R33
    R06=+02390139    " R39=R39-R01
N10 G33 IR27        ZR31 " G33 Резка резьбы по расстоянию
N11 G00             XR09 " Возврат в X
    G27 R05=+00340009 " R09=R34
    R06=+16390010    " IF R39>=R00 THEN R10=73 ELSE R10=78
N12 G00             ZR09 " Возврат в Z
    GR10 L9 Q9998    " Условный скачок
N13 G70             " Конец
    
```

12.5.6 Резка резьбы на цилиндрической плоскости с выездом - G85

Использованные параметры:

R04			Угол выезда из резьбы – описание параметра см. раздел резка резьбы !!!
R19			Сколько материала оставить в случае переменной черновой обработки резьбы на стене резьбы (напр. .05 мм). Этот прирост отнимется во время последнего п-прохода согласно параметру R21
R20			Имеет значение только для переменной резки резьбы: Для метрической резьбы вводится R20=0 (или 60 градусов). Для резьбы Витворта вводится R20=55 (градусов) Для трапециoidalной резьбы вводится R20=30 (градусов) Другие величины, чем здесь указанные, выполняют также метрическую резьбу. Пометка: Пока установлена только метрическая резьба!!!
R21			Способ резки резьбы (0=в оси угла резьбы, другие чем ноль = переменная, номер указывает сколько стружек до конца закончить с переменной)
R22			Длина выезда
R23			Количество повторяемых переездов по окончательной траектории
R24	(+/-)	(Ø)	Сколько мм до обрабатываемого полуфабриката в оси X начинает РС – знак определяет направление
R25	(+/-)		Сколько мм до торцевой части в оси Z начинает РС – знак определяет направление
R26			Подъем резьбы
R27			Угловой сдвиг начала резки резьбы
R28			Во сколько раз последующая стружка меньше, чем предшествующая (коэффициент < 1), рекомендованная величина коэффициента 0.8
R29			Количество стружек, которыми должна быть резьба вырезана

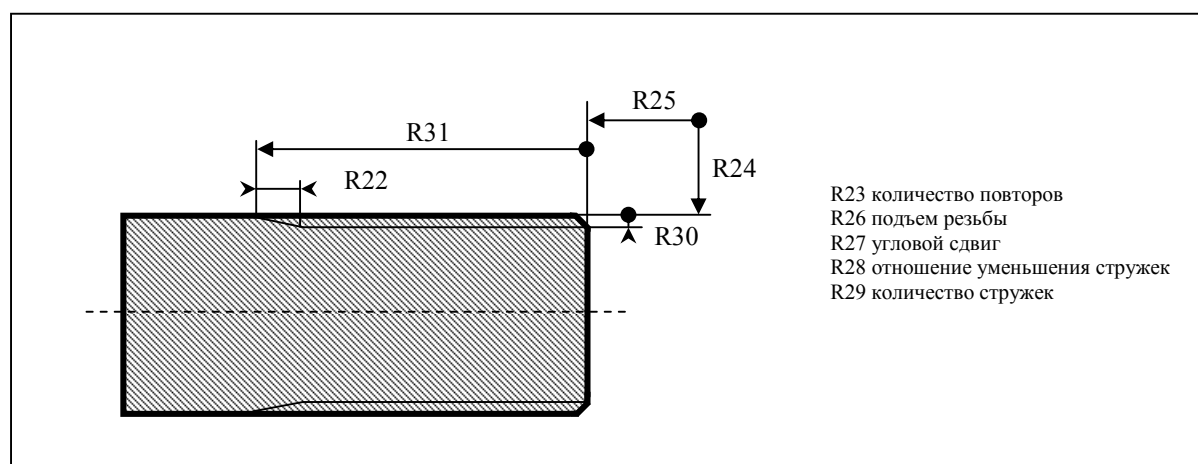
			Пометка: во время переменной резки резьбы под одной стружкой понимаются два прохода на одинаковой глубине резьбы !!!
R30	(+/-)	(Ø)	Глубина резьбы в оси X – знак определяет направление
R31	(+/-)		Длина резьбы в оси Z – знак определяет направление

Параметр R28 – см. описание G84

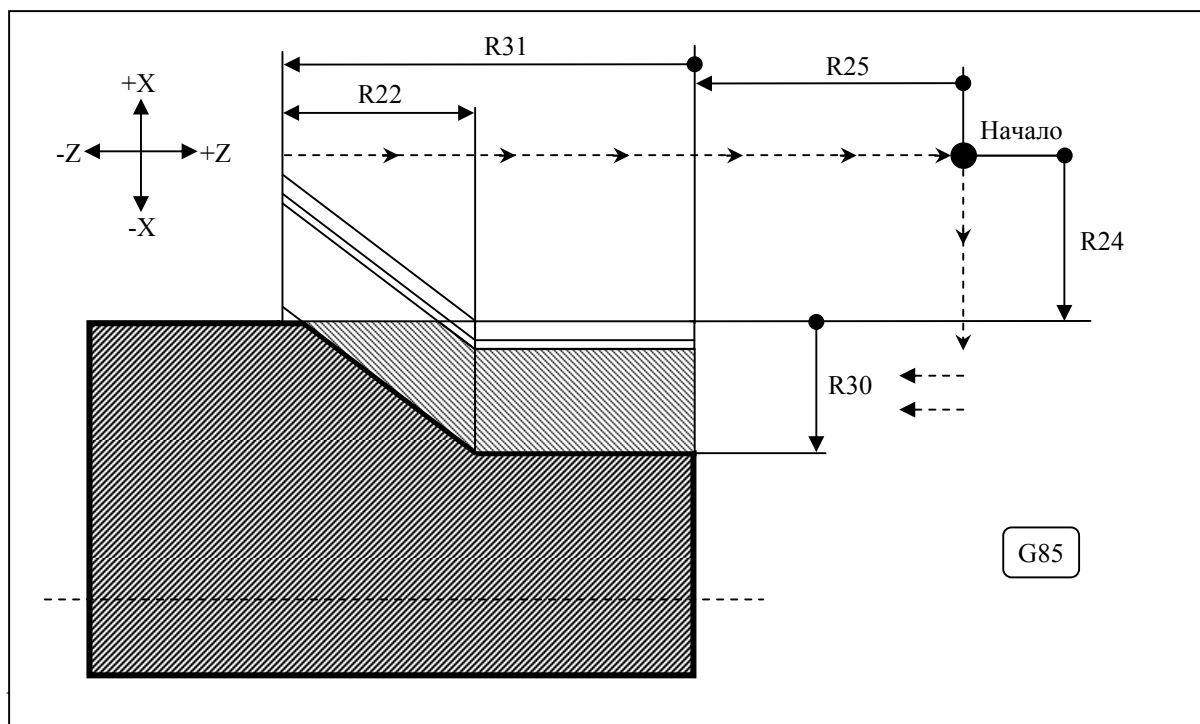
Параметр R22 определяет длину выезда (положительный номер). Общая длина резьбы определена параметром R31, включая длину выезда.

Угол выезда определен или машинной константой номер 8 (файл TAB0.REK) или параметром, номер которого указан в этой машинной константе. Подробности см. раздел РЕЗКА РЕЗЬБЫ. Длина выезда должна быть определена таким способом, чтобы во время последней резки резьбы хода выехал нож над материалом.

Резку резьбы можно использовать во всех четырех зеркальных плоскостях. Плоскости резки резьбы определены знаками параметров R30 и R31. Знаки параметров R24, R25 также зависят от использованной плоскости резки резьбы. Можно сказать, что все эти параметры (R24, R25, R30, R31) запрограммированы приростом с учетом начальной точки аналогично как этому было в адресах I и J во время программирования окружности.



На рисунке указан пример резки резьбы с выходом, в котором предполагается, что параметры R24, R25, R30, R31 обладают отрицательными величинами.



Резка резьбы на цилиндрической плоскости с выездом

```

" РЕЗКА РЕЗЬБЫ НА ЦИЛИНДРИЧЕСКОЙ ПЛОСКОСТИ С ВЫЕЗДОМ - G85
"
N1  G79  L85
N2  G28  R05=+19210002  " IF R21=0 THEN R02=73 else R02=78
      R06=00190047      " R19 -> R47 (прирост нужен только для переменной)
      R07=00260048      " в R48 не нулевая величина как признак перемены
      R00=0
N3  GR02 L5              " если резка в оси, потом скачок на N5
" -----
" ПЕРЕМЕННАЯ РЕЗКА
" -----
N28  G90  G24  FR26      " Абсолютно
      R00=+0            " R00=0
      R01=+1            " R01=1
      R02=-2            " R02=-2
      R03=+2.000        " Вспомогательная для деления на 2

      G26  R05=+33060020 " Программа 06-20
      R06=+00300044      " Запись глубины резьбы в R44
      R07=+23290039      " R39=реал(R29)
      R08=+28283940      " R40=сумма геометрического ряда, коэффициент R28
      R09=+07300039      " R39=абс(R30)
      R10=+04394040      " R40=R39/R40 = 1. шип
      R11=+31300000      " IF R30<R00 THEN
      R12=+08400040      " R40=-R40
      R13=+22000000      " ENDIF
      R14=+26010283      " Запишет X прогр. в R33 и Z прогр. в R34
      R15=+01333333      " Для прогр. диаметра: R33=R33+R33 (01333333) !
      R16=+00800085      " Запись R35,36=R30,31
      R17=+01332411      " R11=R33+R24
      R18=+01113030      " R30=R11+R30 окончательный размер
      R19=+01114011      " R11=R11+R40 (первая стружка)
      R20=+01342538      " R38=R34+R25
N30  G29  R05=+01383131  " R31=R38+R31
      R06=+01022910      " R10=R02+R29 (количество повторов)
      R07=+04260337      " R37=R26/R03: подъем/2
      R08=+00400046      " R40 -> R46 стружка будет считаться в R46

N40  R18=34.000          " Константа для формулы метрическая резьба.
      R20=69999.999      " Только для сравнения, чтобы впервые рассчитать
                          " сдвиг
      G29  R05=+02102121 " R21=R10-R21, R21 это счетчик перемен прохода
      R06=+01210121      " R21=R21+R01 больше на 1
      R07=+01210121      " R21=R21+R01 повышается еще на 1
      R08=+03182641      " (34 x S) в параметр R41 (для формулы)
"
"-----
"----- Первый проход резьбы -----
"-----
" Рассчитает сдвиг Р в оси Z для метрической резьбы по формуле:
" Р = сдвиг, S = подъем, Т = глубина стружки
" Пометка: 34 x S не меняется и рассчитано заранее в R41
" Р=((34 x S) - (55.426 x Т))/96

N55  R17=96.000          " Константа для формулы метрическая резьба.
      R18=34.000          " Константа для формулы метрическая резьба.

```

```

R19=55.426      " Константа для формулы метрической резьбы.
R32=-1          " Программирование радиуса R32= 0 !!!
                " Программирование диаметра R32=-1 (бинарный сдвиг
                направо)

G29  R05=09463242  " В случае программирования диаметра разделяет
                стружку на 2 (бин. сдвиг)
                R06=03194245  " (55.426 x T) в параметр R45
                R07=07450045  " Абсолютная величина R45 (всегда положительная)
                R08=02414543  " ((34 x S) - (55.426 x T)) в R43

N56  G29  R05=21002018  " R20 не = 0? (сдвига нет 0 ?)
                R06=04431720  " ДА: ((34 x S) - (55.426 x T))/96 в R20
                R07=22000000  " Конец условия

" В R20 рассчитан сдвиг, со сдвига вычитается еще случайный прирост
" Или в R20 уже находится 0.

N57  G28  R05=+02204703  " Вычитается еще прирост черновой обработки и
                запишется в R03
                R06=+00030044  " R03 записано еще в R44
                R07=+02340303  " от запрограммированной величины вычитается
                сдвиг

N60  G00  G98 XR11  ZR03  " Перемещается в X и Z на размер
                G27  R05=+00330009  " R33 -> R09
                R06=+00230039  " R23 -> R39 количество проходов по одинаковой
                траектории

"N61/G73 L70 Q9998
"N62 M0
N70  G33  IR27 JR22 ZR31  " G33 резка резьбы
N75                                " Пустой блок должен быть
N80  G00  G24 XR09 ZR31  " Выезд в X
                G26  R05=+00340009  " R34 -> R09, Начальный размер в Z
N90  G00  ZR09            " Возврат в Z
                G28  R05=+00440003  " Выберет записанный R44 (сдвиг) и запишет в R03
                R06=+01340303  " R03 = R34 + R03 ... прогр. величина + сдвиг
                R07=+19480008  " if R48 = 0 then G73 else G78 в R08

N110 GR08 L160 Q9998      " (если признак = 0, уже не выполняет второй
                проход)
"-----
" Второе прохождение на одинаковой глубине X, сдвинутое во вторую сторону
резьбы
"-----

N120 G00 G98 XR11  ZR03  " Перемещается в X и Z на размер
N130 G33 IR27 JR22 ZR31  " G33 - резка резьбы
                G26  R05=00330009  " R09=R33
N135                                " Должен существовать пустой блок
N150 G00 G24 XR09  ZR31  " Выедет в оси X
N160 G26 R05=+33120017  " ПРОГРАММА ОТ R12 ДО R17
                R12=+03402840  " R40=R40xR28 - умножение на коэффициент
                R13=+01404646  " стружка записывается в R46
                R14=+01114011  " R11=R11+R40 (следующая стружка)
                R15=+00340009  " R09=R34
                R16=+02210121  " R21=R21-R01, уменьшит датчик проходов на 1
                R17=+15210008  " if R21 > 0 then G73 else G78 в R08
N180 GR08 L200 Q9998      " Датчик > 0 потом скачок на N200
N185 G26  R05=19480008  " признак перемены в R48 уже = 0 ?
N186 GR08 L200 Q9998      " ДА, потом скачок на N200

```

```

N190 G28 R05=+00000048 " Обнуляет признак перемены в R48
      R06=+00000020 " Обнуляет R20 (рассчитанный сдвиг)
      R07=+00000047 " Обнуляет R47 (прирост черновой обработки)
N200 G00 ZR09 " Возврат в Z
      G73 L55 QR10 " Условный скачок на блок N55
N210 G00 G98 FR26 XR30 " Перемещается в X на последний размер
      G27 R05=+00330009 " R09=R33
      R06=+02390139 " R39=R39-R01
N220 G33 IR27 JR22 ZR31 " G33 Резка резьбы по размеру
N225 " Должен находиться пустой блок
N230 G00 G24 XR09 ZR31 " Возврат в X
      G27 R05=+00340009 " R09=R34
      R06=+16390010 " IF R39 >= R00 THEN R10=73 ELSE R10=78
N240 G00 ZR09 " Возврат в Z
      GR10 L210 Q9998 " Условный скачок
N250 G70 " Конец
"-----
" РЕЗКА В ОСИ
"-----
N5 G90 G24 FR26 " Абсолютно
      R00=+0 " R00=0
      R01=+1 " R01=1
      R02=-2 " R02=-2

      G26 R05=+33060021 " Программа 06-21
      R06=+23290039 " R39=реал(R29)
      R07=+28283940 " R40=сумма геометрического ряда, коэффициент
                        R28
      R08=+07300039 " R39=аbc(R30)
      R09=+04394040 " R40=R39/R40 = 1. шип
      R10=+31300000 " IF R30<R00 THEN
      R11=+08400040 " R40=-R40
      R12=+22000000 " ENDIF
      R13=+26010283 " Запишет X прогр. в R33 и Z прогр. в R34
      R14=+01333333 " Для программирования диаметра: R33=R33+R33
(01333333) !
      R15=+00800085 " Запись R35,36=R30,31
      R16=+01332411 " R11=R33+R24
      R17=+01113030 " R30=R11+R30 окончательный размер
      R18=+01114011 " R11=R11+R40 (первая стружка)
      R19=+01342538 " R38=R34+R25
      R20=+01383131 " R31=R38+R31
      R21=+01022910 " R10=R02+R29 (количество повторов)

N50 G00 G98 XR11 " Перемещается в X на расстояние
      G27 R05=+00330009 " R09=R33
      R06=+00230039 " R39=R23 количество по одинаковой траектории
"N51/G73 L6 Q9998
"N52 M0
N6 G33 IR27 JR22 ZR31 " G33 Разка резьбы
N65 " Пустой блок должен быть
N7 G00 G24 XR09 ZR31 " Выезд в X
      G28 R05=+03402840 " R40=R40.R28 - умножение на коэффициент
      R06=+01114011 " R11=R11+R40 (следующая стружка)
      R07=+00340009 " R09=R34
N8 G00 ZR09 " Возврат в Z
      G73 L50 QR10 " Условный скачок на блок N50
N9 G00 G98 FR26 XR30 " Перемещается в X на последнее расстояние
      G27 R05=+00330009 " R09=R33
      R06=+02390139 " R39=R39-R01
N10 G33 IR27 JR22 ZR31 " G33 Резка резьбы на расстояние

```

```

N14                                " Должен находиться пустой блок
N11 G00          XR09      ZR31    " Возврат в X
      G27 R05=+00340009          " R09=R34
      R06=+16390010              " IF R39>=R00 THEN R10=73 ELSE R10=78
N12 G00 G24          ZR09          " Возврат в Z
      GR10 L9 Q9998              " Условный скачок
N13 G70                          " Конец

```

12.5.7 Снятие фаски под углом - G86

```

"-----
" КРОМКА ПОД УГЛОМ
"
" Используемые параметры:"
" R24 = Сколько мм до обрабатываемого полуфабриката в оси X начинает РС -
знак указывает направление (Xp)"
" R25 = Сколько мм до торцевой части в оси Z начинает РС - знак указывает
направление (Zp)"
" R26 = Склонность сдвига"
" R27 = не использовано"
" R28 = Толщина стружки в мм (перпендикулярно к оси Z) (Tx)"
" R29 = не использовано рабочее для (Tz)"
" R30 = Длина снятой фаски в оси X (знак указывает направление движения)
(Dx)"
" R31 = Длина снятой фаски в оси Z (знак указывает направление движения)
(Dz)"
""
" Знак является решительным только согласно параметрам R24 и R25 !!!"

N1 G79 L86
""
" Знак запишет в параметр R16 для X и R17 для Z"
""
N12 G26 R05=00280012 " Запишет толщину стружки в X до R12"
      R10=1           " Когда будет R10=0, потом будет конец"
      R16=0 R17=0
      G24             " ускорение коммуникации"
N30 G26 R05=16241615 " if R24<R16(=0) then R16=-1 (R15=73 или 78)"
N40 GR15 L60 R16=1.0
N50 R16=-1.0
N60 G26 R05=16251715 " if R25<R17(=0) then R17=-1 (R15=73 или 78)"
N70 GR15 L100 R17=1.0
N80 R17=-1.0
""
" Переведет все размеры для расчетов на положительные величины"
""
N100 G29 R05=07240024
      R06=07250025
      R07=07300030
      R08=07310031
N110 G26 R05=07280028
" "
" Пересчитает толщину стружки в оси:"
" Tz = (Tx . Dz)/Dx ... R29=(R28.R31)/R30"
N120 G29 R05=03283127 "R27=R28.R31 "
      R06=04273029    "R29=Tz (толщина в оси Z) "

" Пересчитает a = (Zp . Dx)/Dz .... R20 = (R25 . R30)/R31"
" и находится в R20"

```

```

R07=03253020      "R20=R25.R30"
R08=04203120      "R20=R20/R31"

" Пересчитает  $b = (X_p \cdot D_z) / D_x$  ....  $R_{21} = (R_{24} \cdot R_{31}) / R_{30}$ 
"  $b$  находится в R21"
N160 G29 R05=03243121      "R21=R24.R31"
      R06=04213021      "R21=R21/R30"

" Пересчитает  $A = a + X_p$        $R_{22} = R_{20} + R_{24}$ 
      R07=01202422

" Пересчитает  $B = b + Z_p$        $R_{23} = R_{21} + R_{25}$ 
      R08=01212523

" Рассчитает перемещение в X и Z (абсолютно) "
"  $X = A + T_x$        $R_{18} = R_{22} + R_{28}$ 
"  $Z = B + T_z$        $R_{19} = R_{23} + R_{29}$ 

N200 G29 R05=01222818
      R06=01232919
""
" Первый размер прироста с учетом знака "
" Получает их созданием знака указанного в R16 (X) и R17 (Z) "
      R07=03181618
      R08=03191719
      R15=-1.0      "ДЛЯ УМНОЖЕНИЯ -1"
      R09=2.0      "Для умножения в случае прогр. диаметра 2х"

" Перемещение приростом: "
""
N240 G91 G00 XR18 G26      R05=03181518 "заранее поменяет знак X.-1"
N250 G01 G26 XR18 ZR19 FR26 R05=03191519 "заранее поменяет знак Z.-1"
N260 G00      ZR19
      R0=0

" И опять находится в исходной точке "
" Прибавит стружку  $T_x = T_x + T_x$  ...  $R_{28} = R_{28} + R_{12}$ 
" "
" Тестирует R10"
" "
N264 G26 R5=19001001
N266 GR1 L1000      " СКАЧОК НА КОНЕЦ"
N270 G26 R05=01281228
" Если стружка  $T_x$  R28 больше длины снятой фаски  $D_x$  "
N280 G26 R05=15283011      " if R28 > R30 then R11=73 else R11=78"
N290 GR11 L310
N300 G73 L120 Q9998      " Нормальное прохождение"
N310 G26 R05=00300028      " Переместит  $D_x$  в  $T_x$ , т.е. R30 в R28 "
N315 R10=0      " Признак последнего прохождения "
N320 G73 L120 Q9998
N1000 G70
"

```

12.5.8 Резка резьбы на шаровой плоскости - G87

```

"
" -----
" РЕЗКА РЕЗЬБЫ НА ШАРОВОЙ ПЛОСКОСТИ - G87
"
" Используемые параметры:
" R22 = Подъем шара

```

```

" R23 = Количество повторов перемещения по окончательной траектории
" R24 = Сколько мм до обрабатываемого полуфабриката в оси X начинает РС -
знак определяет направление
" R25 = Сколько мм до торцевой части в оси Z начинает РС - знак определяет
направление
" R26 = Подъем резьбы
" R27 = Угловой сдвиг начала резки резьбы
" R28 = Во сколько раз следующая стружка меньше чем предыдущая (коэффициент
< 1)
" R29 = Количество стружек, которыми должна быть резьба вырезана
" R30 = Глубина резьбы в оси X - знак определяет направление
" R31 = Длина резьбы в оси Z - знак определяет направление
"
"
N1  G79  L87
N2  G90  G24  FR26          " Абсолютно
                                " R00=0
                                R00=+0
                                " R01=1
                                R01=+1
                                " R02=-2
                                R02=-2

                                G26  R05=+33060021          " Программа 06-21

                                R06=+23290039          " R39=реал(R29)
                                R07=+28283940          " R40=сумма геометрического ряда, коэффициент
                                                                R28
                                R08=+07300039          " R39=абс(R30)
                                R09=+04394040          " R40=R39/R40 = 1. шип
                                R10=+31300000          " IF R30<R00 THEN
                                R11=+08400040          "     R40=-R40
                                R12=+22000000          " ENDIF
                                R13=+26010283          " Запишет X прогр. в R33 и Z прогр. в R34
                                R14=+01333333          " Для программирования диаметра: R33=R33+R33
                                                                (01333333) !
                                R15=+01332411          " R11=R33+R24
                                R16=+01113030          " R30=R11+R30 окончательный размер
                                R17=+01114011          " R11=R11+R40 (первая стружка)
                                R18=+01342538          " R38=R34+R25
                                R19=+01383131          " R31=R38+R31
                                R20=+01022910          " R10=R02+R29 (количество повторов)
                                R21=+01112212          " R12=R11+R22

N50  G00                                XR11          " Перемещается в X на расстоянии
                                G27  R05=+00330009          " R09=R33
                                R06=+00230039          " R39=R23 количество по одинаковой траектории
N6   G33  IR27  XR12  ZR31          " Резка резьбы
N7   G00                                XR09          " Выезд в X
                                G29  R05=+03402840          " R40=R40.R28 - умножение на коэффициент
                                R06=+01114011          " R11=R11+R40 (следующая стружка)
                                R07=+00340009          " R09=R34
                                R08=+01112212          " R12=R11+R22
N8   G00                                ZR09          " Возврат в Z
                                G73  L50  QR10          " Условный скачок на блок N50
N9   G00  FR26          XR30          " Перемещается в X на последнем расстоянии
                                G28  R05=+00330009          " R09=R33
                                R06=+02390139          " R39=R39-R01
                                R07=+01302212          " R12=R30+R22
N10  G33  IR27  XR12  ZR31          " Резка резьбы на расстоянии
N11  G00                                XR09          " Возврат в X
                                G27  R05=+00340009          " R09=R34
                                R06=+15390010          " IF R39>R00 THEN R10=73 ELSE R10=78
N12  G00                                ZR09          " Возврат в Z

```

GR10 L9 Q9998	" Условный скачок
N13 G70	" Конец
"	

13

13. ПЕРЕСЧЕТ КООРДИНАТ

Пометка: Пересчет координат установлен от версии программного обеспечения 30.35 и 40.13. (26.3.2002 г.).

13.1 Общие принципы

Для расстановки различных пересчетов координат служит 11. группа G функций. При помощи пересчетов координат можно часть программы различным способом повернуть, уменьшить или сдвинуть в отношении оригинала.

Каждый пересчет эффективен немедленно после программирования и действителен до его аннуляции или до его собственного изменения. Все пересчеты относятся к актуальным сдвигам начала G53-G59 и коррекции длины и радиуса пересчетом не затронуты.

В каждом блоке можно запрограммировать только одну из функций 11. группы G, поэтому если необходимо использовать комбинацию различных видов пересчета, необходимо его запрограммировать в нескольких различных блоках. Все функции из 11. группы G имеют непрерывное воздействие и взаимно не влияют друг на друга, поэтому их можно комбинировать.

Каждое изменение пересчетов координат должно осуществляться самостоятельно в блоке программы и при отключенном радиусе коррекции. Коррекцию радиуса можно использовать для активного пересчета.

Каждая из функций служит для установки, а также для отключения собственного пересчета. Только функция G34 аннулирует все доступные пересчеты сразу, поэтому годится функцию G34 включить в приоритетный блок.

Все пересчеты координат автоматически отключаются центральной аннуляцией системы (без учета того, если функция G34 включена в приоритетный блок, или нет).

Пересчеты координат, описанные в этом разделе, не годится использовать для общей трансформации системы координат, потому что пересчеты осуществляются на уровне подготовки блока и пересчет не повлияет, например, на ручные сдвиги и т.п. Для общей трансформации будут в системе определены другие средства.

Перечень функций 11. группы G (состояние для версии 40.13):

G34	Основное состояние, все пересчеты координат отключены
G35	Отражение
G36	Изменение масштаба
G37	Независимый сдвиг

13.2 Отражение

Отражение приобретает форму зеркального отражения и программируется при помощи функции **G35**. При помощи отражения меняется направление в оси или в нескольких осях. Для отражения в оси системы меняется знак, по интерполяции плоскости случайно меняется G2 на G3 и знак функций I и J, и по плоскости коррекции меняется случайно G41 на G42.

Отражение будет выполнено с учетом действительной нулевой точки и актуального сдвига начала. Отражение также влияет на коррекцию длины и радиуса инструмента (ни сдвиг головки в случае двухголового токарного станка).

Если не будут применены остальные меры, меняется отражение одной оси при попутной обработке за непопутное, и наоборот.

Система позволяет ввести отражение по любой плоскости системы координат тем, что вводятся оси, которые должны переводить направление.

Программирование:

В самостоятельном блоке без движений запрограммируем **G35** и введем оси, которые должны переводить направление. Оси программируются с любой величиной, например 0 (X0, Y0, Z0, U0, V0, W0, A0, B0, C0).

Если в самостоятельном блоке запрограммирована самостоятельная функция **G35** без осей, отражение отменяется.

Действительное отражение действительно до изменения при помощи функции G35 или программированием G34 (отмена всех видов пересчета координат) или центральным обнулением системы.

Примеры программирования отражения:

G35	X0	"Переводит направление X	(отражение согласно оси Y)
G35	Y0	"Переводит направление Y	(отражение согласно оси X)
G35	X0 Y0	"Переводит направление X и Y	(отражение согласно центру X, Y)
G35	Y0 U0 C0	"Переводит направление Y, U и C	(отражение согласно центра Y, U и C)
G35		"Аннуляция отражения	

Пример для отражения:

"В примере повторяется 4 раза вызов одинаковой подпрограммы с различным введенным" отражением:

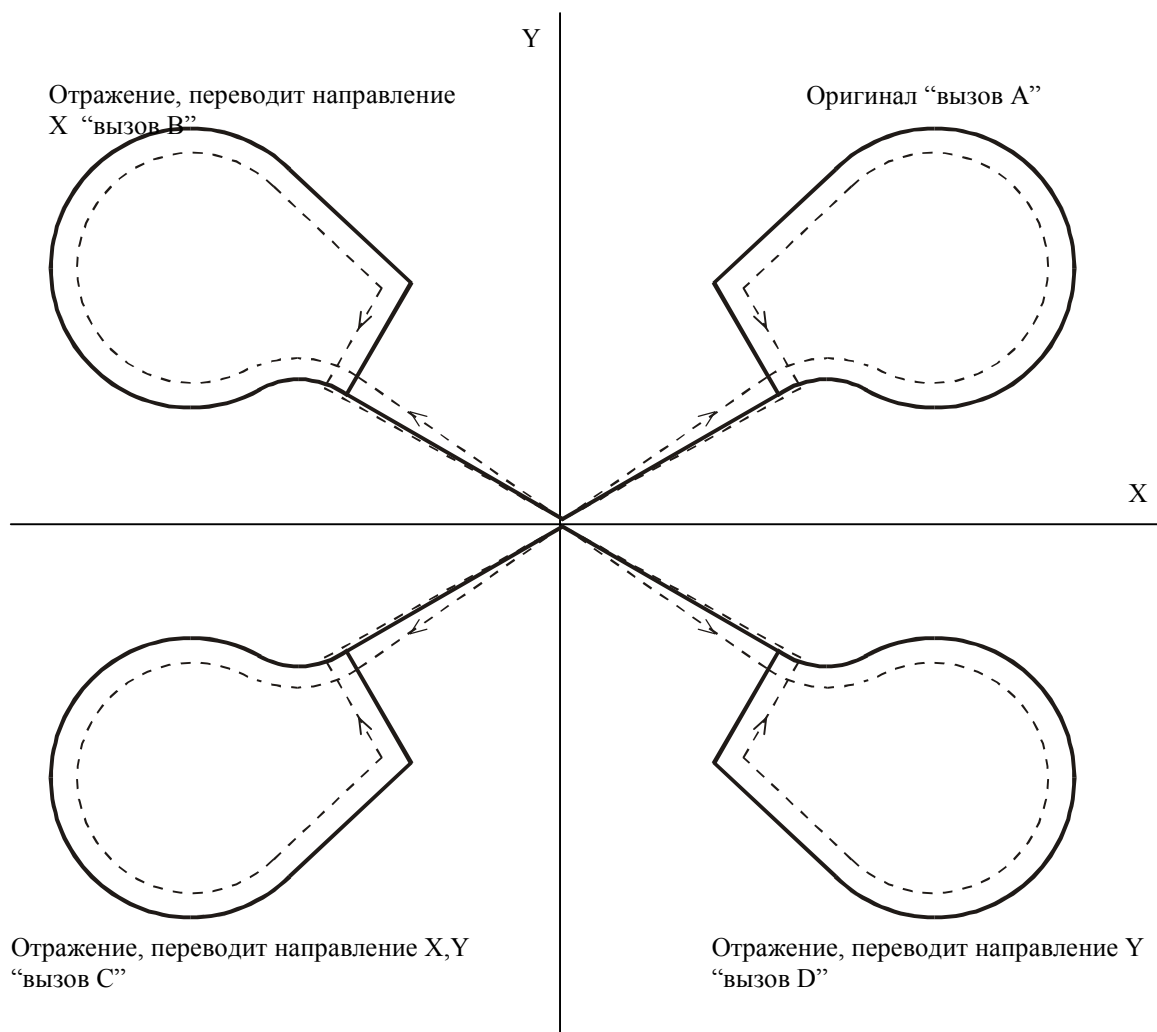
```
%1
N10 X0 Y0 F5000 G23
N20 G71 L100      "Вызов подпрограммы - A
N30 G35 X0        "Отражение (переводит направление X)
N40 G71 L100      "Вызов подпрограммы - B
N50 G35 X0 Y0     "Отражение (переводит направление X, Y)
N60 G71 L100      "Вызов подпрограммы - C
N70 G35 Y0        "Отражение (переводит направление Y)
N80 G71 L100      "Вызов подпрограммы - D
N90 G35           "Отмена отражения
N99 M30
```

“Подпрограмма с коррекцией радиуса

```

N1000 G79 L100      “Подпрограмма L100
N1010 G01 X0 Y0
N1020 G01 X30 Y20 G41 D3
N1030 G02 X50 Y24.849 I16.641 J-24.962
N1040 G03 X30 Y73.484 I3.359 J29.811
N1050 G01 X20 Y61.075
N1060 G01 X30 Y20
N1070 G01 X0 Y0 G40
N1080 G70

```



13.3 Изменение масштаба

Изменением масштаба получается геометрически увеличенная или уменьшенная форма и программируется при помощи функции **G36**. Изменение масштаба можно создать от одного оригинала увеличенное или уменьшенное отражение.

Изменение масштаба выполняется с учетом действительной нулевой точки к актуальному сдвигу начала. Изменением масштаба также не влияем на коррекцию длины и радиуса инструмента (ни сдвига головок в случае двухголового токарного станка).

Масштаб для увеличения и уменьшения может вводиться или прямо величиной, как для введения координат с точностью на 3 десятичные числа, так при помощи дроби, где числитель и знаменатель это величины с точностью на 3 десятичные числа.

Программирование:

В самостоятельном блоке без движений запрограммированы **G36** и вводятся оси, которые определяют отношение увеличения или уменьшения.

- Если будет запрограммирована только первая координата (X), масштаб определен прямо его величиной.
- Если запрограммирована первая координата (X) и одновременно вторая координата (Y), масштаб определяется величиной дроби, где числитель является величиной первой координаты и знаменатель является величиной второй координаты.

Если в самостоятельном блоке запрограммирована самостоятельно функция **G36** без осей, изменение масштаба отменяется.

Эффективное изменение масштаба действительно до изменения при помощи функции **G36** или программирования **G34** (отмена всех видов пересчетов координат), или центральным обнулением системы.

Примеры программирования изменения масштаба:

G36	X1.27	"Изменение масштаба, увеличение 1.27
G36	Y0.38	"Изменение масштаба, уменьшение 0.38
G36	X1.54 Y5.28	"Изменение масштаба, уменьшение $1.54/5.28 = 0.2916$
G36		"Отмена изменения масштаба

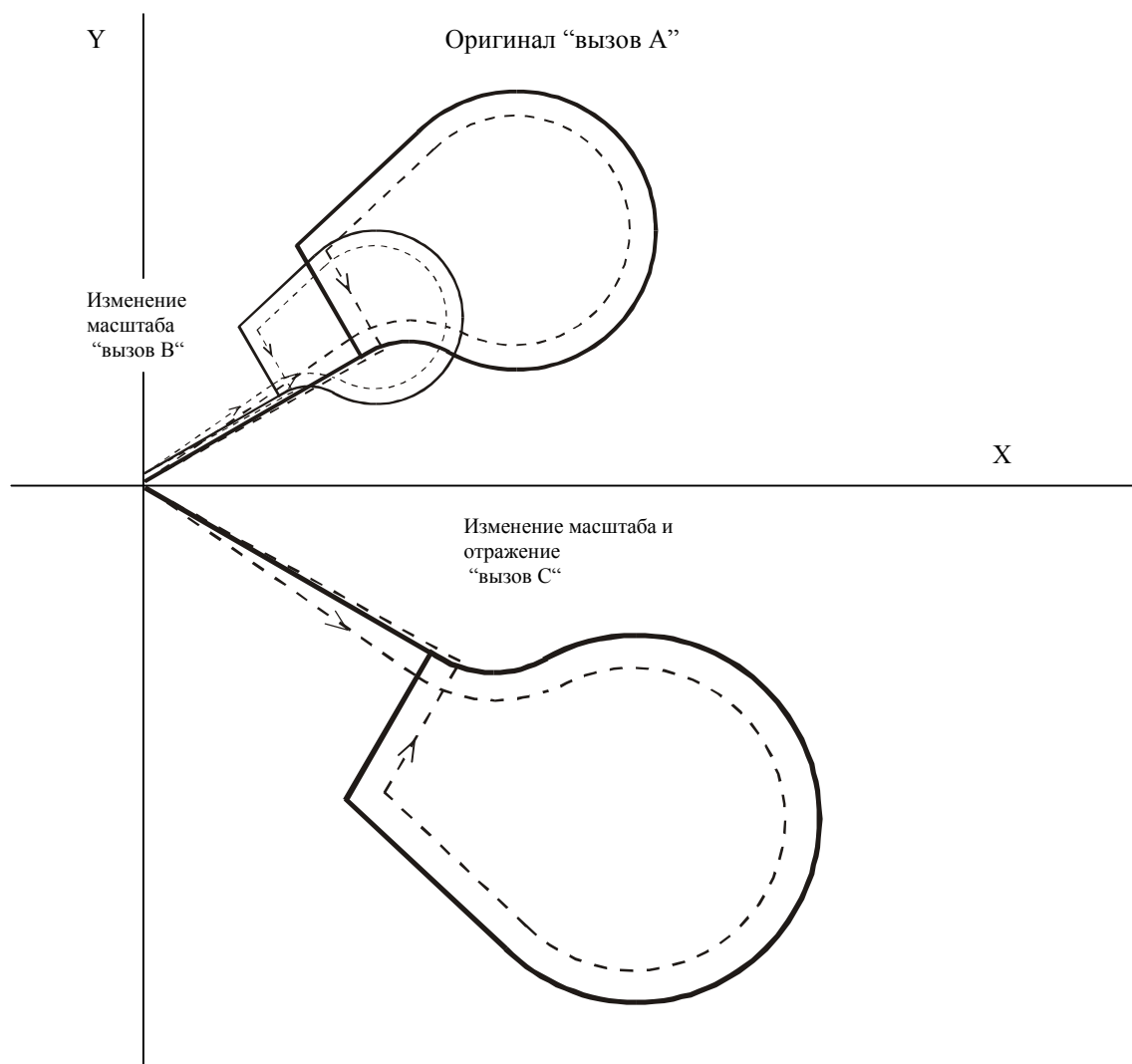
Пример для изменения масштаба:

"В примере повторяется 3х вызов одинаковой подпрограммы. Во второй вызов «вводится изменение масштаба», в третьем вызове кроме изменения масштаба, вводится также «отражение» (Подпрограмма вызывается с различной коррекцией радиуса)

%1	
N10	X0 Y0 F5000 G23
N20	G71 L100 D3 "Вызов подпрограммы - А
N30	G36 X1.54 Y3.28 "Изменение масштаба уменьшения $1.54/3.28 = 0.46951$
N40	G71 L100 D4 "Вызов подпрограммы - В
N50	G35 Y0 "Отражение (переводит направление Y)
N50	G36 X1.27 "Изменение масштаба увеличения 1.27
N60	G71 L100 D5 "Вызов подпрограммы - С
N90	G35 "Отмена отражения
N95	G36 "Отмена изменения масштаба
N99	M30

“Подпрограмма с коррекцией радиуса

```
N1000 G79 L100      “Подпрограмма L100
N1010 G01 X0 Y0
N1020 G01 X30 Y20 G41
N1030 G02 X50 Y24.849 I16.641 J-24.962
N1040 G03 X30 Y73.484 I3.359 J29.811
N1050 G01 X20 Y61.075
N1060 G01 X30 Y20
N1070 G01 X0 Y0 G40
N1080 G70
```



13.4 Независимый сдвиг

Независимым сдвигом получаем сдвинутую форму и программируем его при помощи функции **G37**. Независимый сдвиг не зависит от актуального сдвига начала G53 – G59 и проявляется как следующая дополнительная составная часть сдвига траектории.

Эффект независимого сдвига одинаковый как сдвиг траектории при помощи G53 - G59, только сдвиг при помощи таблицы сдвига не вводится, а вводится прямо в программе при помощи функций G37.

На изменение сдвига также не влияет коррекция длины и радиуса инструмента (ни сдвига головок в случае двухголового токарного станка).

Величина независимого сдвига вводится прямо во время программирования функции G37 величиной в отдельных координатах, которые представляют собой вектор сдвига с учетом нулевой точки станка.

Программирование:

В самом блоке без движения программируется **G37**, и вводятся оси, которые определяют размер независимого сдвига.

Когда в самом блоке запрограммирована сама функция **G37** без осей, независимый сдвиг отменяется.

Действительный сдвиг действителен до изменения при помощи функции G37 или программированием G34 (отмена всех видов пересчетов координат), или же центральным обнулением системы.

Примеры программирования независимого сдвига:

```
G37 X10.127      "Сдвиг оси X на 10.127
G37 X-0.138 Y12  "Сдвиг оси X на -0.138 и Y на 12.0
G37              "Отмена независимого сдвига
```

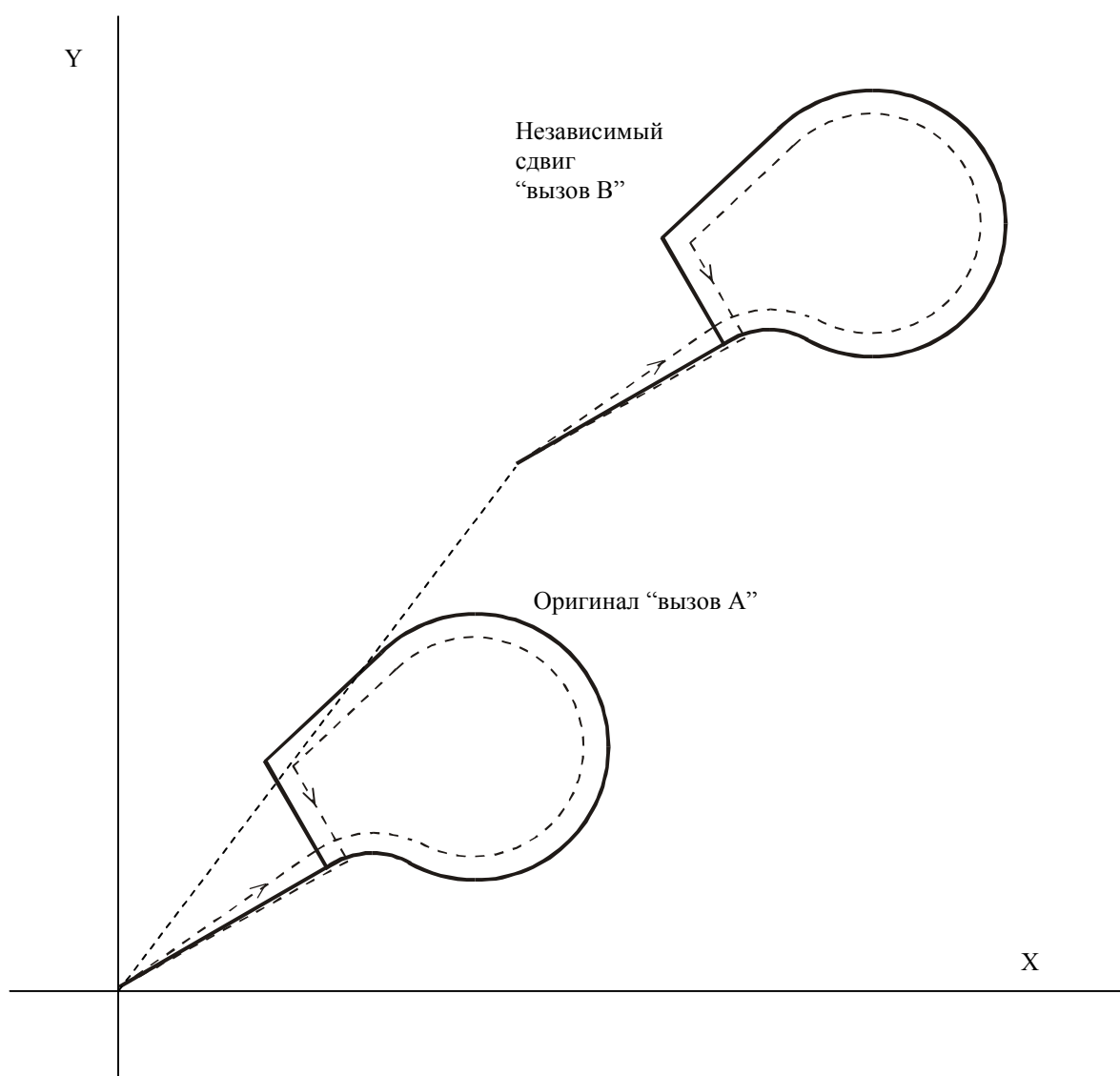
Пример для независимого сдвига:

"В примере повторяется 2 раза вызов одинаковой подпрограммы. Во втором вызове «вводится сдвиг»

```
%1
N10 X0 Y0 F5000 G23
N20 G71 L100 D3      "Вызов подпрограммы - А
N30 G37 X80 Y120     "Независимый сдвиг на (80,120)
N40 G71 L100 D4      "Вызов подпрограммы - В
N90 G37              "Отмена сдвига
N99 M30
```

"Подпрограмма с коррекцией длины

```
N1000 G79 L100      "Подпрограмма L100
N1010 G01 X0 Y0
N1020 G01 X30 Y20 G41 D3
N1030 G02 X50 Y24.849 I16.641 J-24.962
N1040 G03 X30 Y73.484 I3.359 J29.811
N1050 G01 X20 Y61.075
N1060 G01 X30 Y20
N1070 G01 X0 Y0 G40
N1080 G70
```





Приложение А – плавная связь между блоками

Плавность

Системы CNC8x6 с процессором в кассете CPU04 (80486) и системы серии CNC8x9 (DUAL) позволяют плавную обработку сложных форм, которые запрограммированы, например, при помощи большого количества малых блоков. Под названием **плавность** понимает такое свойство системы, когда система переместится без понижения скорости из одного блока в следующий блок движения. Для плавного перемещения между блоками не является необходимым условием, чтобы блоки плавно продолжали друг друга его касательной (директивы движения обоих блоков не должны быть одинаковыми).

Комплексная скорость

Системы серии CNC8x9 (DUAL) более этого позволяют использовать управление плавностью т.наз. «**комплексной скоростью**». Под названием «комплексная скорость» понимает такое свойство системы, когда система перемещается **плавно**, но при этом пытается достичь запрограммированной скорости даже во время прохождения нескольких блоков. Таким образом, предоставлена возможность, что также блоки малой длины проходятся с многократно большей скоростью, чем бы они проходились без управления комплексной скоростью. Из этого вытекает, что система должна предполагать и корректировать скорость на основании определенных критерий, которые вытекают из следующих блоков. Анализ критерий по следующим блокам осуществляется макс. 500 блоков вперед. Проблематика комплексной скорости описана в этом разделе ниже.

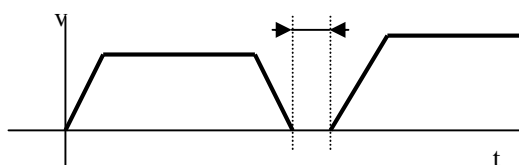
В следующем тексте будут описаны возможности проблематики плавности.

Приложение А1 – соединение блоков без плавной связи

Стандартный режим связи блоков выбирается функцией **G98**. Речь идет о неявной установке после включения системы. В этом случае движение в каждом блоке начинается и заканчивается рампой. Крутизна разбега и остановки рампы определена машинной константой R52.

В системе не происходит подготовка следующего блока подпрограммы вперед, и поэтому возникает между блоками определенная задержка времени, ограниченная мощностью технического комплекта системы.

Стандартный режим соединения блоков не годится использовать для обработки с большими требованиями к качеству поверхности.



G98

Приложение A1 – плотное соединение блоков

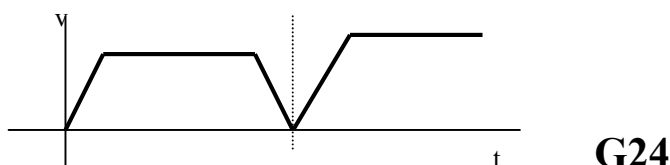
Плотное соединение блоков включается при помощи непрерывной функции **G24**. В этом случае передается с пульта системы следующий блок в кассеты (или во вторичный процессор в системах CNC8x9-DUAL) уже в течение предшествующего блока. Движение в каждом блоке начинается и заканчивается рампой, и после окончания рампы остановки система ждет остановки минимального отклонения согласно машинным константам R06 – R07. Если система оценит остановку минимального отклонения, немедленно начнет выполнять следующий блок.

Условием плотного соединения блоков является минимальное время длительности одного блока. Если соблюдается минимальное время длительности блока и запрограммирована функция G24, система плотно присоединит следующий блок.

Минимальное время длительности блока движения составляет для плотного соединения блоков:

30 мс для систем серии CNC8x6
2 мс для систем серии CNC8x9

Плавность движения в этом случае очень сильно зависит от динамики сервоприводов и от крутизны рампы разбега и остановки.



Приложение A1 – ручное управление плавным соединением

Необходимым условием ручного управления плавным соединением является установка 1. декады машинной константы R97 на величину 1.

1. декада машинной константы R97:

- 0** переход между блоками без расчета остатков (не используется)
- 1** ручное управление плавным соединением функций: G23 – G24
- 2** автоматическое отсоединение плавного соединения.

Плавным соединением блоков во время ручного управления управляют функции **G23** и **G24**. Блоки движения с рабочим сдвигом, в которых запрограммирована функция G23, будут следовать друг за другом плавно. Это означает, что система в этом случае не управляется никакой рампой и не ждет остановки минимального отклонения. Более этого система рассчитывает остатки последней траектории в такте блока равномерно в нескольких тактах разбега, так что, если разница в касательных небольшая (директива движения обоих блоков), будет переезд выполнен без изменения скорости. Большая разница в касательных соединенных блоков проявляется определенным изменением скорости переезда, которое, однако, очень мало и в этом случае вообще не является значительным. В общем можно сказать, что чем меньше будет разница в касательных обоих соединенных блоков, тем будет переход более плавным.

Необходимым условием для плавного соединения является соблюдение минимального времени длительности блока для плавного перемещения.

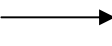
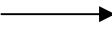
Минимальное время длительности блока для плавного перемещения является:

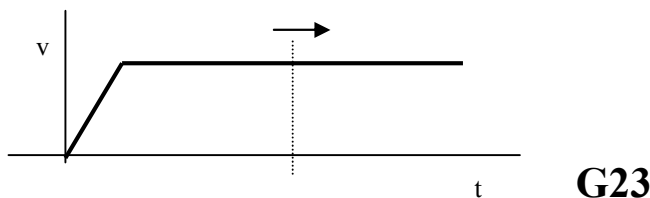
60 мс для систем серии CNC8x6
4 мс для систем серии CNC8x9

Система во время плавного перемещения (без активной комплексной скорости) проверяет только траекторию остановки следующего блока, что обозначает, что в случае необходимости тормозит максимально один блок вперед. Поэтому необходимо еще обеспечить, чтобы отмена плавного перемещения не находилась в очень коротком блоке, который следует после плавных блоков с большей запрограммированной скоростью.

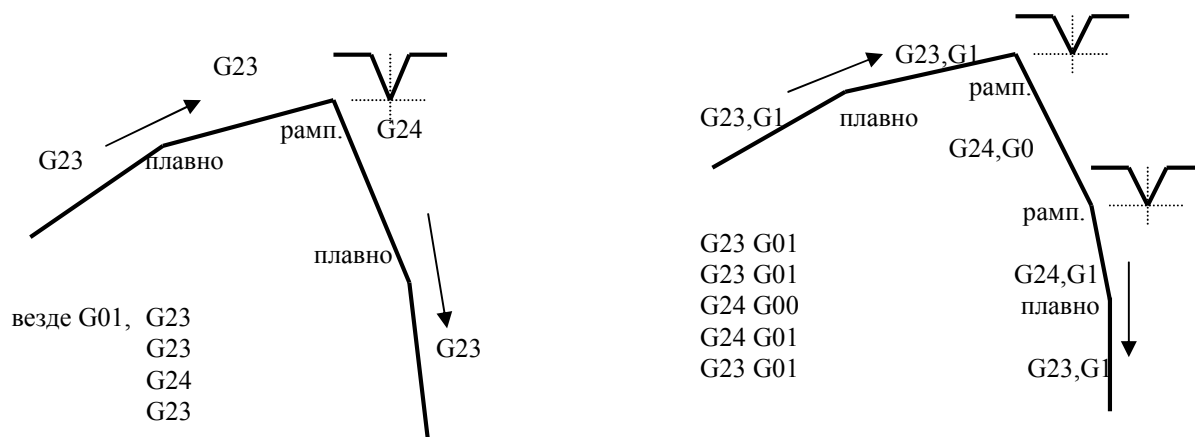
Во время, когда система управляет движением при помощи плавного соединения, определяет ускорение (рампу) машинная константа **R236**. Если система не управляется плавным соединением и запрограммирован рабочий сдвиг, ускорение определяет машинная константа R52. Для быстрого сдвига применяется машинная константа R237.

Оценка плавного соединения в двух по себе следующих блоках:

Блок n-1	Блок n	Знак плавности	Описание плавности
G24	G24		Нет плавного соединения
G23	G24		Нет плавного соединения
G24	G23 (G01,G02,G03)		Есть плавное соединение
G23	G23		Есть плавное соединение



Примеры использования ручного управления плавного соединения блоков:



Приложение A1 – автоматическое определение плавного соединения

Необходимым условием автоматического управления плавной связи устанавливается в 1. декаде машинной константы R97 на величину 2. В последней версии программного обеспечения можно автоматически определять использование для первых 3 координат.

Автоматическое определение плавного соединения блоков включается при помощи функции **G23**. Если 1. декада машинной константы R97 установлена на величину 2, имеет функция G23 другое значение, чем в ручном управлении плавным соединением. В этом случае **система сама определяет возможность плавного соединения**, и сама также ее включает или отменяет. В большинстве случаев достаточно, если будет в начале программы указана функция G23, которая является плавной и та будет действительна для всей программы. В следующем тексте будет описано, какие критерии во время определения система использует.

Автоматическое определение плавного соединения отменяется при помощи функции **G24**. В этом случае система полностью отменяет плавное соединение (даже с автоматическим определением) и будет управлять движением в плотном соединении блоков с включением рампы разезда и остановки. Действительно все, что было написано в части „Плотное соединение блоков.“ (Функция G24 в случае автоматического определения используется только исключительно.)

G23	Включение автоматического определения плавного соединения, будет зависеть от системы, если она определит плавное прохождение блока.
G24	Отмена плавного соединения, система управляет движением в тесном соединении блоков с включением рампы разезда и остановки. (Функция в случае автоматического определения используется только исключительно.)

В случае если система определяет плавное соединение и включит его, блоки будут переходить друг в друга плавно (аналогично как в случае G23 во время ручного управления плавным соединением). Это означает, что система в этом случае не включит никакую рампу и не ждет остановки минимального отклонения. Более этого система рассчитывает остатки последней траектории в этом такте блока, равномерно в нескольких тактах переезда, так что если разница в касательных небольшая (директива движения обоих блоков), будет переезд без изменений скорости. Большая разница в касательных связанных блоков проявится определенным изменением скорости переезда, которое, однако, очень мало и в этом случае вообще не важно. В общем можно сказать, что чем будет разница касательных обоих связанных блоков больше, тем будет переход более плавным.

Необходимым условием для плавного соединения является соблюдение минимальной длительности блока для плавного перемещения.

Минимальная длительность блока для плавного перемещения составляет:

60 мс	для систем серии CNC8x6
4 мс	для систем серии CNC8x9

Система в случае плавного перемещения (без активной комплексной скорости) проверяет только тормозную траекторию следующего блока, что означает, что в случае надобности тормозит максимально один блок заранее. Из-за этого необходимо еще обеспечить, чтобы не было отменено плавное перемещение за очень короткий блок, который следует после плавных блоков с большей запрограммированной скоростью. (Если это требование нельзя обеспечить, необходимо активировать комплексную скорость).

Угол допуска для автоматического определения

В машинную константу **R39** вводится угол допуска для автоматического определения. Он вводится с точностью на одну тысячную градуса, так что десятичный знак действителен и данные указаны в градусах. Система определит угол между актуальными направляющими движения по месту соединения

блоков (угол между касательными движения) и сравнивает их с углом допуска, введенным в машинной константе R39. Если угол между направляющими движения меньше, чем угол допуска, может произойти плавное соединение блоков. Система предотвратит плавные переезды, если запрограммирован быстрый сдвиг G0.

Во время, когда система управляет движением при помощи плавного соединения, определяет ускорение (рампу) в машинной константе **R236**. Если система не управляет плавным соединением и запрограммирован рабочий сдвиг, определяет ускорение машинная константа R52. Для быстрого сдвига применяется машинная константа R237.

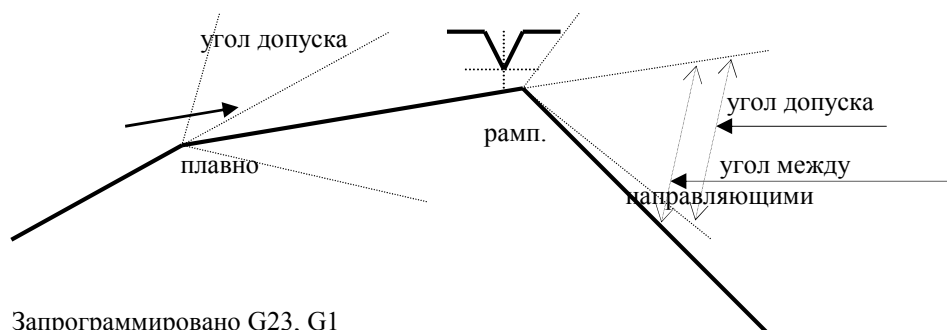
Условия, необходимые для определения включения плавного соединения

Запрограммирована G23 и 1. декада машинной константы R97 имеет значение 2.

Угол между направляющими движения обеих блоков допустим.

Предшествующий и актуальный блок являются подвижными и запрограммирована одна из функций G1, G2, G3.

Длительность блоков не меньше, чем минимальная длительность блока для плавного перемещения.



Приложение A1 – графическая диагностика плавного соединения

Диагностика плавного соединения служит для определения спорных мест в программе в точке плавного перемещения. Для целей диагностики плавности в систему установлены два графические формата. Остальное описание действительно для программной версии пульта 5.033, 6.030 и больше.

Графическое изображение программы

Активация графической программы в системе разрешается в файле **CNC836.KNF** в параметре **\$57**, который должен быть установлен на величину 2 (два прохода).

Графическое изображение программы активируется кнопкой **GRAF** на пульте системы (или при помощи выбора индикации WIN).

После выбора программы в режиме AUT имеет кнопка GRAF функции, которые циклически меняются по 3 нажатиях:

Функция графического изображения (кнопка GRAF):

1. нажатие кнопки GRAF	Система запомнит актуальное состояние форматов в режиме AUT и в первой части экрана нарисует графическое изображение выбранной программы. Изображение управляется машинной константой R387 , в которую вводится количество блоков для изображения. Если константа является R327 нулевой, будет изображена вся программа (если программа состоит, например из 1000000 блоков, может таким образом перечеркнуть весь экран и форма не будет разборчива). Если машинная константа R387 установлена, например, на величину 50000, изобразится только первых 50000 блоков программы. Во время нажатия кнопки ENTER экран сотрется и изобразится следующих 50000 блоков.	
2. нажатие кнопки GRAF	Изобразится самостоятельное MENU, которое относится к графическому изображению программ. Если функции, которые принадлежат к этому меню, нам не надо использовать, продолжаем 3. нажатием кнопки GRAF.	
	LUPA (лупа)	После нажатия изобразится ориентировочный крест, который при помощи курсорных кнопок установим на место на графике, которое необходимо увеличить. Кнопкой ENTER (ввод) постепенно переключаем режим увеличения (ZOOM) и режим перемещения (SHIFT). Для управления увеличением и сдвигом служат курсорные кнопки. После 1. нажатия кнопки LUPA на графической траектории изобразятся желтые и красные точки на переходе блоков. Желтые точки обозначают плавное перемещение блоков. Красная точка символизирует нарушение плавного перемещения блоков.
	KOREKCE (коррекция)	Изобразит траекторию с коррекцией радиуса (эквидистанту) синим цветом.
	PLYNULOST (плавность)	В блоках, где плавность нарушена, (красные точки на переходе блоков) изобразятся соответствующие номера блоков.
	ROVINA (плоскость)	Выбор плоскости изображения X-Y, X-Z, Y-Z (должны быть соответствующим способом установлены машинные константы R340 и R341 для плоскости коррекций).
3. нажатие кнопки GRAF	Возобновит первоначальное состояние форматов и первоначальное MENU до 1. нажатия кнопки GRAF.	

Для контроля действительного хода скорости можно использовать формат **контроль прохождения скорости**. При помощи выбора индикации WIN выбирается формат контроля прохождения скорости и выбирается контроль блоков N (вместо координат X,Z,...). После калибровки можем контролировать ход окончательной пространственной скорости в реальном времени, и при помощи белых вертикальных линий могут быть изображены отдельные блоки.

Приложение A1 – комплексная скорость

Системы серии CNC8x9 (DUAL) от версии программного обеспечения 40.10 пульта и 6.030 вторичного процессора, позволяют для управления плавностью использовать т.наз. „**комплексную скорость**“. Под термином „комплексная скорость“ понимаем такое свойство системы, когда система движется **плавно** и при этом пытается достичь запрограммированной скорости также в течение большего числа блоков. Таким образом, разрешено, что также блоки с малой длительностью проходят с многократно большей скоростью, чем бы они проходили без управления комплектной скоростью. Из этого вытекает, что система должна предполагать и корректировать скорость на основании определенных критериев, которые вытекают из следующих блоков. Анализ критериев из следующих блоков происходит макс. 500 блоков вперед.

Комплексная скорость позволит на станке использовать **высокоскоростную обработку**, хотя программа составлена из большого количества малых блоков. В случае больших скоростей обработки необходимо компенсировать регуляционное отклонение связи с положением для достижения требуемой точности обработки. Компенсация регуляционного отклонения до нулевой величины, и это также в случае динамических состояний станка, позволяет подходящую установку параметров сервопетли для **feedforward** с производной (см. Инструкцию по программированию PLC: Установка параметров

сервоприводов, или машинных констант R356 - R381).

Критерии для управления скоростью из следующих блоков

- **Изменение установленной скорости.** Ограничение скорости с достаточным опережением, если в следующих блоках запрограммирована меньшая скорость.
- **Конец плавности.** Конец отдела плавного движения, когда пространственный угол связи двух по себе следующих блоков больше, чем разрешенный предел. Например, система с опережением тормозит в случае перехода на другую горизонталь во время пространственной обработки.
- **Критерий для круговой интерполяции.** Критерий центробежного ускорения и точности генерации траектории для круговой интерполяции. Система с опережением ограничит скорость таким способом, чтобы соблюдались введенные параметры для круговой интерполяции.
- **Динамический критерий.** Принцип основан на предположении, что система приводов со всей механикой станка обладает определенной максимальной разрешенной перегрузкой. На основании запрограммированной скорости и угла связи блоков скорость корректируется с достаточным опережением таким способом, чтобы не превысит эту максимальную разрешенную перегрузку.
- **Критерий точности.** Система основана на запрограммированной скорости и углу связи блоков с достаточным опережением ограничивает скорость таким способом, чтобы отклонение являлось от идеальной траектории меньше, чем установленный предел.
- **Минимальное время длительности блока.** Система контролирует действительное время длительности блока и в случае необходимости с достаточным опережением ограничит скорость таким способом, чтобы длительность блока была больше или равна минимальному времени длительности блока для плавного перемещения (4 мс).

Активация комплексной скорости

Необходимым условием установки для двухходовой записи программы в запоминающее устройство (в CNC836.KNF будет установлено \$57 на величину 2) и 2. декады машинной константы R338 на величину 1.

Во время, когда система управляет движением при помощи плавного соединения, определяет ускорение (рампу) машинная константа **R236**.

Если система не управляет плавным соединением и запрограммирован рабочий сдвиг, определяет ускорение машинная константа R52. Для быстрого сдвига применяется машинная константа R237. Рекомендуем установить рампу R236 меньше, чем величина параметра R52.

Перечень всех параметров для активации и деятельности комплексной скорости:

Параметр	Пример	Описание
CNC836.KNF: \$57 = 2	2	Двухходовая запись программы. Разрешение двухходовой записи программы в запоминающее устройство во время выбора программы. Второе прохождение записи позволяет анализ критериев для плавного перемещения (кроме этого также графическое изображение траектории с коррекцией радиуса).
2. декада R338 = 1	1	Разрешение комплексной скорости. Предположение для коррекции скорости макс. 500 блоков вперед.
R39	60.000	Угол допуска для плавного соединения. Константа указывает угол касательных в конечной точке предшествующего блока и актуального блока. Если запрограммирована функция G23, рабочий сдвиг и угол касательных меньше, чем введенный предел, считается

		траектория плавной и скорость может корректироваться на основании остальных критериев. Если угол касательных больше, чем введенный предел, будет скорость в конце предшествующего блока понижена до нулевой (проверка минимального отклонения) и после этого разъезд во время следующего блока. Размер угла в R39 рекомендуется установить на большую величину (например, 60 градусов), потому что остальные критерии коррекции скорости могут соответствующим способом урегулировать скорость согласно актуальному углу связи блоков, и общее движение потом является более плавным.
1. декада R97 = 2	2	Автоматическое определение плавной связи. Включается при помощи функции G23, когда система сама тестирует пространственный угол между направляющими по месту соединения блоков, сравнивает его с углом допуска, введенным в машинной константе R39, и принимает решение о плавности.
R236	400	Ускорение для плавной связи. Константа R236 определяет ускорение (рампу) во время, когда система управляет движением при помощи плавного соединения. Если в константе R236 указана нулевая величина, ускорение для плавного соединения будет принято от машинной константы R52. (Рекомендуем установить меньшую величину хотя бы на 100, чем установленная в константе R52.)
1. - 4. декады R232	0000	Геометрический критерий для круговой интерполяции. Система ограничивает скорость в случае круговой интерполяции таким способом, чтобы отклонение от идеального круга была меньше, чем предел. (См. Приложение F – машинной константы системы CNC8x6 CNC8x9.)
5. - 8. декады R232	0040	Динамический критерий для круговой интерполяции. Система ограничивает скорость в случае круговой интерполяции таким способом, чтобы не была превышена максимальная разрешенная перегрузка станка. (См. Приложение F – машинные константы системы CNC8x6 CNC8x9.)
R382	0	Длина блока для отмены плавного соединения. Параметр R382 определяет предел длины блока в случае плавного соединения блоков G23. Если длина блока меньше, чем введенный предел, будет на этом месте плавное соединение прервано. Знак минус заблокирует активацию предела. Параметр может применяться для комплексной скорости, если отмена плавного перемещения для очень малых блоков может иметь более оптимальное прохождение по времени.
R383	0.005	Длина блока в случае плавного соединения для отмены блока. Если длина блока меньше, чем введенный предел, потом этот блок соединен с соседним блоком. Знак минус заблокирует активацию предела. Параметр может применяться для комплексной скорости, когда отмена очень малых блоков слишком замедляет обработку. Очень малые блоки (несколько микрометров) могут быть генерированы некоторыми предлагаемыми программами CAD. Соединение блоков разрешено только для линейного движения и только тогда, если в блоке не запрограммированы никакие технологические функции. Программа не должна менять скорость и не должна быть запрограммирована функция G91.
1. – 4. декады R384	0.008	Ограничение скорости – критерий точности. Система ограничивает скорость во время плавного соединения и

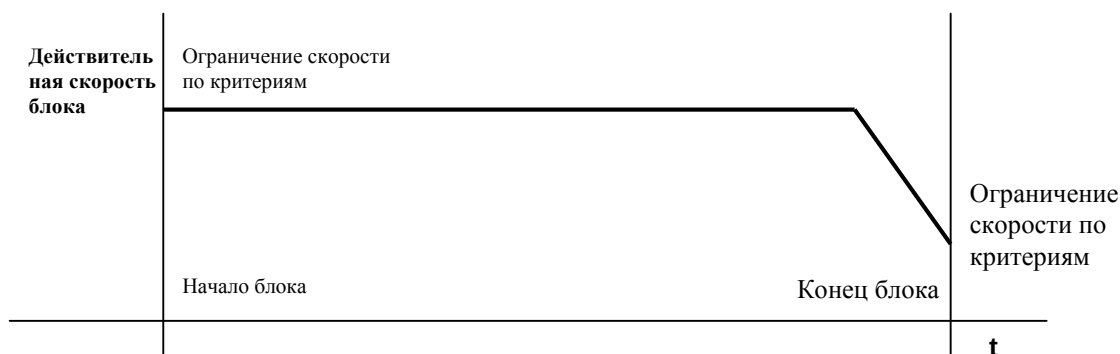
		в случае активной комплексной скорости таким способом, чтобы было отклонение от идеальной траектории меньше, чем введенный предел в параметре R384. Величина вводится в микрометрах. Максимальная скорость с точки зрения требуемой точности зависит от угла соединения соседних блоков. (См. ниже)
5. – 6. декады R384	20	Процент понижения скорости для графического изображения. Графическое изображение программы можно использовать для диагностики плавности. После 1. нажатия кнопки LUPA на графическом изображении изобразятся желтые и красные точки перехода блоков. Желтые точки обозначают плавное перемещение блоков. Красные точки символизируют понижение скорости ниже процента от запрограммированной скорости, который вводится в 5. - 6. декадах R384. Например, для величины 20 в графическом изображении появятся красные точки в случае понижения скорости ниже 20 процентов от запрограммированной скорости.
1. – 7. декады R385	10.000	Ограничение скорости – динамический критерий. Система ограничивает скорость в случае плавного соединения и во время активной комплексной скорости таким способом, чтобы не была превышена разрешенная перегрузка станка, указанная в параметре R385. Величина вводится в мм/с². Максимальная скорость с точки зрения требуемой перегрузки зависит от угла соединения соседних блоков. Установка параметра узко связана с динамикой приводов. Принцип основан на предположении, что система приводов со всей механикой станка обладает определенной максимальной разрешенной перегрузкой и система ограничит скорость таким способом, чтобы она не была превышена. (См. ниже.)
8. декада R385	1	Коррекция динамического критерия на длину блока. В восьмой декаде параметра R385 может активироваться коррекция критерия перегрузки на длину блока, причем для правильной функции необходимо ввести величину согласно используемой длине блоков во время установки динамического критерия (референционная длина): 0 Коррекция динамического критерия не активирована 1 Референционная длина блока составляет 1 мм 2 Референционная длина блока составляет 0.5 мм 3 Референционная длина блока составляет 0.1 мм 4 Референционная длина блока составляет 0.05 мм
R386	10.000	Длина блока для повышения скорости. Если длина блока меньше, чем введенный предел в машинной константе R386, система сохранит в рамках блока скорости, рассчитанные по динамическому критерию и критерию точности начала и конца блока. Система сохраняет переходные скорости во всем блоке. Если длина блока больше, чем введенный предел в R386, разрешено повышение скорости максимально на запрограммированную скорость и повторное повышение скорости в рамках одного блока. Система поэтому сохраняет переходные скорости, рассчитанные по динамическому критерию и критерию точности только на начало и конец блока. Величина ноль в R386 разрешит повышение скорости для всех блоков. (См. ниже.)
R387	50.000	Количество блоков для графического изображения. Напечатание графического изображения руководствуется

		машинной константой R387, в которую вводится количество блоков для изображения. Если константа R327 является нулевой, изобразится вся программа (если имеет программа, например 1000000 блоков, может таким образом разрисовать весь экран и форма будет очень мало разборчива). Если машинная константа R387 установлена, например, на величину 50000, изобразятся только первых 50000 блоков программы. Во время нажатия кнопки ENTER экран сотрется и нарисуеться следующих 50000 блоков.
--	--	--

Приложение A1.1 – длина блоков для повышения скорости, параметр R386.

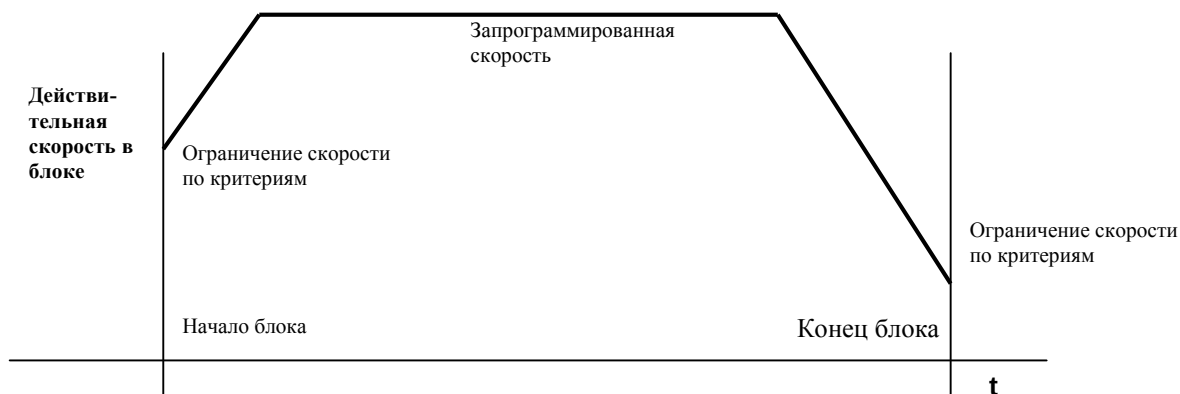
Если длина блока меньше, чем введенный предел в машинной константе R386, система сохраняет в рамках блока скорости, рассчитанные по динамическому критерию и критерию точности в начале и в конце блока. Это может быть невыгодно в случае длительных блоков, которые соединены большим углом соединения, потому что соблюдение переходных скоростей по установленным критериям, будут действительны в течение всего блока. Следующие блоки потом будут проходить зря медленно.

Пример сохранения скорости в рамках блока для длины блока меньше, чем предел в R386:



Если длина блока больше, чем введенный предел в R386, разрешен рост скорости максимально на запрограммированную скорость и обратное понижение скорости в рамках одного блока. Система поэтому сохраняет переходные скорости, рассчитанные согласно динамическому критерию и критерию точности только в начале и в конце блока. Подходящей установкой константы R386 может произойти основательное ускорение программы.

Пример для разрешенного повышения скорости в рамках блока для блока длиной больше, чем предел в R386:



Приложение A1.1 – критерий точности, параметр R384

Система ограничивает с достаточным опережением скорость во время плавного соединения и в случае активной комплексной скорости таким способом, чтобы отклонение от идеальной траектории было меньше, чем введенный предел в параметре **R384**. Величина вводится в микрометрах.

Максимальная разрешенная скорость с точки зрения требуемой точности, зависит от угла соединения соседних блоков согласно отношению:

$$v_m = \frac{L_m}{T_s \cdot \tan(\alpha / 2)}$$

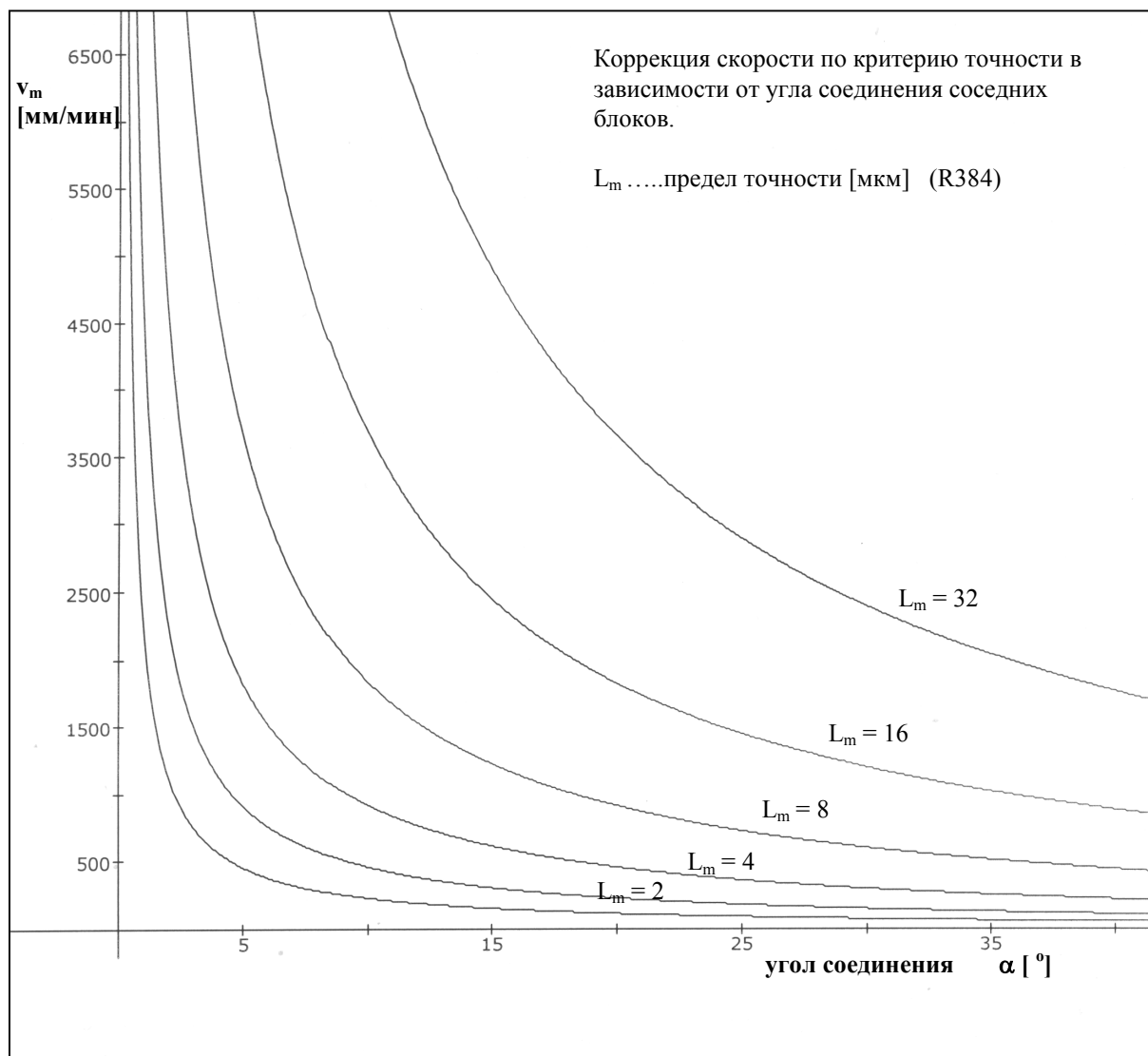
v_m максимальная разрешенная скорость по критерию точности

L_m предел точности, введенный в машинную константу R384

T_s такт интерполяции

α актуальный угол соединения соседних блоков

Параметр R384 годится устанавливать по типу обработки (черновая обработка, точная обработка). Если будет введена величина ноль, критерий точности не активирован. Если критерий точности не активирован или в параметр R384 установлена большая величина для предела (50 микронов), в большинстве случаев будет ограничиваться скорость динамическим критерием по параметру R385, который для данного станка твердо установлен и не должен меняться.



Приложение A1.1 – динамический критерий, параметр R385

Система с достаточным опережением ограничивает скорость в случае плавного соединения и в случае активной комплексной скорости таким способом, чтобы не превышалась разрешенная перегрузка станка, установленная в 1. - 7. декадах параметра **R385**. Величина вводится в мм/с^2 .

Установка параметра узко связана с динамикой приводов и с установленной величиной усиления (K_v) для сервопетлей. Принцип основан на предположении, что система приводов со всей механикой станка обладает определенной максимальной разрешенной перегрузкой (ускорением) и система ограничит скорость таким способом, чтобы она не была превышена.

Максимальная скорость с точки зрения требуемой перегрузки зависит от угла соединения соседних блоков по 7 формуле:

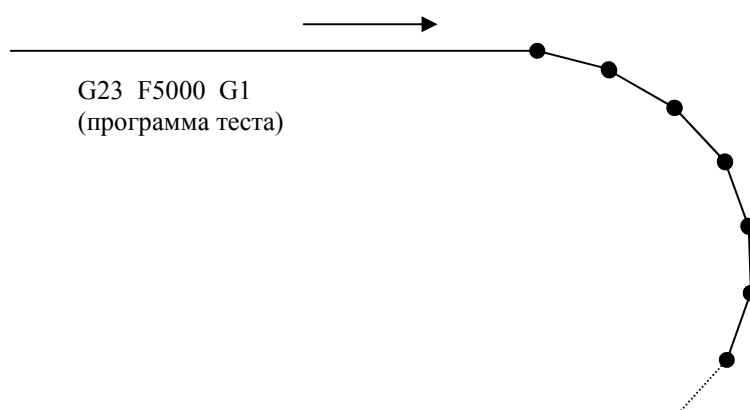
$$v_m = \frac{a_m \cdot T_s}{2 \cdot \sin(\alpha / 2)}$$

- V_m максимальная разрешенная скорость по динамическому критерию
- a_m максимальное разрешенное ускорение, введенное в машинную константу R385
- T_s такт интерполяции
- α актуальный угол соединения соседних блоков

Динамический критерий – это параметр для конкретного станка, и поэтому он не должен после установки меняться. Во время старта уместно составить программу для теста, которая состоит из малых блоков с линейным движением одинаковой длины. Длина блоков должна соответствовать приблизительно минимальной длине блоков, которые генерирует большая проектируемая система CAD. Рекомендуем использовать длину блока серии **1мм, 0.5мм, 0.1мм** или **0.05мм**, потому что после этого будет можно активировать коррекцию динамического критерия по длине блока. Блоки должны быть между собой повернуто всегда на одинаковый угол, например 10 градусов. Вся программа для тестов бы таким способом создавала резкий поворот движения на 180 градусов. Программа должна содержать функцию G23 и скорость, которая будет типичной для обработки конкретным станком. Во время прохождения программы для теста постепенно начинается динамический критерий, например при помощи контроля отклонения от положения во время выбора индикации.

В восьмой декаде параметра R385 может активироваться коррекция критерия перегрузки на длину блока, при этом для правильной работы необходимо ввести величину по использованной длине блоков во время установки динамического критерия (референционная длина):

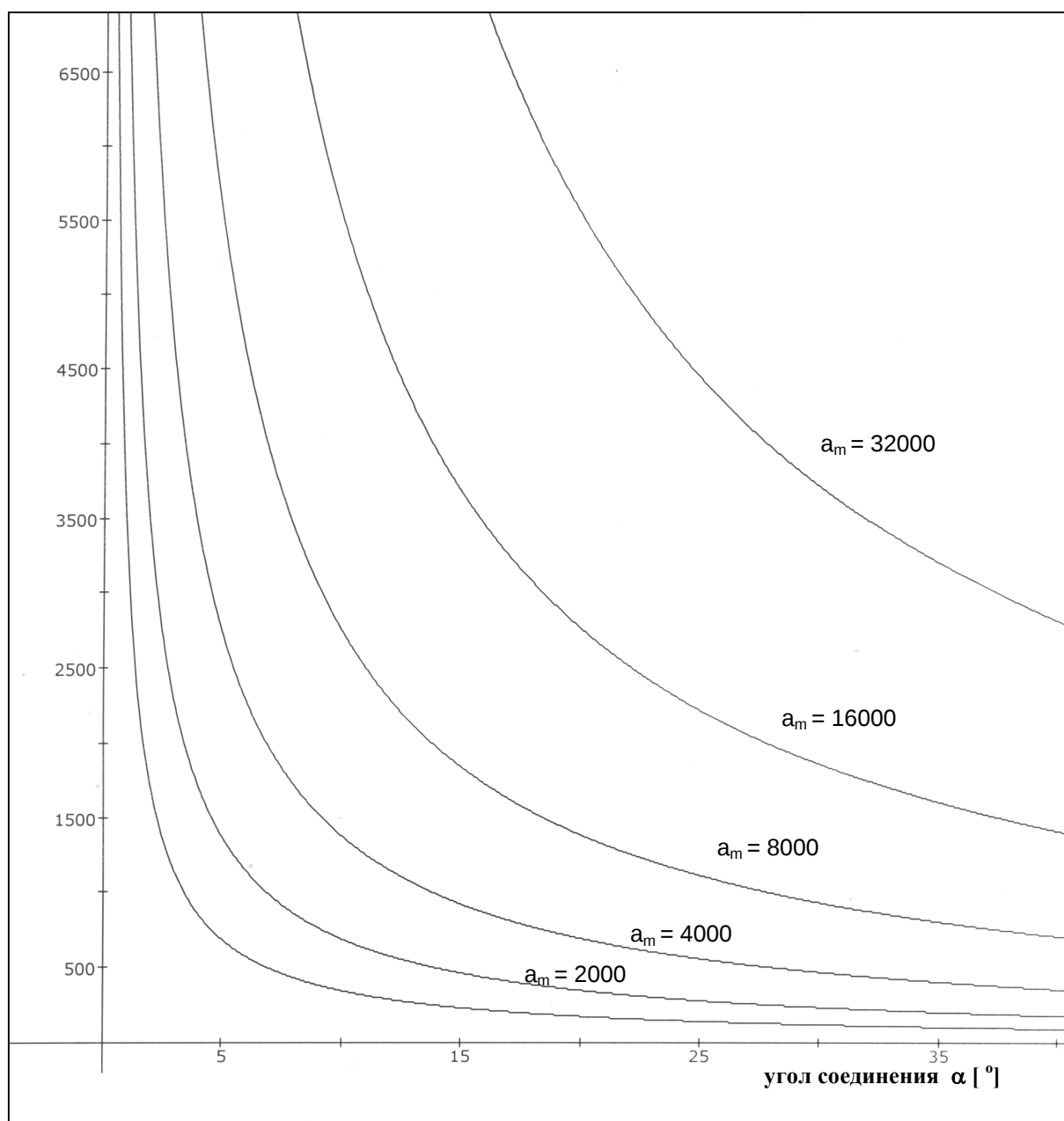
8. декада R385	0	Коррекция динамического критерия не активирована.
	1	Референционная длина блоков составляет 1 мм.
	2	Референционная длина блоков составляет 0.5 мм.
	3	Референционная длина блоков составляет 0.1 мм.
	4	Референционная длина блоков составляет 0.05 мм.

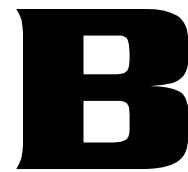


Коррекция скорости по динамическому критерию в зависимости от угла соединения соседних блоков.

V_m
[мм/мин]

a_m максимальная перегрузка [мм/с²] (R385)





Приложение В – приоритетный блок

(Действительно для версии пульта 20.23 и выше.)

Во время выбора партпрограммы и после центрального обнуления системы заполняется рабочее запоминающее устройство т.наз. приоритетным блоком, в котором указаны следующие адреса и функции:

F0	нулевая скорость
S0	нулевые обороты
&0	отмена коррекций длины
G1	линейная интерполяция
G17	плоскость коррекции G17
G98	отмена плавного соединения блоков
G94	сдвиг в мм/мин.
G40	отмена коррекции радиуса
G53	включенный сдвиг начала
G80	отмена твердых циклов
G90	абсолютное программирование
M5	стоп шпинделя
M36	сдвиг программируется прямо
M9	отключение охлаждения 1 и 2
M53	отключение охлаждения 3 и 4
M48	отмена перехода FEED OVERRIDE

Если в **первом блоке** партпрограммы запрограммировано другое, эти адреса и функции применяются. Все эти адреса и функции применяются, т.е. будут переданы также во время **ЦЕНТРАЛЬНОГО ОБНУЛЕНИЯ**.

В некоторых случаях может пользователь потребовать остальные функции, или адреса, или наоборот некоторые функции или адреса из блока исключить.

Указываем некоторые случаи, которыми может пользователь воспользоваться:

1. Требование к другому приоритетному началу, чем G53. В некоторых случаях может мешать, что центральное обнуление всегда начинается с G53. Можно выбрать другую приоритетную установку, напр. G54. Также возможно не вводить в приоритетный блок никакую функцию для сдвига начала. В этом случае центральное обнуление, ни выбор программы не меняют начало и будет действительна в последний раз введенная функция (напр. из партпрограммы или из режима RUP). В каждом случае рекомендовано всегда в начале партпрограммы запрограммировать соответствующий сдвиг начала, что принадлежит к хорошим привычкам программиста.
2. Отмена адреса F0. После центрального обнуления скорость не будет нулевой и останется по установленному в последний раз значению. В режиме RUP потом не является необходимым скорость для G01 опять вводить.

В любом случае необходимо тщательно взвесить случайные изменения приоритетного блока. Во время изменений приоритетного блока рекомендуем консультироваться с производителем.

Приложение В1 – установка приоритетного блока

Установка руководствуется машинной константой 283, 1 декадой.

Величина 1. декады	Описание
0	Приоритетный блок обладает значением, указанным во введении этого раздела. Система выполняет эту установку самостоятельно. Эта установка действительна также для версии системы 20.22 и ниже.
1	Приоритетный блок обладает величинами, указанными в файле BLOCKCNF.SYS. Эти величины после каждого включения системы будут введены из этого системного файла.
2	Во время выбора партпрограммы BLOCKCNF.NCP создается (или переписывается, если уже существует) системный файл BLOCKCNF.SYS.

Установка на 0 совместима со старшими версиями и удовлетворяет в большинстве случаев. Система выполняет эту установку самостоятельно.

Изменение приоритетного блока будет выполнено по следующему порядку:

В системе находится файл партпрограммы с (обязательным) названием **BLOCKCNF.NCP**, который имеет следующее содержание первого блока (он схожий с первоначальной установкой):

%1

N1 F0 S0 &0 G1 G17 G98 G94 G40 G53 G80 G90 M5 M36 M9 M53 M48

N2 M30

1. Пользователь в редакторе установит блок N1 по своим требованиям, напр. поменяет G53 на G54.
2. Установит первую декаду машинной константы номер 283 на 2.
3. Сделает выбор партпрограммы BLOCKCNF.NCP. После выбора появится сообщение о создании нового приоритетного блока.
4. Будет установлена первая декада машинной константы номер 283 на 1.
5. После нового включения системы (и во время каждого следующего) уже будем работать с новым приоритетным блоком.
6. После проверки правильной работы можно файл BLOCKCNF.NCP из запоминающего устройства стереть, или данные сохранить в системном файле BLOCKCNF.SYS.

Пометка 1:

Важно после создания приоритетного блока установить первую декаду назад на 1, в противоположном случае бы во время каждого выбора файла BLOCKCNF.NCP создавался системный файл BLOCKCNF.SYS снова.

Пометка 2:

Из-за безопасности созданных данных с приоритетным блоком для системного файла BLOCKCNF.SYS установлен атрибут «скрытый» и его не видно даже через системный редактор. Если хотим приоритетный блок пользователя отменить, установим первую декаду на 0.

Пометка 3:

Рекомендуется файл BLOCKCNF.NCP стереть из запоминающего устройства. Если сотрем файл BLOCKCNF.NCP из запоминающего устройства, можем его в случае необходимости опять записать из запоминающего устройства EPROM (A:\PROG) для систем без твердого диска или с твердого диска (C:\SYSTPROG) для систем с постоянным диском.

Пометка 4:

Файл BLOCKCNF.SYS является двоичным файлом и не может храниться в архиве при помощи системных средств архивации, т.е. нельзя его записать в файл ARCHIV.SYS. В файл ARCHIV.SYS можно в необходимом случае записать только в форме текста BLOCKCNF.NCP.