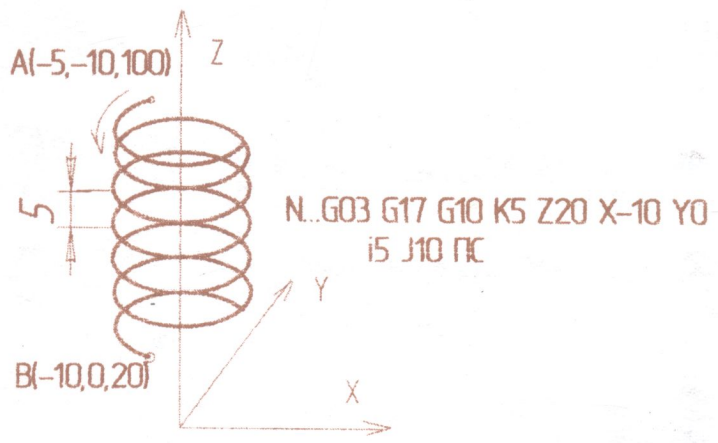


04-9(075)  
сд8

Л.М. СЕДІНКІН

# ПРОГРАМУВАННЯ ОБРОБКИ ДЕТАЛЕЙ НА ВЕРСТАТАХ З ОПЕРАТИВНИМИ СИСТЕМАМИ КЕРУВАННЯ

Навчальний посібник



19

621.9(075)  
С2.8



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Л.М. СЕДІНКІН

**ПРОГРАМУВАННЯ ОБРОБКИ ДЕТАЛЕЙ НА ВЕРСТАТАХ З  
ОПЕРАТИВНИМИ СИСТЕМАМИ КЕРУВАННЯ**

*Рекомендовано Міністерством освіти і науки України  
як навчальний посібник для студентів вищих навчальних закладів*

З обмінного фонду бібліотеки  
Сумського державного  
університету



621.9(075) С28 2006

Седінін Л.М. Програмування обробки деталей

Суми  
Вид-во СумДУ  
2006  
З обмінного фонду  
бібліотеки  
Сумського державного  
університету

**ББК 34.632**  
**С 28**  
**УДК 621.941**

Рецензенти:

д-р технічних наук, професор А.І. Грабченко  
(Харківський національний технічний університет);

д-р технічних наук, професор М.В. Захаров  
(Сумський національний аграрний університет);

д-р технічних наук, професор М.П. Мазур  
(Хмельницький національний університет)

*Рекомендовано Міністерством освіти і науки України  
(лист № 14 18.02 - 244 від 02.02. 2006 р.)*

Седінкін Л.М.

454260

С 28 Програмування обробки деталей на верстатах з  
оперативними системами керування. – Суми:  
Вид – во СумДУ, 2006. – 209 с.

ISBN 966-657-068-8

У навчальному посібнику розглянуті методики розроблення керуючих програм для найбільш помірних вітчизняних оперативних систем програмного керування. Матеріал кожного розділу ілюстрований великою кількістю практичних прикладів, що дозволяє засвоювати програмування під час самостійного вивчення курсу при впровадженні в навчальний процес модульно - рейтингової системи контролю знань. Навчальний посібник призначений для студентів інженерних спеціальностей, які вивчають курс „Програмування обробки деталей на верстатах з ЧПК”, але може бути використаний для виробничого навчання операторів верстатів з ЧПК.

**ББК 34.632**

ISBN 966-657-068-8

© Седінкін Л.М., 2006

© Вид – во СумДУ, 2006

НТБ ВНТУ  
м. Вінниця

## Розділ 1

<b>Основні поняття і визначення.....</b>	<b>12</b>
1.1 Верстати з ЧПК як новий вид робочих машин.....	12
1.2 Терміни і визначення.....	14
1.3 Система координат верстатів із ЧПК.....	17
1.4 Мова програмування.....	19

## Розділ 2

**Програмування обробки деталей на  
токарному верстаті із системою керування.**

**«Електроніка НЦ-31».....**

<b>2.1 Функціональні можливості і відмітні риси машини.....</b>	<b>21</b>
2.2 Адреси та їх призначення.....	21
2.3 Допоміжні команди.....	23
2.4 Програмування робочої подачі.....	24
2.5 Програмування головного руху.....	24
2.6 Програмування найпростіших рухів верстата.....	24
2.6.1 Програмування однопрохідної зовнішньої обробки.....	24
2.6.2 Програмування підрізування торця за один прохід.....	25
2.6.3 Програмування обробки конуса за один прохід.....	25
2.6.4 Програмування обточування фасок під кутом $45^{\circ}$ .....	27
2.6.5 Програмування обробки галтелей.....	27
2.6.6 Програмування обробки дуг кола менше $90^{\circ}$ .....	28
2.7 Технологічні цикли.....	29
2.7.1 Багатопрохідний цикл нарізування різи різцем G31.....	30
2.7.2 Однопрохідний цикл поздовжнього обточування G70.....	32

2.7.3 Однопрохідний цикл підрізування торця G71.....	33
2.7.4 Багатопохідний цикл зовнішнього обточування G77.....	33
2.7.5 Багатопохідний цикл підрізування торця G78.....	35
2.7.6 Цикл глибокого свердління G73.....	36
2.7.7 Цикл прорізання концентричних торцевих канавок G74.....	37
2.7.8 Цикл прорізання канавок з рівним кроком на циліндричній поверхні деталі G75.....	38
2.8 Спеціальні команди.....	39
2.8.1 Безумовний перехід R.....	39
2.8.2 Повтор частини програми за функцією G25.....	40
2.8.3 Витримка часу.....	43
2.9 Підпрограми.....	43
2.9.1 Функція G23.....	43
2.10 Зсув нуля прив'язки G92.....	46
2.11 Група циклів умови руху.....	48
2.11.1 Програмування з використанням циклів умови руху.....	49

## Розділ 3

<b>Параметричне програмування для керуючої машини «Електроніка НЦ-31».....</b>	<b>51</b>
3.1 Команди пересилання.....	51
3.2 Команди додавання.....	51
3.3 Команди непрямого відпрацьовування.....	53
3.4 Параметричний виклик підпрограми за функцією G21.....	55
3.5 Програмування умовних переходів.....	58
3.5.1 Перехід за зовнішньою умовою.....	59
3.5.2 Умовний перехід за плюсом.....	59
3.5.3 Умовний перехід за мінусом.....	61
3.5.4 Перехід за нулем або плюсом.....	61

3.5.5 Перехід за нулем або мінусом.....	61
3.5.6 Перехід за нулем лічильника.....	61
3.6 Приклад керуючої програми, розробленої з використанням методики параметричного програмування.....	64

## Розділ 4

### Програмування обробки деталей

#### на токарному верстаті із системою 2P22..... 71

4.1 Характеристика системи керування.....	71
4.2 Адреси та їх призначення.....	71
4.3 Службові функції.....	72
4.4 Допоміжні функції.....	73
4.5 Програмування подачі.....	73
4.6 Програмування головного руху.....	73
4.7 Програмування найпростіших переміщень.....	74
4.7.1 Програмування зовнішнього обточування за один прохід.....	74
4.7.2 Програмування підрізування торця за один прохід.....	74
4.7.3 Обточування конуса за один прохід.....	75
4.7.4 Програмування обробки фасок під кутом $45^{\circ}$ .....	76
4.7.5 Програмування обробки галтелей.....	77
4.7.6 Програмування обробки дуг окружністю менше $90^{\circ}$ .....	78
4.8 Програмування технологічних циклів.....	79
4.8.1 Технологічний цикл нарізання різі різцем.....	80
4.8.2 Програмування циклу прорізування широкої канавки.....	80
4.8.3 Програмування циклу «петля зовнішня».....	81

4.8.4 Програмування циклу «петля внутрішня».....	83
4.8.5 Програмування циклу «петля торцева».....	83
4.8.6 Програмування циклу «глибоке свердління».....	84
4.8.7 Програмування циклів «чорнова обробка за описом креслення».....	84
Коментарі до стандартних циклів.....	87

## Розділ 5

<b>Методика програмування обробки деталей на вертикально – фрезерних верстатах із системою керування 2C42-65.....</b>	<b>89</b>
5.1 Деякі особливості програмування для фрезерування кінцевою фрезою.....	89
5.2 Адреси та їх призначення.....	90
5.3 Службові функції.....	91
5.4 Допоміжні функції.....	91
5.5 Програмування головного руху.....	91
5.6 Програмування подачі.....	92
5.7 Лінійна інтерполяція.....	92
5.8 Кругова інтерполяція.....	94
5.9 Лінійно – кругова інтерполяція.....	95
5.10 Приклади безеквідистантного програмування.....	95
5.10.1 Програмування в збільшеннях.....	98
5.10.2 Програмування в абсолютних координатах.....	100
5.11 Програмування з використанням стандартних циклів..	101
5.11.1 Цикл свердління отвору.....	103
5.11.2 Цикли розточування.....	104
5.11.3 Цикл глибокого свердління.....	105
5.11.4 Цикл переривчастого свердління.....	106
5.12 Повторення кадру програми.....	

## Розділ 6

<b>Параметричне програмування для машини 2C42-65.....</b>	<b>107</b>
---------------------------------------------------------------	------------

6.1 Алфавіт мови макропрограмування .....	107
6.2 Представлення чисел .....	107
6.2.1 Константи.....	107
6.3 Формальна пам'ять.....	107
6.4 Системна пам'ять.....	107
6.5 Вираження мови макропрограмування.....	108
6.5.1 Безпосередній спосіб звертання до комірки пам'яті .....	109
6.5.2 Непрямий спосіб звертання до комірки пам'яті.....	109
6.6 Оператор присвоєння.....	109
6.7 Неявний оператор присвоєння.....	110
6.8 Арифметичні вирази .....	110
6.8.1 Оператор додавання.....	110
6.8.2 Оператор віднімання .....	111
6.8.3 Оператор множення .....	112
6.8.4 Оператор ділення .....	112
6.9 Обчислення функцій .....	113
6.9.1 Обчислення кореня квадратного.....	113
6.9.2 Обчислення синуса і косинуса .....	113
6.9.3 Обчислення арктангенса .....	114
6.9.4 Округлення чисел.....	114
6.9.4.1 Округлення за правилами арифметики.....	114
6.9.4.2 Округлення з нестачею.....	115
6.9.4.3 Округлення з надлишком.....	115
6.10 Визначення модуля числа.....	115
6.11 Умовні вирази .....	115
6.12 Текстові вирази.....	116
6.13 Структура програми.....	116
6.13.1 Технологічний кадр.....	117
6.13.2 Обчислювальний кадр.....	117
6.14 Керування виконанням програми.....	118
6.14.1 Номер кадру.....	118
6.15 Оператор передачі керування.....	118
6.15.1 Безумовний оператор передачі керування .....	118



6.15.2 Умовний оператор передачі керування.....	119
6.15.3 Призначуваний оператор передачі керування.....	120
6.16 Послідовність виконання операцій у виразі.....	120
6.17 Розроблення підпрограм .....	122
6.17.1 Виклик підпрограм.....	122
6.17.2 Відпрацьовування підпрограм і повернення в основну програму.....	122
6.17.3 Виконання програм.....	124
6.18 Приклади програм, виконаних мовою макропрограмування.....	124
6.18.1 Програма 1.....	124
6.18.2 Програма 2.....	128
<b>Розділ 7</b>	
<b>Програмування обробки деталей на верстатах із системою керування WL4T.....</b>	<b>133</b>
7.1 Призначення і характеристика машини.....	133
7.2 Адреси та їх призначення.....	133
7.3 Структура кадру програми.....	134
7.4 Службові функції.....	135
7.5 Допоміжні функції.....	136
7.6 Програмування подачі.....	137
7.7 Програмування головного руху.....	138
7.8 Програмування паузи.....	138
7.9 Програмування гальмування наприкінці кадру.....	138
7.10 Програмування геометричної інформації.....	138
7.11 Програмування зміни інструмента...	139
7.12 Лінійна інтерполяція.....	139
7.13 Кругова інтерполяція.....	140
7.14 Програмування технологічних циклів.....	141
7.14.1 Багатопрохідний цикл нарізування різі різцем.....	141

7.14.2 Різальний рух.....	145
7.14.3 Зворотно – поступальний різальний рух.....	146
7.14.4 Однопрохідний поздовжній цикл.....	146
7.14.5 Однопрохідний поперечний цикл.....	147
7.14.6 Цикл глибокого свердління.....	147
7.14.7 Багатопохідний цикл прорізання каналок на торцевій поверхні деталі G74.....	148
7.14.8 Багатопохідний цикл прорізування циліндричних каналок.....	149
7.14.9 Багатопохідний цикл чорнової обробки циліндричної поверхні.....	149
7.14.10 Багатопохідний цикл чорнового підрізування торця.....	150
7.15 Методика програмування контурної обробки...	151
7.16 Зміщення нульової точки деталі.....	155
7.17 Постійні цикли.....	155
Приклад керуючої програми для вертикально – фрезерного верстата.....	156

## Розділ 8

### Параметричне програмування

для системи WL4T.....	160
8.1 Види пам'яті машини.....	160
8.2 Вирази.....	161
8.2.1 Унарні операції.....	162
8.2.2 Бінарні операції.....	162
8.2.3 Порядок обчислення виразів.....	163
8.3 Керування ходом виконання програми.....	163
8.3.1 Команди безумовного переходу.....	163
8.3.2 Команди умовного переходу.....	164

## Розділ 9

Методика підготовки керуючих програм для верстатів із системою керування серії NC-110 (200)).	165
9.1 Загальна характеристика системи керування.....	165
9.2 Характеристики програмування.....	165
9.2.1 Системи вимірювання.....	165

9.2.2 Програмування головного руху.....	165
9.2.3 Програмування подачі.....	166
9.2.4 Програмування зміни інструмента...	166
9.2.5 Службові функції та їх призначення.....	166
9.2.6 Допоміжні функції.....	167
9.2.7 Початок і кінець програми.....	169
9.2.8 Деякі особливості застосування адрес.....	169
9.3 Види рухів виконавчих механізмів.....	170
9.3.1 Програмування прискореного переміщення.....	170
9.3.2 Лінійна інтерполяція.....	171
9.3.3 Кругова інтерполяція.....	171
9.3.4 Гвинтова інтерполяція.....	173
9.3.5 Нарізування різі різцем.....	174
9.3.6 Програмування обертання деталі.....	175
9.4 Безеквідистантне програмування обробки контуру.....	178
9.5 Програмування з використанням стандартних циклів.....	183
9.5.1 Програмування стандартного циклу свердління.....	185
9.5.2 Програмування циклу глибокого свердління.....	186
9.5.3 Програмування циклу нарізування різі мітчиком.....	188
Коментарі до стандартних циклів.....	190

## Розділ 10

<b>Програмування з використанням мови GTL.....</b>	<b>193</b>
10.1 Векторна геометрія.....	193
10.2 Збереження в пам'яті машини геометричних елементів.....	194
10.3 Визначення точок початку відліку.....	194
10.4 Визначення точок.....	197
10.5 Визначення прямої лінії.....	198

10.6	Визначення окружностей.....	201
10.6.1	Визначення окружностей в явній формі.....	201
10.6.2	Визначення окружностей в неявній формі....	202
10.7	Приклад програмування мовою GTL.....	205
10.8	Деякі правила програмування мовою GTL.....	207

**Розділ 1****Основні поняття і визначення****1.1 Верстати з ЧПК як новий вид робочих машин**

Уперше верстат із ЧПК був представлений на Всесвітній виставці в Брюсселі в 1959 році. Це був верстат, створений у ЕНІМВ (м. Москва) на базі універсального вертикально-фрезерного верстата Горьківського заводу фрезерних верстатів.

Незважаючи на примітивність його конструкції з позиції сучасного уявлення про подібні верстати, йому було присуджено «ГРАН-ПРІ» цієї виставки.

З цього часу починається широке застосування числового програмного керування для різних робочих машин (верстати, преси, ливарні машини, контролюючі машини і т.д.).

У чому причина настільки широкого застосування ЧПК?

Справа в тому, що з виникненням автоматизованих засобів виробництва між ступенем автоматизації і мобільністю засобів виробництва з'явилося не вирішене протиріччя.

Чим вище ступінь автоматизації, тим менш мобільне виробництво. Це пов'язано з великими капіталовкладеннями, і поки вони не окупляться, змінювати автоматизовані засоби нерозумно і не вигідно. Іноді намагаються всі нові розробки покласти „під сукно”. Тому багато прикладів в історії техніки, наприклад (ГАЗ).

Найбільш мобільним є виробництво, оснащене універсальним устаткуванням, тому що його легко перебудувати на випуск нової продукції. Але воно малопродуктивне, енергоємне і вимагає великих виробничих площ та багато висококваліфікованої робочої сили, що теж є не вигідним ходом розвитку.

Сучасний же стан розвитку техніки вимагає дуже часті зміни об'єктів виробництва і швидкого насичення новою продукцією ринку.

Наприклад, літак застаріває через 3 роки, модель автомобіля повинна оновлюватися через 2-3 роки і т.д.

Розділ 1 Основні поняття і визначення

Поява устаткування з числовим керуванням дозволила створювати мобільні високоавтоматизовані виробництва, тому що в основу такого устаткування покладено універсальну робочу машину з усіма її перевагами, а нею керує теж універсальна електронно-обчислювальна машина.

Якщо тепер до такої робочої машини додати універсальну транспортне – завантажувальну машину (робот), то в результаті ми будемо мати гнучкий технологічний модуль. Група таких модулів з додаванням автоматичних складів заготовок, деталей, інструменту і пристосувань і з керуванням від центральної ЕОМ буде являти собою автоматизовану ділянку, цех, завод. Так створюються виробництва з технологією, де не задіяні люди

Відмінність їх від звичайних заводів - автоматів (підшипникові й ін.) полягає в тому, що ці виробництва залишаються високо мобільними, тому що для зміни об'єкта виробництва досить замінити керуючі програми, різальний інструмент і оснащення.

Отже, перша ознака, що устаткування з ЧПК – це принципово новий вид робочих машин, полягає в тому, що воно дозволило зняти протиріччя між ступенем автоматизації виробництва і його мобільністю.

Друга ознака полягає в тому, що відповідальність за якість продукції і продуктивність перекладається з робочого на інженера. Від того, наскільки якісно розроблені керуючі програми, наскільки оптимальні режими обробки і грамотно складена технологія обробки, будуть залежати і якість продукції, і продуктивність праці. При цьому робочий практично не втручається в процес обробки.

Третя ознака – це зниження вимог до рівня кваліфікації оператора верстата з ЧПК. А це, у свою чергу, веде до зниження потреби у висококваліфікованих робітниках, підготовка яких коштує досить дорого.

Четверта ознака – це стабілізація якості продукції і здешевлення процесу складання за рахунок виключення підганяльних робіт.

І нарешті п'ята ознака – це підвищення культури виробництва. Керуюча машина, що знаходиться біля верстата, – це така сама ЕОМ, як і ті, що знаходяться в обчислювальних центрах. Тому в цеху повинні створюватися такі самі комфортні умови для надійної роботи ЧПК, як і в обчислювальних центрах.

Для нормальної роботи верстатів потрібні стабільні розміри і якість заготовок, отже, підвищуються вимоги до культури заготівельного виробництва.

Ремонт верстатів із ЧПК та їх керуючих машин вимагає високої кваліфікації ремонтної служби і появи на звичайних машинобудівних підприємствах відділів електроніки та ін.

Для розроблення керуючих програм потрібні не просто технологи, а технологи – програмісти, що вміють будувати числові моделі технологічного процесу з використанням математичних методів і обчислювальної техніки.

І нарешті, технологія машинобудування з науки, що мала переважно якісний характер, стала перетворюватися в точну науку.

Програмування технологічних процесів обробки деталей для верстатів із ЧПК – якісно новий етап, на якому виконується значна частина роботи, перенесена зі сфери виробництва у сферу його технологічної підготовки. Тобто дії кваліфікованого робітника, що обробляє заготовку на звичайному верстаті, замінюються на верстаті з ЧПК обробкою за керуючою програмою без утручання людини.

### **1.2 Терміни і визначення**

Стандартом ДСТУ 20523-80 встановлені основні терміни і визначення, що ми тут частково розглянемо.

**Керуюча програма (КП)** – повний обсяг геометричної, технологічної, допоміжної та іншої інформації, необхідної для автоматичної обробки заготовки на даній установці.

**Числове програмне керування (ЧПК)** – керування обробкою заготовки на верстаті за КП, у якій усі дані задані в цифровій формі.

**Позиційне ЧПК** – робочі органи верстата переміщуються в задані точки по неконтрольованих траєкторіях. Мета ЧПК – вивести інструмент у задану точку із заданою точністю (свердлильні, розточувальні верстати).

**Контурне ЧПК** – робочі органи верстата переміщуються по заданій траєкторії і з заданою швидкістю. Мета такого керування - не тільки привести інструмент у задану точку, але і забезпечити таку траєкторію руху формотворної точки інструмента, яка б забезпечила задану форму і розміри оброблюваного при цьому елемента деталі.

**Адаптивне ЧПК верстатом** – відбувається автоматичне пристосування процесу обробки заготовки до реальних умов (зміна глибини різання при зміні припуску, зменшення подачі при зростанні крутного моменту і т.д.).

**Кадр керуючої програми** – складова частина програми, що вводиться і що відпрацьовується як єдине ціле і утримує не менше однієї адреси. Кадр містить повний обсяг інформації, необхідної для обробки елементарної технологічної ділянки деталі. Кадр програми починається з адреси N і записаного за ним номера кадру і закінчується знаком «кінець кадру» (ПС, ЕФ, ◇).

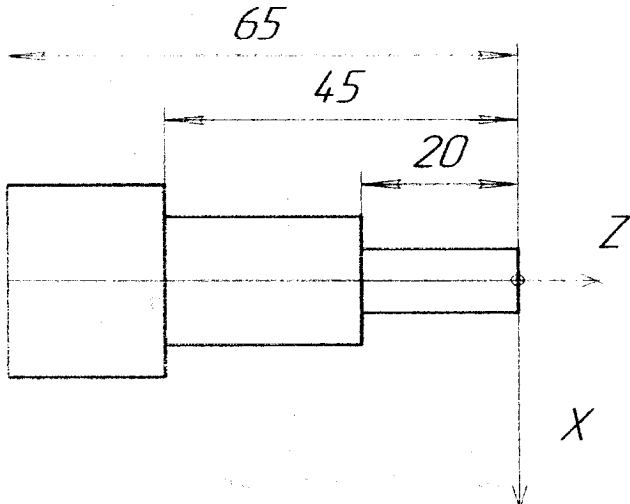
Адреси в межах цих двох знаків усередині кадру можна розміщувати в будь-якій послідовності.

**Елементарна технологічна ділянка** – відрізок прямої або дуги кола, за допомогою яких описаний оброблюваний контур.

**Опорна точка** – точка початку і кінця елементарної технологічної ділянки.

**Абсолютний розмір** – лінійний або кутовий розмір, що заданий в КП і показує положення описуваної точки щодо прийнятого нуля системи координат.





*Рисунок 1.1 - Задання розмірів в абсолютній системі координат*

**Розмір у збільшеннях** – лінійний або кутовий розмір, заданий в КП за відповідними координатами, що показують відстань від початкової точки елементарної технологічної ділянки до його кінцевої точки.

**Нульова точка верстата** – (нуль верстата) – точка, взята за початок координат верстата.

**Вихідна точка верстата** – точка, обумовлена щодо нуля координат верстата і використовується для початку роботи з КП. У цій точці відбувається зміна інструмента.

**Нульова точка деталі** – точка деталі, взята за нуль системи координат даної деталі.

**Дискретність відпрацювання переміщення** – мінімальне переміщення або кут повороту робочого органа верстата, які можна задати в програмі і проконтролювати у процесі відпрацювання.

**Геометрична інформація** – інформація, що описує форму деталі, розміри елементів деталі та інструмента і їх взаємне положення в просторі.

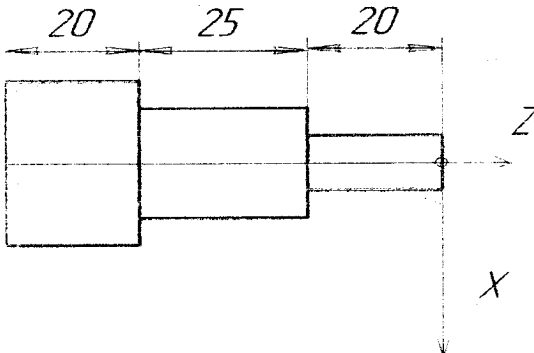


Рисунок 1.2 Задання розмірів у збільшеннях

**Технологічна інформація** – інформація, що описує умови обробки деталі (подача, глибина різання, швидкість різання, частота обертання шпинделя).

**Еквідистанта** – лінія контуру, що проведена на відстані  $r$  фрези від контуру деталі.

### 1.3 Система координат верстатів із ЧПК

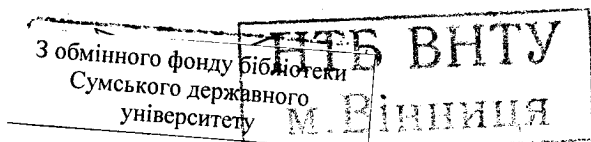
Для того щоб задати рух робочого органа верстата і щоб рух можна було однаково позначати у верстаті будь-якої лінії виробника, Міжнародна організація стандартів ISO зробила рекомендації ISO – R841 (1968 р.) на вибір і означення координатних осей.

За основу взята правобічна прямокутна система координат. Основною віссю є вісь Z. Ця вісь завжди спрямована вздовж осі шпинделя, а її позитивний напрямок прийнято вважати напрямком **від деталі до інструмента**.

Вісь X перпендикулярна осі Z, а її позитивний напрямок буде вправо, якщо дивитися вздовж осі Z у її позитивному напрямку.

Вісь Y перпендикулярна до площини XOZ, а її позитивний напрямок визначається поворотом осі X навколо осі

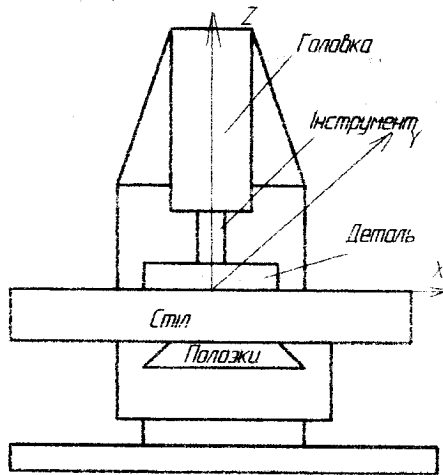
Сумського державного університету



Z за годинниковою стрілкою, якщо дивитися вздовж осі Z у її позитивному напрямку.

Подальший розвиток верстатобудування і поява багатоцільових верстатів із ЧПК потребували додаткової кількості адрес для позначення руху виконавчих механізмів, що переміщуються паралельно основним осям. У зв'язку з цим ISO рекомендувало ще три вторинних і три третинних лінійних осей і п'ять кругових осей. Ці осі позначаються:

- первинні – X, Y, Z;
- вторинні – U, V, W;
- третинні – P, Q, R;
- кругові – A, B, C, D, E.



*Рисунок 1.3 - Осі координат верстата*

Таким чином, якщо у верстаті є додатковий виконавчий механізм, що переміщується паралельно основному вздовж осі X, то його переміщення програмують за адресою U.

Наступний механізм, що рухається в цьому самому напрямку, буде задаватися за адресою P і т. д. Аналогічно - аналоги адрес Y і Z. Кругові осі A, B, C також прив'язані до первинних осей. Якщо механізм робить обертальний рух

Розділ 1 Основні поняття і визначення

навколо осі X, то його позначають адресою A. Навколо осі Y — адресою B, а навколо осі Z — адресою C. Кругові осі D і E не мають закріплення і використовуються за необхідністю.

Позитивний напрямок руху по круговій осі вважається рухом за годинниковою стрілкою, якщо дивитися вздовж лінійної осі в її позитивний бік.

Таким чином, якщо потрібно задати який-небудь рух виконавчого механізму, у програмі записується адреса, тобто назва осі, по якій відбувається цей рух, і величина переміщення — цифрами зі знаком. Знак плюс звичайно не пишеться.

Наприклад, треба перемістити інструмент із точки A в точку B (рис. 1.4). У програмі буде записано N025 X 100 Y 60 Z -10 LF; цей запис зроблений в абсолютних координатах. У збільшеннях цей самий кадр буде мати такий вигляд:  
N025 X 90 Y 40 Z -70 LF.

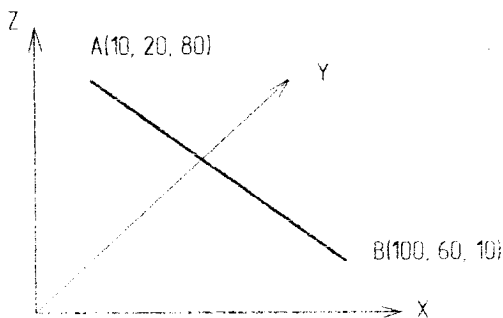


Рисунок 1.4

Рекомендації ISO прийняті всіма національними стандартами, тому програмістові немає необхідності щоразу вивчати систему кодування переміщень при придбанні верстата в будь-якій країні - виробнику.

### **1.4 Мова програмування**

Відповідно до Міжнародного стандарту ISO 840, прийнятого всіма національними стандартами розвинутих країн світу, для кодування інформації для верстатів із ЧПК використовується код ISO - 7bit.

Будь-яка мова починається з алфавіту. Як алфавіт мови ISO використовуються такі знаки:

- 26 букв латинського алфавіту;

- десять цифр від 0 до 9;

- і цілий ряд знаків і значків, таких, як „+”, „-”, LF, #, @, <, >, та ін.

Усередині алфавіту адреси також мають чітке закріплення за видами інформації. Так, адреси X, Y, Z, I, J, K, P, Q, R, U, V, W, A, B, C, D, E використовуються тільки для задання геометричної інформації, адреса G – для задання службової функції, адреса M – для допоміжних команд, адреси S, T, F – для задання технологічної інформації, адреса N – для зазначення номера кадру.

## Розділ 2

### Програмування обробки деталей на токарному верстаті із системою керування «Електроніка НЦ – 31»

#### 2.1 Функціональні можливості і відмітні риси машини

У кадрі програми можна задати тільки одну адресу:

- геометричні розміри можна задавати як в абсолютній системі, так і по збільшеннях;
- дискретність задання геометричної інформації і робочої подачі 0,01мм;
- пам'ять машини має 250 комірок (0 – 249);
- кожен кадр програми займає свою комірку і пересунути його в сусідню неможливо. Тому для коректування програми треба періодично залишати резервні комірки пам'яті;
- крім зовнішньої пам'яті, машина може поставлятися з додатковою пам'яттю, що складається з чотирьох зон по 250 комірок у кожній.

#### 2.2 Адреси та їх призначення

X,Z – координати точки щодо нуля деталі. Якщо в кадрі записаний знак



, то тоді за цими адресами задане збільшення за координатами;

F – робоча подача, мм/об;

S – частота обертання шпинделя, об/хв, або швидкість різання м/хв (при заданні умови сталості швидкості різання);

T – номер інструмента;

R – параметр, призначення якого визначається в кожному випадку видом технологічного циклу;

N – номер кадру;

G – службова функція;

M – допоміжна команда;

+45<sup>0</sup>, -45<sup>0</sup> – ознака обробки фаски під кутом 45<sup>0</sup>;

~ – ознака швидкого ходу.

Таблиця 2.1 – Службові функції

Службова функція	Найменування службової функції ( назва стандартного циклу)	Час дії
G 02	Колова інтерполяція за годинниковою стрілкою (дуга менше $90^{\circ}$ )	В одному циклі
G 03	Колова інтерполяція проти годинникової стрілки (дуга $<90^{\circ}$ )	В одному циклі
G 04	Витримка часу	В одному циклі
G 12	Обробка галтелі за годинниковою стрілкою	В одному циклі
G13	Обробка галтелі проти годинникової стрілки	В одному циклі
G 21	Параметричний виклик підпрограми	В одному циклі
G 23	Виклик підпрограми	В одному циклі
G 25	Повтор частини програми	В одному циклі
G 31	Нарізування різи різцем	В одному циклі
G 32	Різальний рух	В одному циклі
G 33	Нарізування різи мітчиком або плашкою	В одному циклі
G 36	Переривання відпрацьовування кадру	Діє на кадр, у якому задана, і на два наступних кадра
G 55	Програмоване зупинення	В одному циклі
G 56	Задання номера квадранта координатної сітки	В одному циклі
G61-G67	Група циклів умови руху	В одному циклі
G 70	Однопрохідний поздовжній цикл	В одному циклі
G 71	Однопрохідний поперечний цикл	В одному циклі
G 72	Цикл глибокого свердління по осі X	В одному циклі
G 73	Цикл глибокого свердління по осі Z	В одному циклі

*Продовження таблиці 2.1*

G 74	Багатопрхідний цикл підрізування торця з осьовою подачею або проточування концентричних канавок з постійним кроком на торці деталі	В одному циклі
G 75	Цикл обробки прямих зовнішніх канавок	В одному циклі
G 77	Багатопрхідний цикл зовнішнього обточування	В одному циклі
G 78	Багатопрхідний цикл підрізування торця з радіальною подачею	В одному циклі
G 92	Зсув нуля настроювання	В одному циклі
G 94	Подача, мм/хв	До надходження функції G 95
G 95	Подача, мм/об	До надходження функції G 96
G 96	Сталість швидкості різання	До надходження функції G 97
G 97	Скасування G 96	До надходження функції G 96

### **2.3 Допоміжні команди**

M 00 – безумовний зупинник. За цією командою, записаною в кадрі програми, відбудеться зупинення верстата, і поки оператор не натисне кнопку «пуск», верстат буде «стояти». Команда використовується для виміру і підналагодження верстата.

M03 – праве обертання шпинделя;

M04 – ліве обертання шпинделя;

M05 – стоїть шпиндель;

M07 – вмикання подачу СОЖ;

M08 – вимкнути СОЖ;

M17 – кінець підпрограми;

M30 – кінець програми;

M37 – відпрацьовування дзеркально по X;

M38 – відпрацьовування дзеркально по Z;

M41, M42 - діапазони чисел обертів.



## 2.4 Програмування робочої подачі

Робоча подача, що задається в програмі, програмується за адресою F, розмірність подачі, мм/об, а дискретність 0,01. Подача, що задається в технологічних циклах, має дискрету 0,0001.

## 2.5 Програмування головного руху

Головний рух може бути задано частотою обертання шпинделя (G97), і тоді його розмірність, об/хв, при заданні умови сталості швидкості різання (G96) задається в м/хв. Але в обох випадках воно кодується за адресою S. Режим G97 встановлюється автоматично при включенні машини.

Приклад: N1 M41 – перший діапазон чисел обертів шпинделя;

N2 M3 – праве обертання шпинделя;

N3 S850 – (850 об/хв) – частота обертання шпинделя.

Режим сталості швидкості різання задається функцією G96 і має структуру G96, P<sub>1</sub>...P<sub>2</sub>..., де P<sub>1</sub> – максимальне число обертів шпинделя на даному діапазоні чисел обертів; P<sub>2</sub> – розрахункове число обертів шпинделя для даного діаметра заготовки. Значення P<sub>1</sub> і P<sub>2</sub> машиною запам'ятовуються до надходження функції G96 з новими параметрами.

Приклад: N50 G96 \* - режим сталості швидкості різання;

N51 P950 \* - максимальне число обертів на діапазоні;

N52 P200 – розрахункове число обертів для швидкості 90 м/хв;

N53 S90 – швидкість різання 90 м/хв.

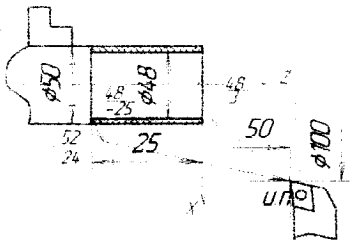
## 2.6 Програмування найпростіших рухів верстата

### 2.6.1 Програмування однопрохідної зовнішньої обробки

Розглянемо приклад програмування обробки поверхні, показаної на рисунку 2.1.

Діапазон чисел обертів задається один раз на початку програми. Аналогічно – напрямок обертання (якщо не потрібний реверс).

Частота обертання може за необхідності задаватися в будь-якому місці програми. Наприклад, при зміні інструмента.



N1 M41	заданий діапазон чисел обертів
N2 M3	напрямок обертів шпинделя
N3 S 450	частота обертів
N4 T1	номер інструмента
N5 F 25	подача 0,25 мм/об
N6 X 4800*	вихід у початок обробки
N7 Z 100	
N8 M 7	вимкнуті ЗОР
N9 Z -2500	обточка циліндра
N10 X 5200*	вихід від поверхні по двох осях
N11 Z -2400	
N12 M 8	вимкнуті ЗОР
N13 X 10000*	} вихід в початкове поло- ження по двох осях
N14 Z 5000	
N15 M 30	

Рисунок 2.1 – Однопрохідне обточування циліндричної поверхні

### 2.6.2 Програмування підрізування торця за один прохід

При програмуванні підрізування торця доцільно задавати умову сталості швидкості різання. Ця умова задається функцією G 96. У наступному кадрі після функції за адресою P записується максимальне число обертів шпинделя на даному діапазоні, потім так само за адресою P – те число обертів, яке необхідне для обраної швидкості різання на периферії заготовки. І вже потім за адресою S задається обрана швидкість різання в м/хв. Якщо по ходу виконання програми потрібно змінити швидкість різання, то в потрібному місці програми треба за адресою S задати нове значення швидкості різання в м/хв.

У будь-якому місці програми умову сталості швидкості різання можна скасувати, задавши функцію G 97. Після цього кадру число обертів за адресою S знову треба задавати в об/хв.

### 2.6.3 Програмування обточування конуса за один прохід

Обточування конуса за один прохід програмується шляхом задання одночасного руху по двох осях на робочій подачі. Це забезпечується за допомогою проставлення зірочки в першому кадрі, де задані координати кінцевої точки конуса (кадр 9 на рисунку 2.3).

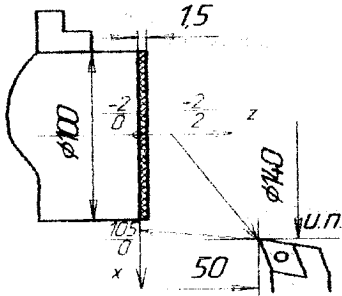


Рисунок 2.2 – Підрізування торця за один прохід

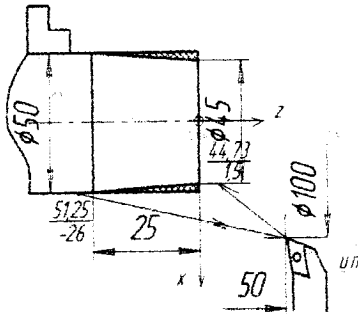


Рисунок 2.3 – Обточування конуса за один прохід

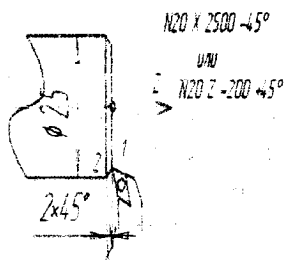
- N1 M41 діапазон частот обертів шпинделя  
 N2 M3 праве обертання шпинделя  
 N3 T2 номер інструмента  
 N4 F15 подача 0,15 мм/об  
 N5 G 96\* умова сталості швидкості різання  
 N6 P2000 \* максимальне число обертів на M41  
 N7 P287 \* розрахункове число обертів для  $V=90$  м/мин  
 N8 S 90 швидкість різання 90 м/мин  
 N9 X 10500 \* ~ вихід у початок обробки по двох координатах одночасно  
 N10 Z 0  
 N11 M 7 вимкнена ЗОР  
 N12 X – 200 робочий хід по шдрізи торця  
 N13 G 97 відміна сталості швидкості різання  
 N14 M8 вимкнуті ЗОР  
 N15 Z 200 ~ відскок від торця  
 N16 X 14000 \* ~ повернення в початкове положення одночасно за двома осями  
 N17 Z5000  
 N18 M30 кінець програми

- N1 M41  
 N2 M3  
 N3 T3  
 N4 S 650  
 N5 F7 робоча подача 0,07 мм/об  
 N6 X 4473 \* ~ вихід в початок розрахункового діаметра конуса  
 N7 Z150  
 N8 M7  
 N9 X5125 \* ~ обточування конуса рухом по двох осях одночасно  
 N10 Z -2600  
 N11 M8  
 N12 X 10000 \* ~  
 N13 Z 5000  
 N14 M30

### 2.6.4 Програмування обточування фасок під кутом $45^{\circ}$

Фаски під кутом  $45^{\circ}$  є типовим елементом деталей, оброблюваних на токарних верстатах. Тому в постійній пам'яті закладений алгоритм їх обробки. Програмування обробки такої фаски здійснюється одним кадром, у якому задається одна з координат кінцевої точки фаски й ознака фаски  $\pm 45^{\circ}$ . Знак у заданій фасці визначається тим, куди відбувається рух інструмента за ненаписаною у кадрі координатою.

Якщо рух йде в позитивному напрямку осі, то ставиться  $+45^{\circ}$ , якщо в негативному напрямку, то  $-45^{\circ}$ .



У розглянутому прикладі різець за попередніми кадрами програми був підведений у точку 1 – початок фаски і кадром 20 задана обробка фаски, тобто рух по відрізку 1-2. Так програмувати можливо тільки фаски під кутом  $45^{\circ}$ . Фаски із кутами, що не дорівнюють  $45^{\circ}$ , треба програмувати як конічні поверхні, тобто двома кадрами.

Рисунок 2.4 - Обробка фаски під кутом  $45^{\circ}$

### 2.6.5 Програмування обробки галтелей

Галтель (дуга  $90^{\circ}$ ), як і фаска, є типовим елементом деталі типу тіл обертання. За допомогою галтелей зменшується концентрація напруг у кутах переходу від циліндра до торця. Приклад програмування обходу галтелі показаний на рисунку 2.5.

Обробка галтелі програмується трьома кадрами. У першому кадрі задається напрямок обходу галтелі. Галтель, що обробляється в напрямку годинникової стрілки, – G 12, і проти – G 13. Потім записується координата кінцевої точки дуги по осі X

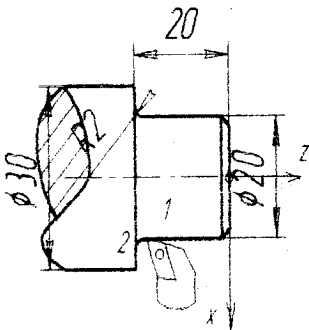
(або відстань від початкової точки до кінцевої по осі X, якщо програмуємо в збільшеннях). І в третьому кадрі задається координата кінцевої точки дуги по осі Z ( або відстань від початкової точки до кінцевої по осі Z).

Цей порядок написання адрес є обов'язковим.

Зверніть увагу на той факт, що збільшення по осі X має дискретність 0,005мм. Тому в кадрі № 16 за адресою X записано не 200, а 400.

### 2.6.6 Програмування обробки дуг кола менше 90°

Дуги кола менше 90° програмуються п'ятьма кадрами. У першому кадрі записується напрямок обходу дуги. Якщо за годинниковою стрілкою, то G02, проти – G03. Далі двома кадрами задаються координати кінцевої точки дуги або відстань від початкової точки дуги до кінцевої по осях X і Z ( якщо програмуємо в збільшеннях). Причому порядок написання саме такий – спочатку X потім Z. Далі два кадри мають адресу R і по них задаються відстані від початкової точки дуги до її центра по осях X і Z. Зазначений порядок запису інформації також обов'язковий. Приклад програмування обходу дуги менше 90° показаний на рисунку 2.6.



У результаті виконання попередніх кадрів програми різець «прийшов» у точку 1 – початок галтелі. Програма обробки галтелі буде така:

N15 G 13\*  
N16 X 2400\*  
N17 Z -2000

У збільшеннях це буде так:

N15 G 13\*  
N16 X 400\*  
N17 Z -200

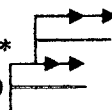


Рисунок 2.5 – Обробка галтелі

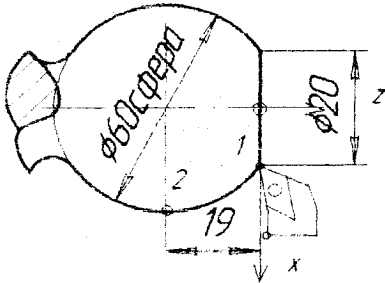


Рисунок 2.6 Обробка дуги менше  $90^\circ$

Розглянемо програмування дуги 1-2, наведеної на рисунку 2.6. У результаті попередніх кадрів інструмент «прийшов» у точку 1. Далі програма буде така:

```

N 22 G02*
N23 X 6000 *
N24 Z -1900 *
N25 P -1000 *
N26 P -1900
N27 G02 * Починаємо описувати наступну дугу.
  
```

## 2.7 Технологічні цикли

Для скорочення обсягу КП і спрощення підготовки програми в пам'яті машини зашиті алгоритми деяких прийомів обробки на токарному верстаті, які мають назву – технологічний цикл.

При програмуванні технологічних циклів необхідно виконувати кілька правил.

**Правило 1** Перш ніж запрограмувати будь-який технологічний цикл, треба інструмент підвести у вихідну точку даного циклу.

**Правило 2** Виконавши заданий цикл обробки, машина поверне інструмент у вихідну точку циклу.

**Правило 3** Подачі, що задаються усередині циклу, записуються з дискретністю 0,0001.

**Правило 4** Якщо програміста влаштовує подача, яка задана до програмування циклу, то усередині циклу подачу можна не задавати.

**Правило 5** Якщо подача задана усередині циклу, то всі чорнові проходи щодо знімання припуску машина виконає на цій подачі, а останній, чистовий, прохід по

оброблюваному контуру проведе з подачею, що була записана у програмі до даного циклу.

### 2.7.1 Багатопрохідний цикл нарізування різі різцем G31

За цим циклом можна нарізати зовнішні, внутрішні, циліндричні, конічні різі з будь-яким типом профілю різі й у будь-яких одиницях виміру, тобто метричну, дюймову, модульну, пітчеву і т.д.

Вихідну точку циклу знаходять так: по осі X повинен бути зазор від вершини різця до фактичного розміру заготовки під різь 2 – 10мм, по осі Z повинен бути зазор від початку різі 2 – 3 кроки різі. Якщо різець підвести у вихідну точку по осі X без зазору, то машина нарізає ліву різь. Зазор по осі Z необхідний, щоб відбулася синхронізація обертання деталі і руху різця.

Структура циклу:

N...G32~\*

N...X.....\* - фактичний розмір діаметра заготовки під різь;

N...Z.....\* - координата кінця різі;

N...F.....\* - крок різі з дискретністю 0,0001мм;

N...R.....\* - глибина різі на радіусі у збільшеннях;

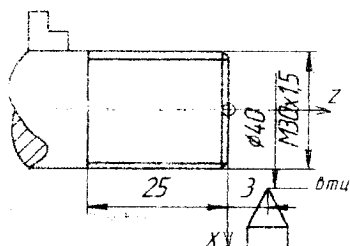
N...R.....\* - глибина різання у кожному проході;

N...R..... - нахил різі (різниця радіусів при конічній різі). Якщо різь циліндрична, її не задають.

Знак ~ у заголовку циклу означає, що переміщення різця на глибину різання буде виконуватися на прискореному ході. Якщо цей знак опустити, то цей рух буде відбуватися на робочій подачі.

Приклад: необхідно нарізати зовнішню метричну різь M30x1,5, довжина різі 25мм. Заготовка під різь буде мати розмір 29,7мм (з урахуванням підйому витків за рахунок пластичної деформації, що має місце при різанні).

Вихідна точка циклу буде по X на діаметрі 40мм, по Z - 3мм від торця різі.



- N10 X 4000\* ~ вихід у вихідну
- N11 Z300            точку циклу
- N12 G31~\*
- N13 X 2970\*
- N14 Z-2500\*
- N15 F 15000\*
- N16 P 87\*
- N17 P 12

*Рисунок 2.7 – Нарізування різи різцем*

Даний цикл має один істотний недолік – глибина різання є постійною в процесі прорізання всієї канавки різи. У той самий час навантаження на різець постійно збільшується. Це часто призводить до викрашування вершини різця, особливо при нарізуванні у важкооброблюваних матеріалах. Для компенсації цього недоліку в системі НЦ – 31 є цикл «різальний рух» G32. Суть цього циклу полягає в тому, що програмується щораз тільки один прохід різця по різі. При цьому за рахунок установлення різця у вихідну точку циклу по осі X ми можемо регулювати глибину різання в кожному різальному русі. Ця точка є внутрішній діаметр канавки в даному проході.

Розглянемо розв'язання задачі попереднього прикладу. Глибина канавки різи 0,87мм. Поділимо її на такі глибини різання:  $0,3+0,2+0,1+0,07+0,07+0,05+0,05+0,03=0,87$ .

Тоді перша вихідна точка по X буде  $29,7-0,3-0,3=29,1$ , друга –  $29,1-0,2-0,2=28,7$ , третя –  $28,7-0,1-0,1=28,5$  і т.д.

Фрагмент програми буде мати такий вигляд:

- N16 X2910\*~    } вихід у першу вихідну точку по осях X і Z
- N17 Z300        } на прискореному ході.
- N18 G32\* різальний рух.
- N19 X2910\* - діаметр дна канавки.
- N20 Z -2500\* - кінцева точка різи по осі Z .
- N21 F15000 – крок різи.
- N22 X2870 ~ - нова вихідна точка по осі X.



N23 G32\* - різальний рух.

N24 X2870\* - діаметр дна канавки в результаті другого проходу.

N25 Z-2500\* - кінцева точка різі по осі Z.

N26 F15000 - крок різі.

N27 X2850 ~ - нова вихідна точка по осі X і т.д.

## 2.7.2 Однопрохідний цикл поздовжнього обточування G70

Працює за такою схемою:

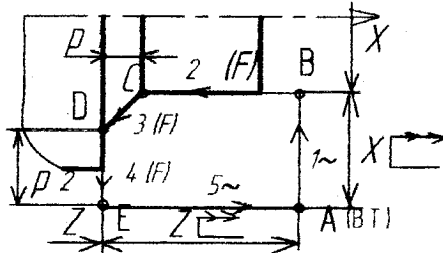


Рисунок 2.8 Схема руху формотворної точки інструмента у циклі G70

B.T. - вихідна точка циклу

1,2,3,4,5 - порядкові номери елементарних рухів циклу

За цим циклом можна здійснити чистову обробку циліндра, конуса і підрізати торець.

Структура циклу в КП

~ G70, X....., (X.....)Z....., (Z.....), F....., P<sub>1</sub>....., P<sub>2</sub>.....,

де ~ - ознака того, що установлення різця в точку B для початку обточування буде відбуватися на прискореній подачі (переміщення 1). Якщо цей знак не задати, то на відрізку 1 буде робоча подача - F.

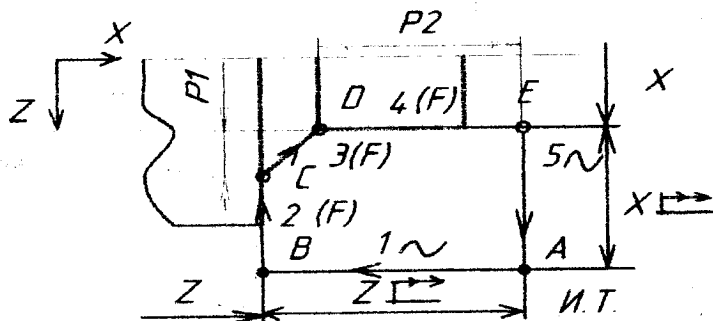
Інші параметри циклу бачимо з рисунка.

При цьому параметри F, P<sub>1</sub>, P<sub>2</sub> можна опускати, наприклад, якщо деталь не має конічного переходу і подача, яка задана в програмі до циклу, буде діяти і при виконанні циклу. Крім того, відстань від точки A до точки B повинна дорівнювати відстані CE.

### 2.7.3 Однопрохідний цикл підрізування торця G71

Схему рухів цього циклу бачимо з рисунка 2.9.

Цей цикл аналогічний до попереднього. Відрізняється тим, що тут підрізається торець.



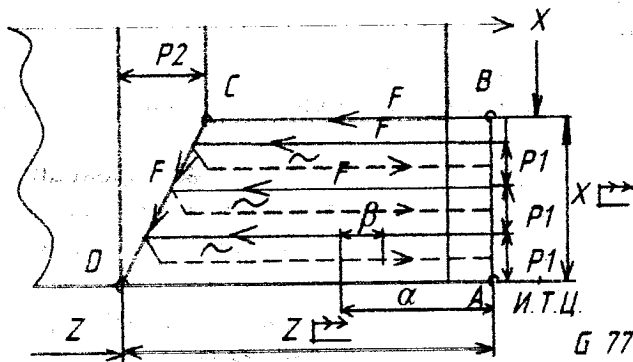
G 71

Рисунок 2.9 – Схема рухів формувальної точки інструмента у циклі G71

Тут так само можна опускати параметри P і F. Якщо параметри P відсутні, то буде тільки підрізатися торець.

### 2.7.4 Багатопрохідний цикл зовнішнього обточування G77

Структура рухів цього циклу показана на рисунку 2.10.

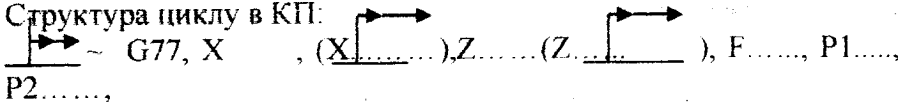


G 77


Рисунок 2.10 – Схема рухів формувальної точки різця при виконанні циклу G77

Цей цикл забезпечує знімання припуску за кілька проходів уздовж напрямних циліндра з автоматичним стружкодробленням.

Структура циклу в КП:

 ~ G77, X , (X), Z (Z), F, P1, P2, .....

де ~ - ознака того, що переміщення на глибину різання (P1) буде відбуватися на швидкому ході. Якщо його не задати, то шлях P1 і величина відскакування будуть відбуватися на робочій подачі F.

При заданні знака  в заголовку циклу машина після виконання циклу поставить різець у точку В. Якщо цей знак відсутній, то різець повернеться в точку А (вихідну точку циклу).

P1- глибина різання;

P2- довжина конічної частини деталі. Параметр P2 не задається, якщо деталь не має конічної частини;

$\alpha$  - довжина проходу до переривання стружки. Задається параметром № 10 за адресою P. Якщо P=0, то дроблення не буде відбуватися;

$\beta$  - відведення інструмента на швидкому ході після проходу різця на величину Z. Ця величина задається параметром № 11 за адресою P.

Ці параметри позитивні і задаються в разі потреби при налагодженні верстата. При цьому повинна виконуватися умова  $\alpha < \beta$ .

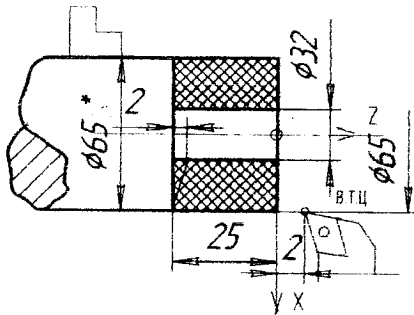
Якщо в циклі подача задана, то всі проходи з глибиною різання P1 будуть відбуватися з цією подачею, а останній прохід, глибина різання якого менше P1, буде відбуватися на подачі, заданій до початку циклу.

Якщо в циклі подачу не задавати, то всі проходи будуть відбуватися на подачі, заданій до циклу.

Вихідна точка циклу знаходиться на діаметрі заготовки з зазором 2 – 3мм по осі Z.

За цим циклом можна обробляти як зовнішні, так і внутрішні поверхні.

Фрагмент програми



N15 X6500 \* ~  
 N16 Z200  
 N17 G77 \* ~  
 N18 X3200 \*  
 N19 Z-2500 \*  
 N20 F2500 \*  
 N21 P300

Рисунок 2.11

Приклад використання циклу G77 показаний на рисунку 2.11.

У прикладі немає ні конічної частини, ні галтелі в місці переходу циліндра в торець, тому другий параметр P не заданий. Якби потрібно було залишити припуск для галтелі, наприклад радіусом 2мм, то в кадрі № 22 ми б записали P 200. І тоді б залишився не зрізаний куточок заготовки у нижній частині 2мм зведеним нанівець діаметр 65мм.

### 2.7.5 Багатопрхідний цикл підрізування торця G78

Схема рухів цього циклу показана на рисунку 2.12.

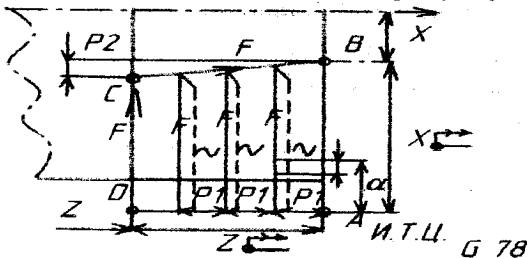
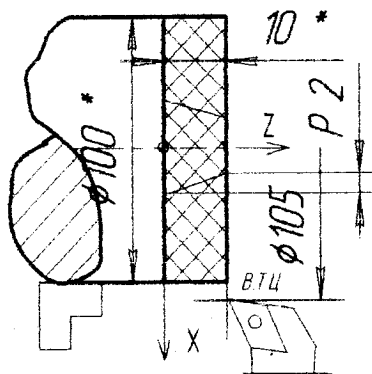


Рисунок 2.12 - Схема рухів формотвірної точки різця при зніманні припуску в циклі G78

Тут усе абсолютно ідентично циклу G77, тільки рух різця при зрізанні припуску відбувається в напрямку, перпендикулярному до осі обертання деталі. Вихідна точка по осі Z – торець заготовки, по осі X – зазор, що залежить від биття заготовки.

Приклад застосування циклу наведений на рисунку 2.13

Фрагмент програми



```

N25 X10500 * ~
N26 Z1000
N27 G78 * ~
N28 X-60 *
N29 Z0 *
N30 F1500 *
N31 P250
  
```

Рисунок 2.13

Якби було потрібно залишити конічну цапфу (як показано тонкими лініями), то ми б задали  $P_2$  і за адресою X записали діаметр меншої основи конуса.

### 2.7.6 Цикл глибокого свердління G73

Схема рухів циклу показана на рисунку 2.14.

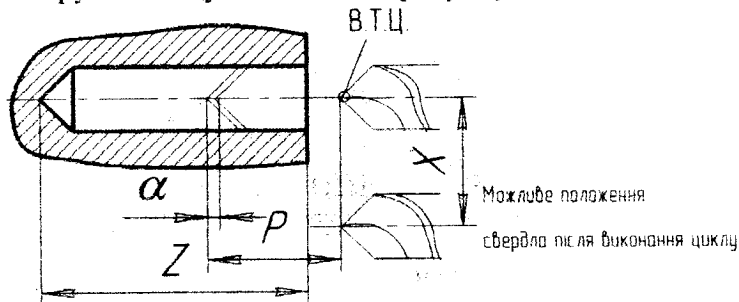


Рисунок 2.14 – Схема рухів свердла у циклі G73

Величина зазору « $\alpha$ » при поверненні свердла в отвір задається параметром № 8 за адресою P.

У циклі величина P – глибина одного свердління – повинна задаватися обов'язково і вона не дорівнює 0. Вихідна точка циклу – вісь обертання по X і зазор 1 – 2 мм від торця заготовки по осі Z. Приклад використання циклу G73 показаний на рисунку 2.15.

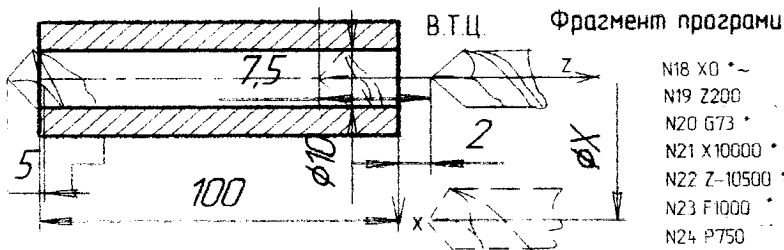


Рисунок 2.15

У прикладі за адресою X нами задане місце відведення свердла по осі X після виконання циклу. Якщо за адресою X записати нуль, то свердло, вийшовши з отвору після виконання циклу, залишиться у вихідній точці циклу по обох осях.

### 2.7.7 Цикл прорізання концентричних торцевих канавок G74

Схему рухів циклу бачимо з рисунка 2.16.

Структура програми циклу: ~ G74, X, (X.....),  
Z, (Z.....), F....., F.....,

де ~ – ознака того, що рух на шаг канавки буде відбуватися на прискореному ході. Якщо цей знак не заданий, різець буде переміщатися на робочій подачі.

Величина P задається на радіус у збільшеннях (тобто дискрета 0,01 мм).

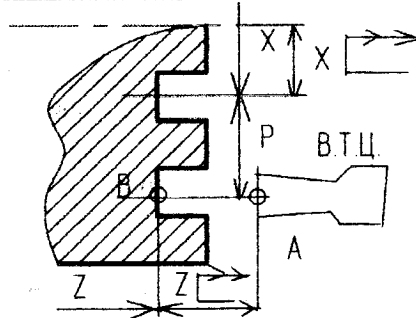


Рисунок 2.16

Вихідна точка циклу – це по осі X координата першої канавки (за порядком прорізання), по координаті Z повинний бути зазор безпеки 1 - 2мм. Виконавши цикл, машина поверне різець у вихідну точку циклу.

Цей цикл можна використовувати для підрізування торця, якщо замість кроку канавок задати глибину різання. Від циклу G78 він буде відрізнятися тільки напрямком знімання припуску. Тут припуск буде зніматися рухом різця уздовж осі Z, а в циклі G78 – у напрямку осі X.

### 2.7.8 Цикл прорізання канавок з рівним кроком на циліндричній поверхні деталі G75

Структуру рухів цього циклу бачимо на рисунку 2.17.

~ G75, X...(X.....),F...,P...

Крок канавки P не повинен бути більше величини Z. Вихідна точка циклу по осі X – зазор безпеки 1-2мм, по осі Z – координата першої канавки. Виконавши цикл, машина поверне різець у вихідну точку циклу.

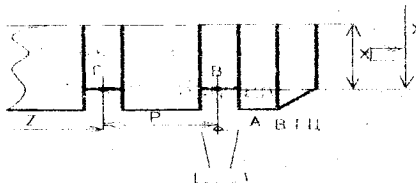


Рисунок 2.17

## 2.8 Спеціальні команди

### 2.8.1 Безумовний перехід Р

Ця команда використовується для зміни порядку виконання програми, указання початкового кадру програми, що у даний момент використовується для обробки деталі або для того, щоб зациклити програму при обкатуванні верстата.

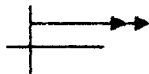
Розглянемо порядок задання цієї команди для перелічених випадків.

Наприклад, при налагодженні програми виявилось, що між 21 і 23 кадрами треба додати кілька додаткових кадрів, але поруч вільних комірок пам'яті немає. Є вільна ділянка пам'яті з комірки 100 по комірку 120. Тоді

початковий варіант програми:

N 20 X 10000 ~  
N 21 Z 20000 ~  
N 22 X 15000 ~

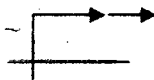
N 23 Z 1000 ~  
N 24 X 2500 ~



відредагований варіант:

N 20 X 10000 ~  
N 21 Z 20000 ~  
N 22 P 100—команда на виконання кадру №100

N 23 Z 1000 ~  
N 24 X 2500 ~



N 100 X 1500 ~  
N 101 Z 1300 ~  
N 102 X 1000  
N 103 P 23—команда на виконання кадру N 23

Коли КП короткі, то в пам'яті машини можна розмістити їх кілька штук. Наприклад, програма на деталь А займає комірку з 1 по 70; на деталь Б - з 75 по 120; на деталь В - з 125 по 195; на



деталь Г- з 200 по 249. Таким чином, у пам'яті зберігається 4 програми.

При цьому кадр № 0 програмами не займається. Коли потрібно, наприклад, обробляти деталь Б, то в нульовий кадр при налагодженні верстата запишемо Р 75. А тому що зчитування завжди починається з нульової комірки, то впливає безумовний перехід до комірки 75, де починається потрібна КП.

Безумовний перехід можна, як сказано вище, використовувати для зациклення програми. Наприклад:

N 10 X 1000 ~

N 11 Z -5000 ~

N 12 X-1000 ~

N 13 Z 5000 ~

N14 P10-команда на виконання 10-го кадру. Рух по прямокутнику 10 на 50 мм буде виконуватися доти, поки оператор не натисне кнопку «СТОП».

### 2.8.2 Повтор частини програми за функцією G 25

Якщо деталь має кілька ідентичних елементів, то не потрібно щоразу ці елементи програмувати. Досить написати програму обробки першого елемента, а всі інші подібні елементи обробляти, повторюючи необхідне число раз типову ділянку програми.

Функція G 25 має формат:

**G 25 \***

**P 000 000 \***

**P 00**

Перші три розряди - це початковий кадр повторюваної ділянки програми, другі три розряди - кінцевий кадр повторюваної ділянки програми.

**P 00** - кількість повторень ділянки програми.

Якщо ділянка повторюється один раз, то другий параметр не задають. У середині цієї ділянки може знаходитися команда G 25. Така процедура називається «вкладенням». Число таких вкладень не може бути більше 7.

Розглянемо приклад обробки корпусу золотника, показаний на рисунку 2.18.

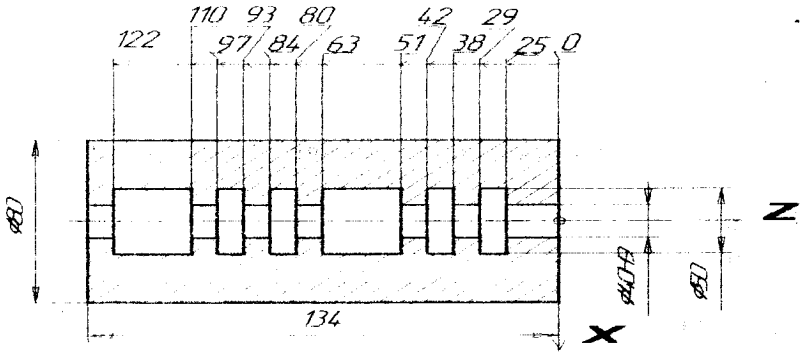


Рисунок 2.18-Корпус золотника

Керуюча програма:

N 0 T 1-свердло Ф 38 мм.

N 1 M 3

N 2 M 42

N 3 G 97

N 4 S 150

N 5 G 95

N 6 F 30

N 7 X 0 ~ \*

N 8 Z 200

N 9 Z - 14500

N 10 Z 200 ~

N 11 X 10000 ~ \*

N 12 Z 20000

N 13 T 2-розточувальний різець

N 14 G 96 \*

N 15 P 900 \*

N 16 P 50 \*

N 17 S 90

N 18 F 25

N 19 X 3950 ~ \*

N 20 Z 100

N 21 Z – 14000

N 22 X 3700 ~

N 23 Z 200 ~

N 24 X 10000 ~ \*

N 25 Z 20000

N 26 T 3-різець внутрішній, канавковий із шириною різальної кромки 4 мм

N 27 S 80

N 28 X 3800 ~ \*

N 29 Z 100

N 30 Z – 2900 ~

N 31 F 7

N 32 X 1200

N 33 X – 1200 ~

N 34 Z – 1300 ~

N 35

N 36

N 37 G 25 \*

N 38 P 32 034

N 39 X – 1200

N 40 X 1200 ~

N 41 Z – 300 ~

N 42

N 43

N 44 G 25 \*

N 45 P 39 041 \*

N 46 P 3

N 47

N 48

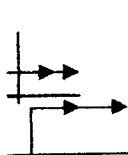
N 49 Z – 8400 ~

N 50 G 25 \*

перехід у

N 51 P 32 034\*

N 52 P 2



Розточення другої канавки і перехід у широкую канавку ( $42+13=55$ )

Один прохід при розточуванні широкої канавки

Повторення для формування широкої канавки ( $63-51=12$ )

Розточування двох вузьких канавок

початок розточування широкої канавки

N 53

N 54

N 55 G 25 \*

N 56 P 39 041 \*

N 57 P 4

N 58 Z 200 ~

N 59 X 10000 ~ \*

N 60 Z 20000

N 61 T 4-розвертка Ф 40 Н9

N 62 G 97

N 63 S 50

N 64 F 150

N 65 X 0 ~ \*

N 66 Z 200

N 67 Z -14000

N 68 Z 200 ~

N 69 X 10000 \* ~

N 70 Z 25000

N 71 M 5

N 72 M 30

} Розточування другої широкої канавки  
(4x3=12мм)

### 2.8.3 Витримка часу

Витримка часу задається функцією G04, що має такий формат:

G04 \*

P.....

де P – витримка часу в сотих частках секунди.

Максимальне число, яке можна записати за адресою P = 32767.

## 2.9 Підпрограми

### 2.9.1 Функція G 23

За допомогою цієї функції можна викликати з відповідної зони пам'яті підпрограму, що істотно скорочує обсяг основної програми і спрощує впровадження програми.

Функція має такий формат:

G 23 \*

P 00 000

Два старших розряди адреси Р указують зону пам'яті, у якій занесена підпрограма, наступні три розряди є адреси початку підпрограми. Кінець підпрограми програмується М 17.

Тут також можливі «вкладення», і їх кількість не може бути більше 7.

Розглянемо приклад на застосування функції G 23. На рисунку 2.19 наведений ескіз налагодження. Обробка ведеться в два прийоми: спочатку чорнова, потім - чистова. Припуск на чистову обробку залишимо таким: 0,5<sub>мм</sub> по діаметру і 0,2<sub>мм</sub> на торцеві поверхні.

Чорнову обробку оформимо як підпрограму.

### Керуюча програма:

N 0 T 1-прохідний різець

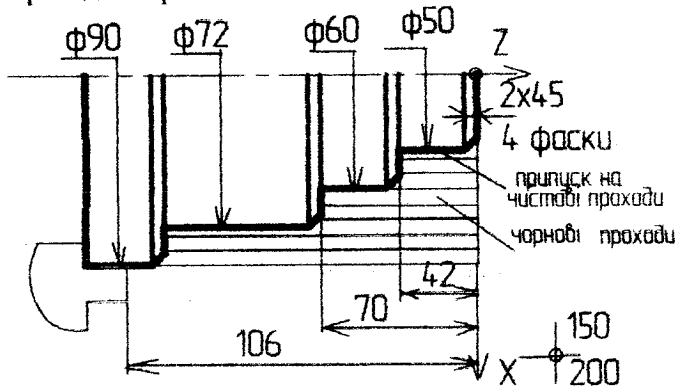
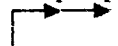
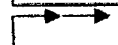

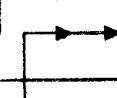


Рисунок 2.19

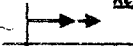
N 1 M 3  
 N 2 M 4 2  
 N 3 G 9 6 \*  
 N 4 P 9 0 0 \*  
 N 5 P 5 0 \*  
 N 6 S 9 0  
 N 7 G 9 5  
 N 8 F 2 5  
 N 9 X 9 2 0 0 ~ \*

- N10Z0  
 N 11 X -60  
 N 12 Z 1 ~  
 N 13 X 9060 ~  
 N 14 G 23 \* } Звертання до підпрограми, що знахо-  
 N 15 P 01 000 } диться в зоні 1 і починається з комірки 0  
 N 16 X 15000 ~ \* }  
 N 17 Z 20000  
 N 18 T 2-чистовий різець  
 N 19 S 110  
 N 20 F 7  
 N 21 X 4400 ~ \*  
 N22Z100  
 N 23 X 4200 - 45°  
 N 34 Z -4200  
 N 25 X 5600  
 N 26 X 6000 -45°  
 N 27 Z -7000  
 N 28 X 6800  
 N 29 X 7200 -45°  
 N 30 Z -10600  
 N 31 X 8600  
 N 32 X 9200 -45°  
 N 33 X 15000 ~ \*  
 N 34 Z 20000  
 N 35 M 30

**Підпрограма, що знаходиться в зоні 1**

- N 0 X -300 ~  устанавлення глибини різання 1,5мм  
 N 1 Z -10680  проточування на довжину 106.8мм  
 N 2 X 200 ~  відскакування на 1мм  
 N 3 Z 10680 ~  повернення до торця

Розділ 2 Програмування обробки деталей на токарному верстаті із системою керування «Електроніка НЦ - 31»

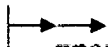
N 4 X -500 ~  заглиблення на другий прохід ( 2,5мм )  
N 5 ( відскакування 1мм і глибина різання 1,5мм )

N 6


N 7 G 25 \* повторення ділянки підпрограми з 0-го по 4-й кадр

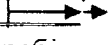
N 8 P 0 004 \*

N 9 P 5

N 10 Z -7080  прохід на довжину 70,8мм із глибиною 1,5мм

N 11 X 200 ~  відскакування на 1мм

N 12 Z 7080 ~  повернення до торця

N 13 X -500 ~  заглиблення на новий прохід ( 2,5мм )

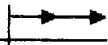
N 14 пробіл

N 15


N 16 G 25 \* повторення ділянки підпрограми з 10-го по 13-й

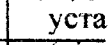
N 17 P 10013 \* кадр

N 18 P 3

N 19 Z -4280  проточування на довжину 42,8мм

N 20 X 200 ~  відскакування на 1мм

N 21 Z 4280 ~  повернення до торця

N 22 X -550 ~  установлення нової глибини різання 2,25мм  
N 23 (відскакування 1мм + глибина різання 1,25мм)

N 24

N 25 G 25 \* повторення підпрограми з 19-го по 22-й кадр

N 26 P 19 022 \*

N 27 P 2

N 28 M 17

## 2.10 Зсув нуля прив'язки G 92

Формат команди:

G 92 \*

X.....( X.....  ) \*

Z.....(Z..... )



Деякі деталі мають кілька конструкторських баз, тому розміри деталі задаються від декількох поверхонь. При розробленні КП, щоб не робити перерахування розмірного ланцюга, зручніше поміняти нуль системи координат деталі. Ось для цієї мети і використовується функція G 92.

У форматі X - зміщення нового нуля відносно колишнього. Для токарних деталей це, як правило, нуль Z - положення нового нуля відносно колишнього по осі Z. Розмір може задаватися в абсолютних координатах і в збільшеннях.

Після відпрацювання ділянки програми щодо нового нуля треба повернути прив'язку на колишнє місце повторним заданням G 92.

Приклад деталі, що вимагає зсуву нуля прив'язки, показаний на рисунку 2.20.

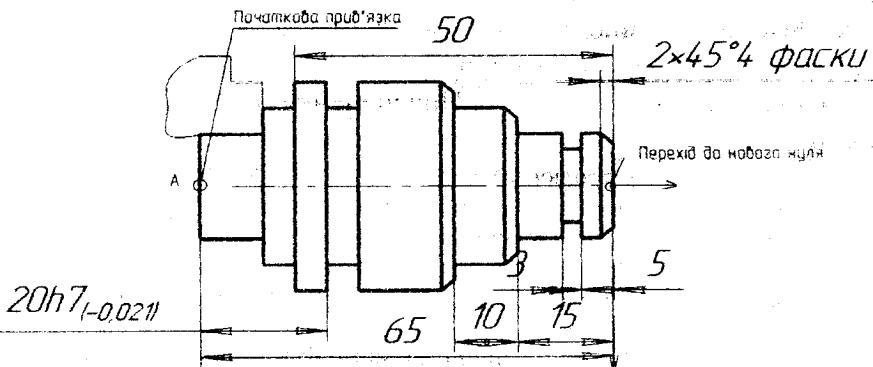


Рисунок 2.20

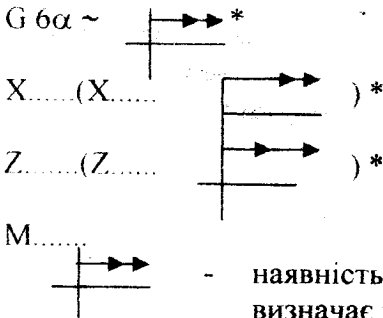
Якщо в програмі працює кілька різців, то при установці кожного різця для нього треба задавати функцію G92.

Наприклад, для приведеної деталі спочатку зміщається прохідний різець, потім - канавковий.



## 2.11 Група циклів умови руху

Ці цикли мають такий формат:



- наявність цього знака в заголовку циклу визначає роздільне виконання даного циклу і наступного за ним у покадровому режимі;

-- наявність цього знака обумовлює рух на швидкому ході.

Відсутність - переміщення на робочій подачі;

X і Z - координати кінцевої точки руху або збільшення до неї;  
 $\alpha$  - код циклу набуває значення від 1 до 7, визначає умову виконання руху (таблиця 2.2);

M-функції, які виконуються наприкінці руху.

Таблиця 2.2 - Цикли умови руху.

Найменування циклів	Призначення циклу
G 61	Контролює зупинник у кінцевій точці руху
G 62	Блокування клавіші «СТОП»
G 63	Контроль зупинника у кінцевій точці руху і блокування клавіші «СТОП»
G 64	Початок руху синхронізується з нульовою міткою шпинделя
G 65	Контроль зупинника в кінцевій точці руху і синхронізація початку руху з нульовою міткою шпинделя
G 66	Блокування клавіші «СТОП» і синхронізація руху з нульовою міткою шпинделя
G 67	Контроль зупинника в кінцевій точці руху, синхронізація початку руху з нульовою міткою шпинделя і блокування клавіші «СТОП»

Як приклад використання циклу групи G60, розглянемо нарізування торцевої канавки (рисунок 2.21).

### 2.11.1 Програмування з використанням циклів умови руху

Головна програма:

N 100 X 20500\*~

N 101 Z 1000

N 102 T 4

N 103 M 3

N 104 M 41

N 105 G 96

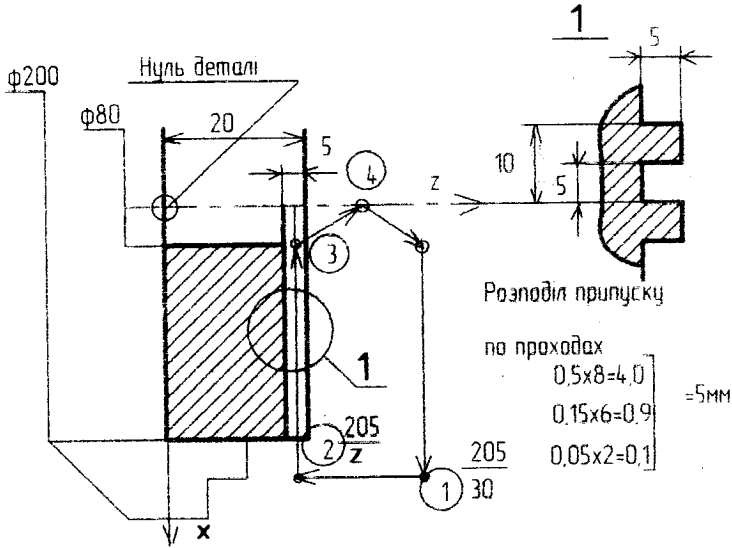


Рисунок 2.21

N 106 P 315

N 107 P 80

N 108 S 60

N 109 F 10

N 110

N 111 Z -35

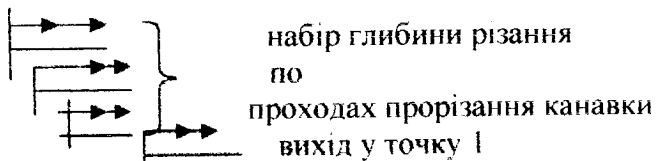
N 112 Z -10

N 113 Z -5

N 114 Z -1000

N 115 G 66 \* - синхронізація з нульовою міткою шпинделя

N 116 X 8000—перший прохід із глибиною 0,5 мм

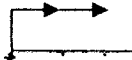


N 117 G 63 \* - контроль зупинника в кінцевій точці руху

N 118 X 0 \*

N 119 Z 600

N 120 X 20500 \* ~ відхід у точку 1 (N 119+N 121=Z 10 мм)



N 121 Z 400

N 122 G 25 \* -

N 123 P 111 122 \*

N 124 P 7

здійснити прохід із глибиною різання 0,5 мм 7 разів, тому що в оригінальних кадрах пройшли вже 1 раз, тобто зняли 0,5мм, при повторенні ділянки програми залишилося зняти  $7 \times 0,5 = 3,5$  мм. А всього зняли 4мм.

N 125 G 25 \* -

N 126 P 112 122 \*

N 127 P 6

N 128 G 25 \* -

N 129 P 113 122 \*

2 проходи з  $t=0,05$ мм

таким чином,

$0,5 + 0,5 \times 7 + 0,15 \times 6 + 0,05 \times 2 = 5$ мм

N 130 P 2

N 131 M 5

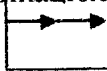
N 132 M 30

## Розділ 3

## Параметричне програмування для керуючої машини «Електроніка НЦ - 31»

Раніше ми розглянули функції G23 і G25, за якими ми могли викликати підпрограму або багаторазово повторити яку-небудь ділянку програми. Це усі випадки, коли підпрограма або ділянка програми має тверду послідовність і постійну сукупність команд. Команди, що ми розглянемо в даному розділі, дозволять організувати підпрограми або ділянки програми таким чином, щоб була можливість їх перебудовувати за деякими параметрами, як геометричними, так і технологічними.

Параметричне програмування здійснюється за допомогою адреси S з комбінацією таких знаків:

+45°, -45°, ~, 

Розглянемо застосування цих команд.

## 3.1 Команда пересилання

Команда пересилання має такий формат:

□ S $\alpha$ ,  $\beta$ ,

де  $\alpha$  - трирозрядне десяткове число, що визначає номер кадру, з якого вибирається команда;

$\beta$  - трирозрядне десяткове число, що визначає номер кадру, куди треба відправити обрану команду.

Приклад:

N10 Z - 250

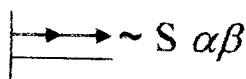
.....  
N50 Z 300

.....  
N100 ~ S 010050

Після виконання кадру №100 зміст кадру №10 буде записано в комірку № 50 і її зміст буде мати вигляд N50 Z - 250.

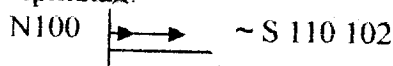
## 3.2 Команда додавання

Дана команда має вигляд



Суть цієї команди зводиться до такого: зміст кадру, записаного за адресою  $\beta$ , алгебраїчно підсумовується із змістом кадру, записаного за адресою  $\alpha$ , і сума записується в комірку  $\beta$ .

Приклад:



N101 Z 20000 ~

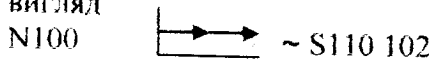
N102 X 10000 ~

N103 F20

.....  
.....

N110P-1000

Після виконання кадру №100 ця область команд буде мати вигляд



N101 Z 20000 ~

N102 X 9000

N103 F 20

.....  
.....

N110 P -1000

Таким чином, за командою кадру №100 відбулося підсумовування:  $10000 + (-1000) = 9000$ , і це число записане в кадр №102.

Застосування даної команди далі наприкінці розділу буде показано на прикладі програми обробки дисків з одного боку і їх відрізування від заготовки, розрахованої на декілька деталей. Після відрізування чергового диска відбуваються повернення до початку програми і підсумовування змісту першого кадру з

другим. А в першому кадрі за адресою Р записаний крок зсуву прив'язки з урахуванням ширини відрізаної деталі і відрізного різця. У другому кадрі спочатку записується  $Z = 0$ , і обов'язково збільшеннях.

### 3.3 Команда непрямого відпрацювання

Ця команда має формат  $\pm 45^{\circ} S \alpha 000$ .

Суть даної команди полягає в тому, що зміст кадру, записаного за адресою  $\alpha$ , записується в той кадр програми, що записаний за цією адресою. Якщо є знак  $- 45^{\circ}$ , то записується величина змінює при цьому свій знак на протилежний.

Розглянемо фрагмент деякої програми:

N9 F 150

N10 X1000  + 45<sup>0</sup>

N20 P 200215

N21 P 216223

N100 - 45<sup>0</sup> S 10 000

N101 G25 \*

N102 +45<sup>0</sup> S 20 000 \*

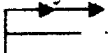
N103 P5

N104 G25 \*

N105 +45<sup>0</sup> S 21 000

N106 +45<sup>0</sup> S 9 000

Після виконання кадру № 100 буде узятий кадр № 10, і його числова частина змінить знак на протилежний. Верстат обробить фаску під кутом  $45^{\circ}$ , але в протилежному початковому напрямку, тому що в кадрі № 10 буде записане

X - 1000 

У кадрі № 101 задане повторення ділянки програми, початковий і кінцевий кадри якого записані у кадрі № 20. І повторити цю ділянку треба 5 разів. Далі в кадрі № 104 також задане повторення ділянки програми, межі якої записані в кадрі № 21. Але ця ділянка використовується тільки один раз.

Виконання кадру № 106 означає підстановку величини подачі, записаної в кадрі № 9, у кадр № 106.

А тепер, з урахуванням вивчених команд, переробимо програму нарізання торцевої різі, наведену на сторінці 49 у розділі 2.

Програму обробки поділимо на три функціональні частини.

Перша частина (з кадру № 1 по кадр № 15) буде виконувати технологічні функції: установлення режимів обробки і виклик підпрограми обробки спіральної канавки на торці деталі.

Друга частина (з кадру № 50 по кадр № 59) буде підпрограмою обробки спіральної канавки.

Третя частина (з кадру № 20 по кадр № 38) забезпечує установлення глибини різання по проходах.

N1 G95

N2 T4

N3 M3

N4 M42

N5 S300

N6 F1000

N7 X20500 \* ~

N8 Z3000

N9 ~ S 38050 – установлення початкової адреси таблиці

N10 G25 \* - виклик циклу нарізання спіральної канавки

N11 P50057 \* - початок і кінець ділянки програми прорізки канавки

N12 P 16 – кількість проходів для прорізки канавки

N13 M30

N20 Z - 50

N21 Z - 50

N22 Z - 50

N23 Z - 50

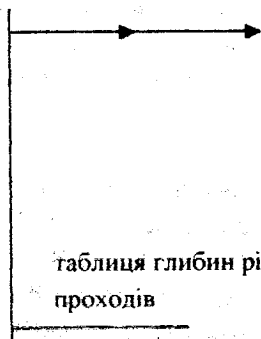
N24 Z - 50

N25 Z - 50

N26 Z - 50

N27 Z - 50

N28 Z - 15



N29 Z - 15

N30 Z - 15

N31 Z - 15

N32 Z - 15

N33 Z - 15

N34 Z - 5

N35 Z - 5

N36 Z - 1000 ~

N37 S 1000 – інкремент номера рядка таблиці

N38 +45<sup>0</sup> S2000 – початкова адреса таблиці – кадр № 20

N50 – комірка для установлення поточного значення глибини різання

N51 G66 \*


N52 X 8000 \*

N53 X0 \*

N54 Z 600

N55 X 20500 \* ~

N56 Z 400

N57  ~ S 37 050 – збільшення вмісту лічильника рядка таблиці на одиницю.

Роботу з цієї програми можна описати в такий спосіб. Після виходу інструмента в точку Г вибирається з таблиці глибини різання перша величина 0,5мм (кадр № 20), записується в комірку № 50 і за циклом, записаним у кадрі № 51, робиться перший прохід. Після цього командою, записаною у кадрі № 57, вміст лічильника проходів збільшується на одиницю (комірка № 37). Система зсуває рядок таблиці в комірку № 21 і робиться другий прохід. Так відбувається доти, поки лічильник рядків не дійде до останньої комірки № 35, тобто поки ділянка програми № 50 - № 57 не повториться 16 разів.

### 3.4 Параметричний виклик підпрограм за функцію G21

Функція параметричного виклику підпрограм G21 має формат

G21 X.....Z.....F.....P<sub>1</sub>.....P<sub>2</sub>.....P<sub>7</sub>



Тут параметр  $P_1$  має те саме значення, що й у функції G23. Тобто перші дві цифри після адреси – це зона пам'яті, де розміщується підпрограма, а три наступні цифри – це адреса початкової комірки пам'яті, де розміщується підпрограма.

Виклик підпрограми за функцією G21 передає переписування усієї пропозиції (від G21 до  $P_7$ ) у зону пам'яті, де знаходиться викликувана підпрограма. При цьому початкова адреса комірки, де розміститься ця пропозиція, задається параметром № 59 групи T.

Роботу функції G21 розглянемо на прикладі керуючої програми для обробки деталі, показаної на рисунку 3.1.

За допомогою функції G21 ми організуємо цикл обробки канавки, що буде поділений на чорнову обробку за циклом G75 і чистову обробку по контуру. Параметр №59 групи T дорівнює 0.

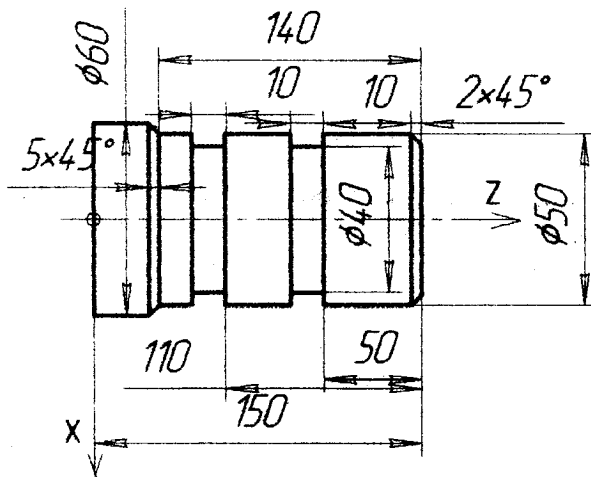


Рисунок 3.1

Керуюча програма:

N0 T1 установлюємо прохідний різець

N1 M3

N2 M 42

N3 G96 \* - умова сталості швидкості різання

N4 P1000 \*

N5 P100

N6 S100

N7 G95 – розмірність задання подачі, мм/об

N8 F30

N9 X 6040 \* ~ - вихід по двох координатах у вихідну точку

N10 Z 15100 } циклу G77

N11 G77 \* ~

N12 X-1000 \*  - знімається припуск 5мм на радіус,

N13 Z 1000 \* включаючи конус. У результаті залишається

N14 P 400 \* припуск 0,4мм на діаметр

N15 P 400 – величина скосу для формування конічної частини деталі

N16 X 4400 ~ вихід на початок фаски

N17 F 15

N18 X 5000 -45<sup>0</sup> - обробка фаски на торці деталі

N19 Z 1400 – чистова обробка циліндра до початку конуса

N20 X 6500 \* - обробка конуса

N21 Z 800

N22

N23 X 10000 \* ~

N24 Z 30000

N25 T2 { - канавковий різець із шириною різальної кромки 4мм,


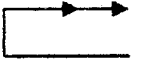
N26 S 70 } різець, прив'язаний по лівій вершині

N27 F 15

N28 Z 9550 \* ~ - вихід у початок першої канавки



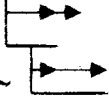

N29 X 5200

N30 G21 \*

N31 X - 1100\*  - значення X для циклу G75, перший параметр функціїN32 Z - 500 \*  - значення Z для циклу G75, другий параметр функції

N33 P 1050 \* - номер зони пам'яті й адреса, з якого починається підпрограма, третій параметр функції

N34 P400 \* - значення P для циклу G75 – крок канавок. Канавку прорізаємо шляхом двох проходів, при цьому залишається припуск на чистову обробку по 0,5 мм по ширині з кожного боку канавки і 0,5мм на радіус (52-5,5+5,5=41), це четвертий параметр функції

N35 X - 1200*		- переміщення по X для чистового проходу (6мм) при обробці канавки, п'ятий параметр функції
N36 Z 550 ~ *		- вихід по осі Z у початкову точку обробки канавки, шостий параметр функції
N37 Z - 600		} переміщення при чистовому проході обробки канавки, сьомий параметр функції
N38 Z - 6050 ~		
N39 G25 * -		} повторення ділянки програми з кадру №30 по кадр №37
N40 P 30 037		
N41 X 10000 * ~		
N42 X 30000		
N43 M5		
N44 M30		

Підпрограма, до якої ми звертаємося за функцією G21.

Підпрограма розміщується в першій зоні і починається з кадру №50.

N50 G 75 ~ \*

N51 +45<sup>0</sup> S 1000 \* - перший параметр функції G21, тобто X - 1100

N52 + 45<sup>0</sup> S 2000 \* - другий параметр, тобто Z - 500 ~

N53 + 45<sup>0</sup> S 4000 - четвертий параметр, тобто P400

N54 + 45<sup>0</sup> S 6000 - шостий параметр, тобто Z 550 ~

N55 + 45<sup>0</sup> S 5000 - п'ятий параметр, тобто X - 1200

N56 + 45<sup>0</sup> S 7000 - сьомий параметр, тобто Z-600

N57 - 45<sup>0</sup> S 5000 - знову п'ятий параметр, тобто X - 1200

N58 M 17

усі параметри тут у збільшеннях, знак яких тут не показаний.

Якщо буде потрібно обробити канавки з іншими розмірами, то досить у функції G21 поміняти параметри і працювати за запропонованою програмою.

### **3.5 Програмування умовних переходів**

Використовуючи команди умовних переходів, можна організувати виконання керуючої програми в довільному порядку. Крім того, ці команди дозволяють програмістові

створювати свої технологічні цикли, оптимізуючи, таким чином, процес обробки деталі.

### 3.5.1 Перехід за зовнішньою умовою

Формат даної команди  $\sim P \alpha$ ,

де  $\alpha$  - трирозрядне десяткове число – номер кадру, куди повинне перейти виконання програми при виконанні деякої умови. Якщо ця умова не виконується, то виконується наступний кадр програми. Припустимо, деталі виготовляються з прутка з автоматичною подачею останнього. Після того як пруток дороблений до визначеної довжини, потрібно подати сигнал операторові на його заміну. Наявність прутка контролюється датчиком, що спрацьовує при довжині прутка меншій, ніж довжина деталі. Керуюча програма буде такою:

N0..... кадри програми з №0 по №100 - це програма обробки  
.....деталі

.....  
N100.....

N101  $\sim P 0$  - умовний перехід за зовнішньою умовою при спрацьовуванні датчика довжини прутка. Якщо пруток не закінчився, команда переходить у кадр №0 і програма відпрацьовується знову. Як тільки датчик довжини прутка спрацює на закінчення прутка, буде виконуватися кадр №102, де задано зупинення програми для заміни прутка.

N102 G 55

### 3.5.2 Умовний перехід за плюсом

Формат даної команди має вигляд  $+45^0 P \beta \alpha$ ,

де  $\beta$  - трирозрядне десяткове число, що визначає номер кадру програми, який аналізується командою умовного переходу.

$\alpha$  - трирозрядне десяткове число, що визначає номер кадру програми, куди потрібно перейти у разі виконання контрольованої умови.

Роботу цієї команди розглянемо на прикладі циклу свердління глибокого отвору. Відмітною рисою цього циклу буде: по – перше, свердло не виходить з отвору у вихідну точку по осі  $Z$ , а тільки відскакує від дна отвору на величину  $\delta$ ; а, по – друге, останній прохід буде відрізнятись від усіх попередніх

тим, що він буде менше і визначається довжиною частини отвору, що залишилася. (У циклі G73 усі довжини свердління однакові).

Структура рухів циклу показана на рисунку 3.2.

Цикл свердління оформимо як підпрограму, яку будемо викликати в основну програму за допомогою функції G21. Фрагмент програми, у якій йде звертання до підпрограми, має вигляд:

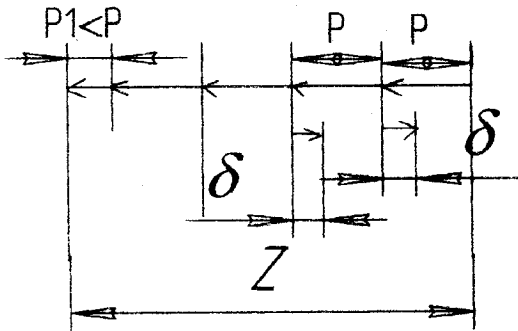


Рисунок 3.2 - Схема рухів свердла

N.....G21 \*

N.....Z.... повна довжина отвору

N.....P 1050 \* - номер зони пам'яті (1) і номер першого кадру підпрограми (№50)

N.....Z.....\* \* глибина одного свердління (P)

N.....Z.....~\* величина відскакування свердла ( $\delta$ )

N.....Z.....~ \* повернення у вихідну точку після цілком просвердленого отвору

Підпрограма циклу свердління має такий вигляд:

N50 S 1200 ~ - в комірку № 200 запам'ятовується глибина свердління (перший параметр функції G21)

N51 S 200259 ~ - ця сама інформація записується в комірку № 249

N52 S 3200 - тут запам'ятовується третій параметр функції, тобто глибина одного свердління

N53 +45 P 20058 - за допомогою переходу за плюсом виробляється перевірка глибини частини отвору, що залишилася. Якщо  $Z+Z_1 < 0$ , то

робиться ще один прохід на глибину  $Z_1$ . Якщо  $Z+Z_1>0$ , то відбувається свердління на величину невимірного залишку.

N54 - 45 S 3000 - команда «свердлити на величину одного проходу»  $Z_1$

N55 + 45 S 1000 - швидке відсакування від дна отвору на величину  $\alpha$

N56 - 45 S 4000 - повернення на швидкому ході в початок наступного свердління

N57 P 51 - перехід до виконання кадру № 51

N58 + 45 S 249000 - команда на досвердлювання невимірного залишку отвору

N59 + 45 S 5000 - встановлюється п'ятий параметр функції G21, тобто відбувається повернення на швидкому ході у вихідну точку циклу свердління

N60 M17 - кінець підпрограми і повернення в основну програму

### 3.5.3 Умовний перехід за мінусом


Команда умовного переходу за мінусом має формат

- 45 P.  $\beta\alpha$

Значення параметрів такі самі, як і в попередньому переході.

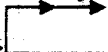
### 3.5.4 Перехід за нулем або плюсом

Команда умовного переходу за нулем або плюсом має

формат + 45  P.  $\beta\alpha$

Значення  $\beta$  і  $\alpha$  такі самі, як і в попередніх двох переходах.

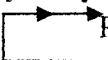
### 3.5.5 Перехід за нулем або мінусом

Формат команди - 45  P.  $\beta\alpha$

Значення всіх параметрів аналогічні до попереднього переходу.

### 3.5.6 Перехід за нулем лічильника

Команда переходу за нулем лічильника має формат

~  P.  $\beta\alpha$ ,

де  $\beta$  - трирозрядне число, що визначає номер комірки, яка служить лічильником;

$\alpha$  - трирозрядне число, що вказує номер комірки, куди необхідно перейти при виконанні програми, якщо в лічильнику ще не нуль.

Розглянемо застосування даної команди на прикладі.

На рисунках 3.3 і 3.4 наведені ескізи двох деталей, що відрізняються тільки числом канавок. У одній їх три, а в іншій - 5. Використовуючи команду «перехід за нулем лічильника», ми за однією керуючою програмою обробимо обидві деталі.

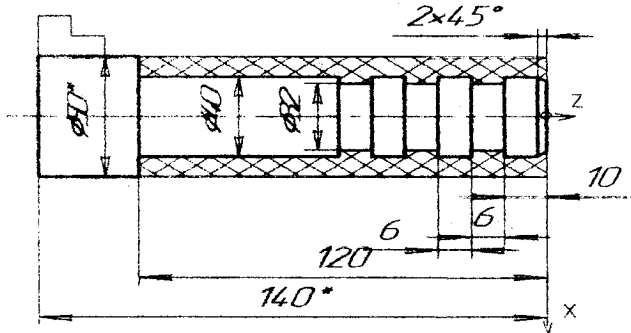


Рисунок 3.3

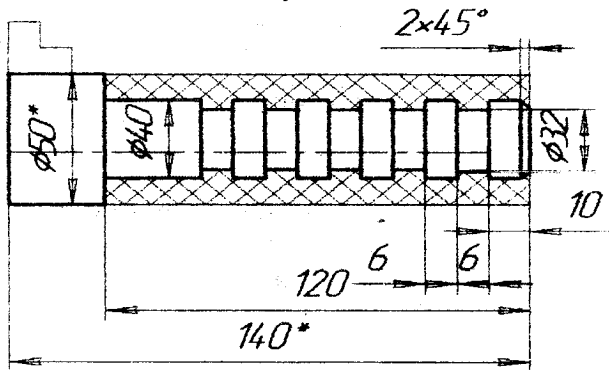


Рисунок 3.4

N0 T1 - установлюємо прохідний різець

N1 M3

N2 M42

N3 G96 \* - умова сталості швидкості різання

N4 P 1000 \*

N5 P 100

N6 S 100

N7 G 95 – функція задання подачі, мм/об

N8 F 15

N9 X 5000 \* ~ - вихід по двох координатах у початкову точку

N10 Z 100            циклу

N11 G 77 \* ~            } - багатопрхідний цикл зовнішнього обточування з припуском по довжині і радіусу 0,2мм

N12 X 4040 \*

N13 Z - 11980 \*

N14 P 300

N15 X 3400 ~ - підхід для зняття фаски

N16 X 4000 -45° - знімається фаска під кутом 45°

N17 F 10 – подача для чистового проходу

N18 Z - 12000 – чистовий прохід – обточування Ф40 на довжину 120мм

N19 X 5400 – чистове підрізування торця

N20 X 8000 \* ~            } - відвід різця у вихідне положення для зміни інструмента

N21 Z 15000

N22 T 2            } -- установлюємо канавковий різець із шириною різальної

N23 S 90            }            кромки 4мм і задаємо швидкість різання 90м/хв

N24 F 20

N25 ~ S 249200 – запис в комірку № 200 числа, що знаходиться в комірці № 249, куди воно заноситься з пульта оператором перед пуском програми.

N26 X 4200 \* ~            }            вихід по двох координатах у точку прорізки

N27 Z - 800            }            першої канавки, але не доходимо до канавки 2мм

N28 Z - 600 ~            }            - у збільшеннях задаємо вихід у точку прорізки

першої канавки (№27+№28=8+6=14мм), це дорівнює відстані до правої кромки канавки 10мм і плюс ширина різця 4мм. Різець прив'язаний до нуля деталі по правій вершині

N29 G 75 \* ~            }            - цикл обробки канавок на циліндрі

N30 X - 800 \*            }            - глибина канавки на радіус (4мм)

N31 Z - 200 \*            }            - зсув на 2мм, щоб різцем із шириною різальної кромки 4мм проточити канавку шириною 6мм

N32 P 200 – крок канавок 2мм, щоб одержати ширину 6мм

N33 ~            }            P 200028 – команда умовного переходу. Поки



вміст лічильника, роль якого відіграє комірка № 200, не зрівняється з нулем, тобто поки не прорізано задане число канавок, програма буде повертатися до кадру № 28 і виконувати цикл прорізки чергової канавки

N34 X 8000 \* ~ - вихід різця у вихідне положення одночасно по двох координатах, коли буде прорізана остання канавка

N35 Z 15000

M30

Кількість прорізуваних канавок буде залежати від того, яке число записане в комірці № 249. Якщо в ній записати P3, то проріжеться три канавки. Якщо P5, то буде зроблено 5 канавок.

### 3.6 Приклад керуючої програми, розробленої з використанням методики параметричного програмування

Ескіз деталі наведений на рисунку 3.5. Деталь виготовляється із заготовки діаметром 72мм, що попередньо просвердлена свердлом діаметром 30мм. Довжина заготовки 120мм. Вона закріплена в трикулачковому патроні.

У налагодженні використовуються два підрізних різці – чорновий і чистовий, два розточувальних – чорновий і чистовий і один відрізний різець.

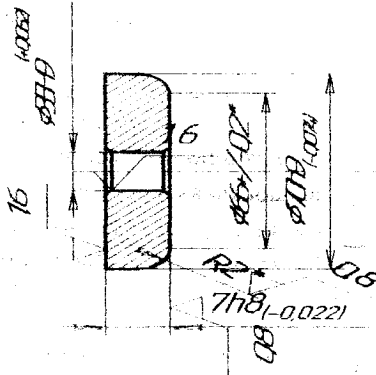


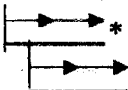
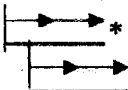

Рисунок 3.5

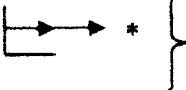
### Керуюча програма



N0 P3

N1 P 11 200 - розмір деталі плюс ширина різальної кромки різця

N2 Z0 → керуючий кадр, вміст цієї комірки буде використовуватися у всіх командах G92 – зміна нуля прив'язки

- N3 M40 – діапазон чисел обертів
- N4 T7 – чорновий підрізний різець
- N5 - резервна комірка
- N6 G92 \*                    } - зміна нуля прив'язки інструмента, при цьому по осі
- N7 X0 \*                     } X завжди буде 0, а по осі Z треба взяти вміст
- N8 S 2000 +45             } комірки № 2 і підставити на це місце
- N9 – резервна комірка – комірка без інформації
- N10 X 66000 \* ~         } - вихід чорнового різця в початок обробки на
- N11 Z 10000               } швидкому ходу по двох координатах,
- дискретність
- 0,001мм (дивись траєкторію на рисунку 3.6)
- N12 S600 – задана частота обертання шпинделя
- N13
- N14 F 150 задана подача 1,5 мм/об для підходу до поверхні заготовки
- N15 M3 – праве обертання шпинделя
- N16
- N17 Z2000 – підхід до поверхні обробки на подачі 1,5 мм/об
- N18 F20 – задана робоча подача 0,2 мм/об
- N19
- N20 X 66000 \* - врізання по двох координатах з подачею 0,2мм/об
- N21 Z0
- N22
- N23 G12 \*                   } - чорнова обробка галтели
- N24 X 4000                 } 
- N25 Z – 2000               } 
- N26
- N27
- N28 Z – 8000 – проточування зовнішньої поверхні на довжину 8мм
- N29 F 25
- N30 X 70500 -45° – вихід з контакту з оброблюваною поверхнею під кутом 45° з подачею 0,25мм/об.
- N31 Z – 12000 – обточування по верху до Z=- (7+5)мм для відрізного різця
- N32
- N33 F60
- N34 X 2000                 }  - відвід різця від поверхні на 1мм

- N35
- N36 Z 60000 ~ - відвід різця в зону зміни інструмента
- N37
- N38 G92 \* } - скасування прив'язки, вміст комірки № 2
- N39 X 0 } беремо з протилежним знаком. Як ми
- N40 S 2000 - 45<sup>0</sup> } покажемо далі, не завжди в комірці № 2 буде нуль
- N41
- N42
- N43
- N44
- N45 T8 - установили чистовий різець
- N46
- N47 G 92 \* - здійснюємо його прив'язку, узявши X=0, а Z - з комірки №2
- N48 X 0 \*
- N49 S 2000 + 45<sup>0</sup>
- N50
- N51 X 71000 ~ \* - } на прискореному ходу виходимо в точку початку
- N52 Z 10000 } обробки одночасно по двох координатах
- N53
- N54 F 150
- N55 S 900
- N56
- N57 Z - 2700 - на подачі 1,5 мм/об підводимо різець до поверхні
- N58 F 30
- N59 X 70300 \* } - з подачею 0,3 мм/об уріжемося по похилій прямій
- N60 Z - 2400 } у зовнішню поверхню циліндра діаметром 70,3мм
- N61
- N62 F 12
- N63 X 70000 - з подачею 0,12 мм/об уріжемося до діаметра 70мм
- N64
- N65 G 13 \* } - обробляємо галтель радіусом 2мм
- N66 X - 4800 }  \*
- N67 Z 0
- N68
- N69 X 34000 - обточуємо торець до отвору
- N70

- N71 X 2000  ~ \* } - відводимо різець у зону зміни інструмента
- N72 Z 70000
- N73
- N74 G 92 \* - \* } скасовуємо зміну нуля інструмента, беручи зміст  
N75 X 0 } комірки № 2 з протилежним знаком
- N76 S 2000 - 45<sup>0</sup>
- N77
- N78
- N79
- N80 T3 – установлюємо чорновий розточувальний різець
- N81
- N82 G92 \* \* } здійснюємо його прив'язку з урахуванням  
N83 X 0 } реального положення заготовки (дивлячись,  
N84 S 2000 + 45<sup>0</sup> } яка по черзі обробляється деталь)
- N85
- N86 X 36500 ~ \* - вихід на швидкому ході в початок роботи
- N87 Z 10000
- N88
- N89 F100
- N90 S 600
- N91
- N92 Z 6000 -- підхід з подачею 1мм/об до поверхні деталі
- N93 F 25
- N94 Z 500 -- з подачею 0,25 мм/об торкаємося торця деталі
- N95 F 20
- N96 X 34500 - 45<sup>0</sup> - на подачі 0,2 мм/об формуємо фаску
- N97 Z - 8000 -- і проточуємо начорно отвір на довжину до Z= - 8мм
- N98
- N99 X - 2000  ~ \* } - виводимо різець з отвору
- N100 Z 30000
- N101
- N102 G92\* \* } - скасовуємо зміну нуля прив'язки
- N103 X 0
- N104 S 2000 - 45<sup>0</sup>
- N105
- N106

N107

N108

N109

N110 T5 – установлюємо чистовий розточувальний різець

N111

N112 G 92 \*

} здійснюємо його прив'язку з урахуванням  
положення заготовки

N113 X 0 \*

N114 S 2000 + 45<sup>0</sup>

N115

N116 X 37000~ \* - вихід по двох координатах у початок обробки

N117 Z 10000

N118

N119 F 150

N120 S 800

N121

N122 Z 500 – на подачі 1,5 мм/об підходимо до торця за 0,5мм до нього

N123 F 14

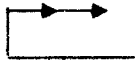
N124 X 35000 - 45<sup>0</sup> – знімаємо фаску з подачею 0,14 мм/об

N125 F 12

N126 Z – 7600 – обточуємо отвір начисто з подачею 0,12 мм/об

N127

N128 F 50

N129 X – 2000  + 45<sup>0</sup> – відводимо різець від поверхні

N130

N131 M5 – стоп шпиндель

N132 Z 200000~ - відвід різця в зону зміни інструмента

N133 M1 – зупинення програми з підтвердженням

N134

N135 G 92 \*

} скасування прив'язки чистового  
розточувального різця

N136 X 0

N137 S 2000 - 45<sup>0</sup>

N138

N139

N140 T 6 – установлюємо відрізний різець



N172 G 92 \*

N173 X 0 \*

N174 S 2000 - 45°

N175

N176

N177 S 1002

N178

N179

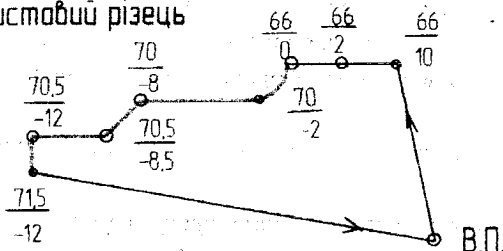
N180 M30

N181 P0 – безумовний перехід до кадру № 0

Операція, що виробляється в кадрі № 177, змінює точку прив'язки інструментів з урахуванням товщини відрізаної деталі - 7,2мм і ширини відрізного різця - 4мм.

Перший раз, коли встановлюється заготовка, всі інструменти прив'язуються до нульової точки, що розміщена на торці заготовки. І в кадрі № 2 знаходиться нульове значення зрушення по осі Z. В міру виготовлення деталей після кожного відрізування вміст комірки № 2 буде по команді додавання, заданої в кадрі № 177, збільшуватися на 11,2мм. Ця величина і буде розміром зсуву інструментів по осі Z.

Чистовий різець



Чорновий різець

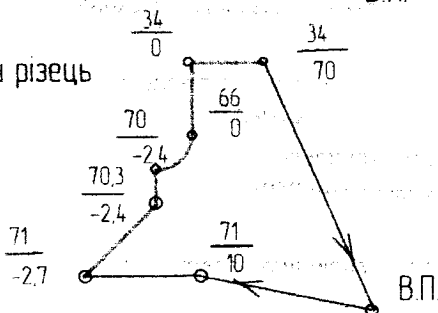


Рисунок 3.6 – Траєкторії руху прохідних різців

## Розділ 4

## Програмування обробки деталей на токарному верстаті із системою 2P22

## 4.1 Характеристика системи керування

4.1.1 Система належить до класу оперативних. Уведення програми здійснюється вручну з пульта оператора без зовнішніх програмоносіїв.

4.1.2 Візуальний контроль за введенням, коригуванням програм і їх відпрацьовуванням здійснюється за допомогою монітора.

4.1.3 Обсяг кадру програми не обмежений. Не можна лише повторювати в кадрі адреси.

4.1.4 Інформація в кадрі задається у тому вигляді, як вона записана в кресленні або технологічному процесі.

4.1.5 Геометрична інформація може бути задана як в абсолютних координатах, так і в збільшеннях.

4.1.6 Зміна інструмента здійснюється у вихідному положенні, куди супорт по команді «зміна інструмента» рухається автоматично роздільно по координатах. Спочатку йде рух по осі X, а потім - по осі Z. Вихідне положення вибирається оператором при налагодженні верстата і запам'ятовується системою керування на період обробки за даною програмою. Також відбувається переміщення супорта і по команді «кінець програми».

4.1.7 При коригуванні програми можна додавати або стирати будь-яку кількість кадрів програми. Виняток складає перший кадр, який коригувати не рекомендується. Тому в першому кадрі задається абсолютно достовірна інформація, наприклад, що подачі задаються в мм/об.

## 4.2 Адреси та їх призначення

A – дана адреса використовується в технологічних циклах і має різне значення (нахил різі в циклі L1, ширина канавки в циклі L 2, припуск на чистовий прохід у циклах L 8 і L 9).

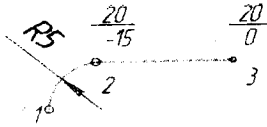
B – номер кадру, з якого починається опис креслення деталі або повторення ділянки програми.



- C – ця адреса також використовується в циклах (збіг різі у циклі L 1, розмір фаски під кутом  $45^0$ ).
- D – витримка часу.
- E – ознака швидкого ходу.
- F – робоча подача.
- G – службова функція.
- H – число повторень ділянки програми.
- L – номер технологічного циклу.
- M – допоміжна функція.
- N – номер кадру програми.
- P – параметр, що використовується в технологічних циклах.
- Q – радіус галтелі.
- R – радіус дуги менше  $90^0$ .
- S – частота обертання шпинделя.
- U – величина збільшення по осі X.
- W – величина збільшення по осі Z.
- X, Y – абсолютні координати точки по цих осях.

### 4.3 Службові функції

- G 05 – скасування гальмування наприкінці кадру, задається при зміні вигляду кривої для попередження утворення сліду переходу. Наприклад, спочатку йде переміщення по дузі, потім – по прямій, дотичній до цієї дуги (рисунок 4.1).
- G10 – задання умови сталості швидкості різання.
- G11 – скасування сталості швидкості різання.
- Ці дві команди повинні задаватися окремими кадрами програми.
- G12 – дозвіл на зміну інструмента в будь-якій точці робочого простору, а не тільки у вихідному положенні.
- G95 – подача у програмі задається в мм/об.
- G94 – подача задається в мм/хв.



N10 X 20 Z – 15 G 05  
N 11 Z 0

Після обходу дуги не відбудеться зменшення подачі і сліду переходу від дуги до прямої на деталі не буде видно.

Рисунок 4.1

#### 4.4 Допоміжні функції

- M00 – безумовна зупинка відпрацювання програми.
- M01 – зупинка програми з підтвердженням.
- M02 – кінець програми.
- M08 – ввімкнути подачу ЗОР.
- M09 – вимкнути подачу ЗОР.
- M17 – кінець опису креслення деталі.
- M18 – кінцевий кадр повторюваної ділянки програми.
- M20 – передати керування роботіві.

#### 4.5 Програмування подачі

Робоча подача програмується за адресою F і записується у тому вигляді, в якому вона задана в технологічному процесі. Наприклад, треба зробити обробку поверхні з подачею 0,25 мм/об. У програмі буде записано F 0,25. Ця подача машиною запам'ятовується до приходу іншої, записаної за цією адресою.

Прискорене переміщення супорта програмується ознакою швидкого ходу E. Ця команда діє тільки в межах того кадру, де вона записана.

#### 4.6 Програмування головного руху

Головний рух задається числом обертів шпинделя за адресою S. Вслід за адресою проставляється цифра номера діапазону чисел обертів (1,2,3). Потім пишеться розрахункове число обертів шпинделя. Якщо потрібно задати зворотне

обертання шпинделя, то перед числом обертів ставиться знак мінус.

Задане число обертів машиною запам'ятовується, до приходу іншого, заданого за цією адресою. Діапазон чисел обертів у даній програмі повинний бути один. Він оператором встановлюється рукояткою на коробці швидкостей верстата перед початком роботи з програми.

Приклад: потрібно задати 357 об/хв на другому діапазоні. У програмі буде записано: S 2 357. При записі S 2 – 357 буде відбуватися зворотне обертання шпинделя.

## **4.7 Програмування найпростіших переміщень**

### **4.7.1 Програмування зовнішнього обточування за один прохід**

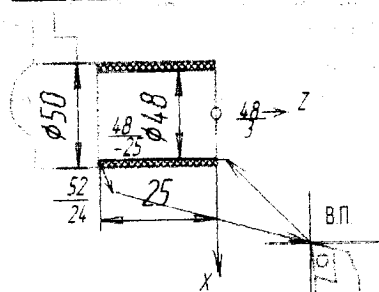
Приклад показаний на рисунку 4.2.

Як видно з наведеної програми, у першому кадрі ми записали абсолютно достовірну інформацію про те, що всі подачі, які задаються у програмі, будуть мати розмірність мм/об. Тому цей кадр не буде коригуватися. У другому кадрі вже задані режими різання, що можуть коригуватися. У третьому кадрі ми на прискореному ході підводимо різець до торця деталі на діаметр обробки 48мм із зазором по осі Z – 3мм.

Одночасно вмикаємо охолодження. У третьому кадрі задаємо координату кінцевої точки обробки по осі Z і машина перемістить супорт у задану точку. Далі ми запрограмували відвід різця від деталі по двох осях одночасно з подачею 1мм/об і вимкнули охолодження. В останньому, шостому, кадрі задаємо команду «кінець програми», за якою супорт автоматично відійде у вихідне положення роздільно по осях. Спочатку по осі X, потім - по осі Z.

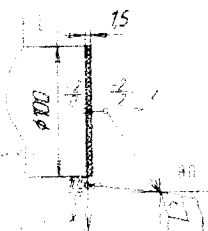
### **4.7.2 Програмування підрізування торця за один прохід**

На заготовці діаметром 100мм треба видалити 1,5мм припуску з торця деталі. Приклад задачі показаний на рисунку 4.3.



N1 G 95  
 N2 T 1 S 3 550 F 0,25  
 N3 X 48 Z 3 E M08  
 N4 Z - 25  
 N5 U 4 W 1 F1 M 09  
 N6 M02

Рисунок 4.2 - Однопрохідне зовнішнє обточування



N1 G95  
 N2 N2 S2 200 F 0,2  
 N3 X 105 Z 0 E M 08  
 N4 G10  
 N5 X -2  
 N6 G11  
 N7 W 1 F 1 M 09  
 M 02

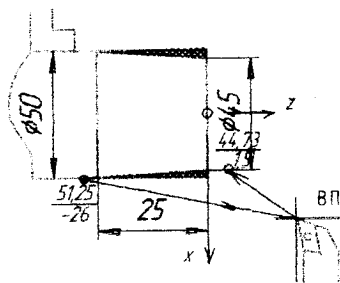
Рисунок 4.3 - Однопрохідне підрізування торця

Тут на додаток до попередніх команд показали задання сталості швидкості різання. Для цього ми підраховали оптимальну швидкість різання на периферії і задали необхідне число обертів. Тепер машина в міру руху до центра заготовки буде намагатися в розумних межах зберігати задану швидкість різання.

#### 4.7.3 Обточування конуса за один прохід

Для програмування обточування конуса ми повинні виконати попередні розрахунки діаметрів основ конуса на початку конуса й у його кінці (рисунок 4.4). Перший потрібний для утворення зазору безпеки при підході до заготовки, а другий

– для виходу різця після обточування конуса. Якщо не програмувати вихід різця лівіше більшої основи конуса, то на поверхні деталі у місці закінчення різання залишиться «деформаційний валик».



N1 G95  
 N2 T3 S3 650 F 0,07  
 N3 X 44,73 Z1,5 E M8  
 N4 X 51,25 Z -26  
 N5 M9  
 N6 M02

Рисунок 4.4 – Обточування конуса за один прохід

#### 4.7.4 Програмування обробки фасок під кутом $45^\circ$

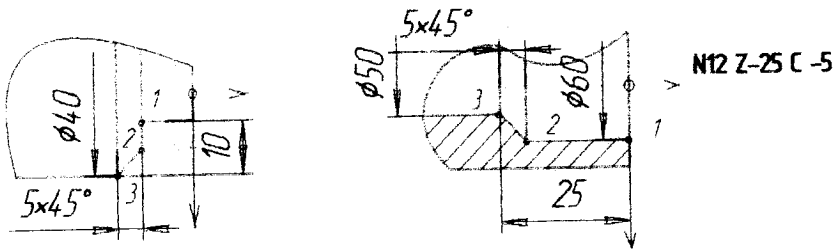
Фаски під кутом  $45^\circ$  програмуються одним кадром, у якому записуються одна координата кінцевої точки фаски і розмір фаски зі знаком за адресою C. При цьому, задаючи обробку фасок, треба виконувати чотири правила:

*Правило 1* Обробку фаски можна задати тільки рухом уздовж осі Z у її негативному напрямку.

*Правило 2* Знак фаски визначається тим, у якому напрямку по осі X рухається різець, обробляючи фаску.

*Правило 3* Якщо фаска є продовженням торця або циліндра, то в одному кадрі можна задати обточування торця (циліндра) і фаски.

*Правило 4* При програмуванні обробки фаски кінцева координата, яка записана в кадрі фаски, повинна бути такою самою, як і в попередньому кадрі програми. Тобто якщо в попередньому кадрі рух був по осі X, то й у кадрі фаски кінцевою координатою повинна бути вісь X. І навпаки, якщо рух був по осі Z, то в кадрі фаски повинна стояти координата кінцевої точки по осі Z.



N10 X 40 C 5, або N10 U 20 C 5

N12 Z-25 C -5

Рисунок 4.5 – Програмування обробки фасок

Як видно з рисунка, при обробці зовнішньої фаски різець знаходився в точці 1, і ми відразу задали прохід по циліндру й обточування фаски. У другому випадку при програмуванні внутрішньої фаски ми задали в одному кадрі обточування циліндра діаметром 60 мм і фаску розміром 5мм.

#### 4.7.5 Програмування обробки галтелей

Галтелі (дуги  $90^{\circ}$ ) програмуються одним кадром, у якому задаються одна координата кінцевої точки галтели і розмір галтели за адресою Q зі знаком. Усі чотири правила, записані для фасок, діють при програмуванні галтелей.

Приклади програмування обробки галтелей показані на рисунку 4.6.

На всіх рисунках точка 1 - це точка, з якої починається запрограмований рух з обробки галтели. Зверніть увагу і на можливості програмування напрямку обходу дуги: за і проти

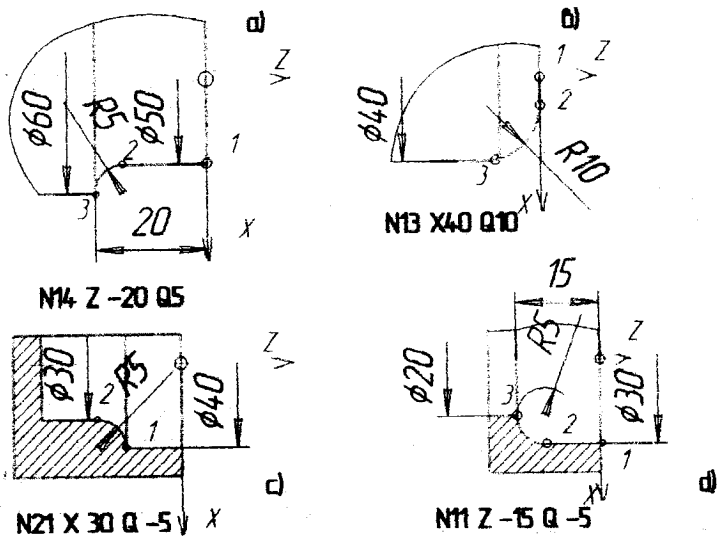


Рисунок 4.6 – Приклади програмування галтелі

X, то завжди дуга буде опуклою. Якщо по осі Z, – то угнута. Але при цьому треба пам'ятати правило про те, що координата кінцевої точки дуги повинна бути такою самою, як і в попередньому кадрі (якщо рух відбувався по одній осі). У цих випадках потрібно звертати увагу на можливість запису в одному кадрі обробки циліндра або торця і галтелі.

#### 4.7.6 Програмування обробки дуг окружністю менше $90^{\circ}$

Дуги, розмір яких менше  $90^{\circ}$ , програмуються кадром програми, у якому задаються координати кінцевої точки дуги або відстань від початкової точки дуги до кінцевої (при програмуванні в збільшеннях), і радіусом дуги за адресою R зі знаком. Знак перед розміром радіуса визначається напрямком обходу дуги. Якщо дуга обходить за годинниковою стрілкою, то це буде плюс, але він за замовчуванням опускається. Якщо дуга обходить проти годинникової стрілки, то ставиться знак мінус. Напрямок обходу щодо осей координат не має значення. Щоб

уникнути обмежень, що накладають правила програмування галтелей, останні також можна програмувати як дуги окружності.

Приклади програмування дуг окружності показані на рисунку 4.7.

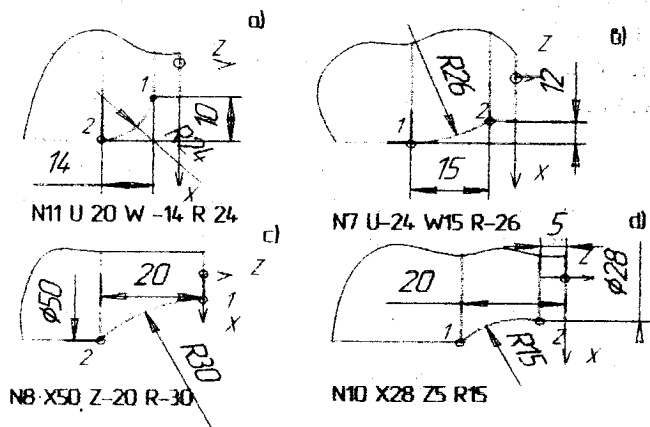


Рисунок 4.7 – Програмування обробки дуг окружністю менше  $90^\circ$

#### 4.8 Програмування технологічних циклів

У пам'яті машини при її виготовленні „зашиито” 11 технологічних циклів, що задаються за адресою L. Програмуючи стандартні цикли, необхідно знати кілька правил.

*Правило 1* Перед тим як задати той або інший технологічний цикл, треба інструмент підвести у вихідну точку циклу.

*Правило 2* Виконавши заданий цикл, машина поверне інструмент у вихідну точку циклу. Винятки складають цикли L8, L9, L10, після виконання яких інструмент залишиться в інцевій точці опису контуру креслення.

Деякі правила, що діють тільки у конкретному циклі, ми будемо розглядати під час вивчення програмування циклів.



### 4.8.1 Технологічний цикл нарізання різі різцем

Технологічний цикл нарізання різі різцем програмується за адресою L1. Структура циклу: L1 F.....W.....X...A.....P.....C. Тут F – крок різі; W – довжина ходу різця при нарізанні різі; X – внутрішній діаметр різі;

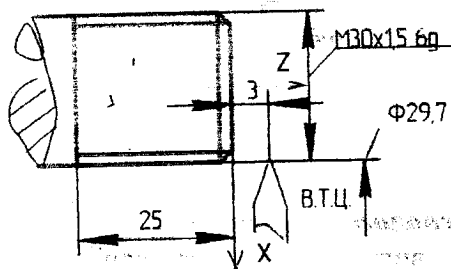
A – нахил різі (різниця радіусів основ конусів у випадку конічної різі), якщо різь циліндрична  $A=0$ ;

P – глибина різання в кожному проході різця; C – збіг різі (якщо різь закінчується канавкою, то C0, якщо канавки в кінці різі немає, то C1).

Вихідна точка циклу по осі X – фактичний діаметр заготовки під нарізання різі. По осі Z – зазор 2 – 3 кроки різі.

За цим циклом можна нарізати усі види різей: як зовнішні, так і внутрішні.

Приклад задання циклу показаний на рисунку 4.8.



N22 X29,7 Z3 E M8

Вихід у вихідну точку циклу

N23 L1 F1,5 W -28 X28,9 A0

P0,12 C1

Цикл нарізування різі з кроком 1,5 мм

Рисунок 4.8 – Програмування циклу нарізання різі різцем

### 4.8.2 Програмування циклу прорізання широкої канавки

Даний технологічний цикл програмується за адресою L2 і має структуру: L2 D...X...A.....P.... Тут D – витримка часу, с; X – внутрішній діаметр канавки; A – ширина канавки; P – ширина канавкового різця. Витримка часу задається для того, щоб на дні канавки не було видно слідів проходів різця.

Під час витримки ліквідуються всі пружні деформації, що виникли під дією сил різання, і різець займає в кожному проході стабільне положення.

Вихідна точка по осі Z знаходиться на лівому краї канавки, а по осі X віддалена на 1 – 2 мм від поверхні заготовки. Приклад програмування циклу показаний на рисунку 4.9.

### 4.8.3 Програмування циклу „петля зовнішня”

Петля зовнішня програмується за адресою L3 і має формат L2 W.....,

де W – довжина робочого ходу різця.

За цим циклом різець зробить рух по замкнутому прямокутнику паралельно твірній циліндра.

Вихідна точка циклу знаходиться на діаметрі, що буде отриманий у результаті виконання циклу з зазором по осі Z 1 – 2 мм.

Даний цикл виконується машиною з автоматичним дробленням стружки. Використовувати цей цикл рекомендується в тих випадках, коли припуск по зовнішній поверхні вимагає кілька проходів. Приклад програмування циклу показаний на рисунку 4.10.

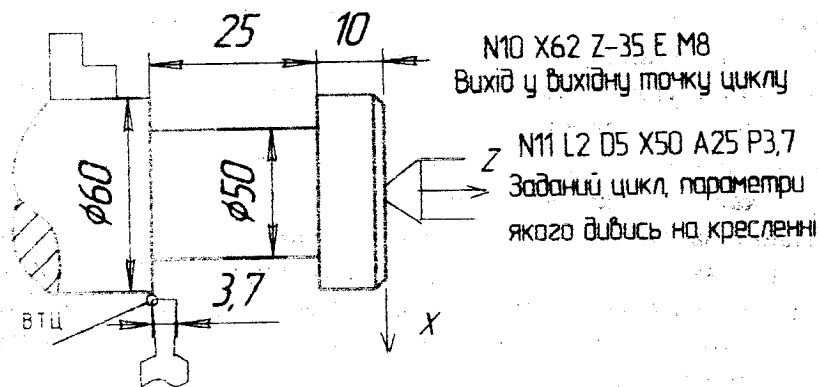


Рисунок 4.9 – Програмування циклу прорізання канавок

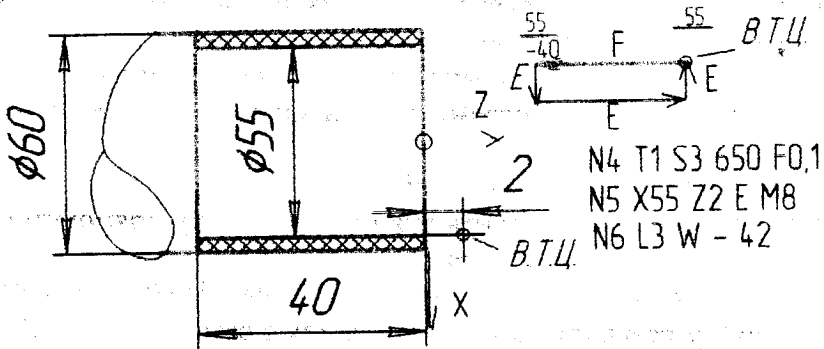


Рисунок 4.10 – Програмування циклу «петля зовнішня»

У наведеному на рисунку 4.10 прикладі у кадрі № 4 установлений прохідний різець і задані режими різання. У кадрі № 5 запрограмований вихід у вихідну точку циклу. У кадрі № 6 запрограмований цикл «петля зовнішня». За цим кадром програми різець зробить рух по замкнутому прямокутнику і повернеться у вихідну точку циклу. Тепер, якщо треба зняти великий припуск, перемістимо різець у нову в.т.ц. по діаметру і задамо цикл повторення ділянки програми L 11. У цьому випадку переміщати різець по діаметру потрібно у збільшеннях.

Покажемо це на прикладі. Припустимо, на деталі, показаній на рисунку 4.10, треба обточити ступінь довжиною 40 мм до діаметра 30 мм. Тоді припуск на обробку  $60-30=30$  мм. На радіус це складе 15 мм. Візьмемо глибину різання 3 мм. Тоді для знімання припуску нам необхідно 5 проходів. Програма буде виглядати так:

N1 G95

N2 T1 S3 650 F0,25

N3 X 54 Z 2 E M8 команда виходу різця у вихідну точку циклу  
( $60-6=54$ )

N5 L3 W -42

заданий цикл «петля зовнішня»

N6 U – 6 E M18 переміщення різця на швидкому ході в нову вихідну точку циклу по X і зазначена команда M18 – кінець повторюваної ділянки програми

N7 L11 B5 H4 заданий цикл повторення ділянки програми, що починається з кадру № 5 і повторюється 4 рази (у сумі буде 5 проходів)

N8 M9

N9 M02

#### 4.8.4 Програмування циклу «петля внутрішня»

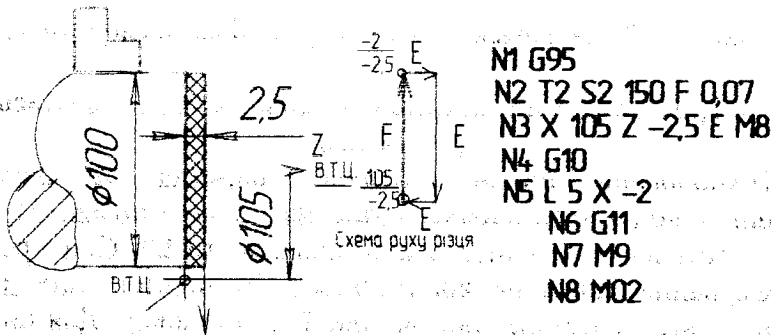
Даний цикл абсолютно подібний до попереднього, програмується за адресою L4 і має таку саму структуру. Застосовується цей цикл для розточення отвору.

#### 4.8.5 Програмування циклу «петля торцева»

Цикл «петля торцева» програмується за адресою L5 і має структуру L5 X..... За адресою X задають діаметр, до якого підрізують торець.

Приклад програмування циклу показаний на рисунку 4.11.

Цей цикл також можна використовувати разом з циклом «повторення ділянки програми» для знімання великих припусків з торця заготовки. У цьому випадку потрібно переміщення в нову вихідну точку циклу по осі Z задати у збільшеннях, як це розглянуто у прикладі з «петлею зовнішньою».



```

N1 G95
N2 T2 S2 150 F 0,07
N3 X 105 Z -2,5 E M8
N4 G10
N5 L 5 X -2
N6 G11
N7 M9
N8 M02

```

Рисунок 4.11 – Програмування циклу «петля торцева»

**4.8.6 Програмування циклу «глибоке свердління»**

Технологічний цикл «глибоке свердління» програмується за адресою L6 і має структуру L6 P.....W..... Тут P – глибина одного проходу свердла; W – загальна довжина ходу свердла. Вихідна точка циклу – це вісь обертання деталі по осі X і зазор безпеки 1-2мм по осі Z.

Приклад програмування циклу глибокого свердління показаний на рисунку 4.12.

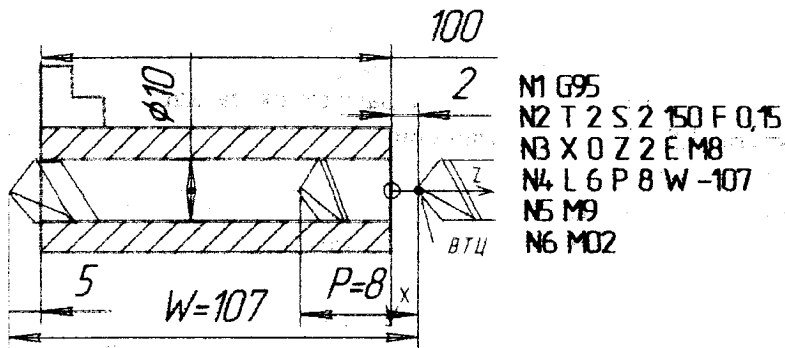


Рисунок 4.12 – Програмування циклу глибокого свердління

**4.8.7 – Програмування циклів «чорнова обробка за описом креслення»**

Технологічні цикли «чорнова обробка за описом креслення» кодуються двояко. Якщо заготовка – прокат, то L8. Якщо заготовка – штамповка або виливок, то L9. Сам процес програмування нічим не відрізняється. Відмінність цих двох циклів полягає у призначенні вихідної точки циклу. Для циклу L8 – це по осі X – зовнішній діаметр заготовки або діаметр отвору, а по осі Z – зазор до торця заготовки 1-3мм. Структура циклів така: L8 A.....P.....,

де А – припуск на чистову обробку; Р – глибина різання.

Для циклу L9 вихідна точка циклу розраховується за наступним алгоритмом.

Для циклу L09 перед програмуванням вихідної точки вимірюють максимальний припуск під обробку на всій заготовці як по діаметру, так і по довжині. А далі розрахунок проводять за правилами:

1 Якщо вчетверо більший припуск по довжині більше припуску по діаметру, то додають вчетверо більший припуск по довжині до розміру по діаметру і це буде точка по X, а припуск по довжині додають до координати торця, і це буде координатою вихідної точки циклу по осі Z.

2 Якщо вчетверо більший припуск по довжині менше, ніж припуск по діаметру, то для розрахунку вихідної точки по X береться припуск по діаметру, а зсув по торцю визначається діленням припуску по діаметру на 4 і це буде точка по Z.

Цикли L8 і L9 можна використовувати для зовнішньої обробки тільки для діаметрів ступеней деталі, що поступово збільшуються. І для отворів – якщо їх діаметри поступово зменшуються. Канавки і різні проточування, що знаходяться в тіні, у даних циклах виконати неможливо.

Кількість кадрів опису креслення деталі в цих циклах обмежено числом 15. При цьому кадри, де описуються фаски або галтелі, вважаються як два.

Розглянемо програмування циклу L8 на прикладі деталі, показаної на рисунку 4.13.

Опис креслення завжди починається з лівого торця деталі і йде зліва направо.

Якщо деталь починається з фаски або галтелі, то перед описом креслення до лівого торця деталі побудовують повітряний циліндр, діаметр якого дорівнює меншій основі фаски. І опис починають з діаметра цього циліндра.

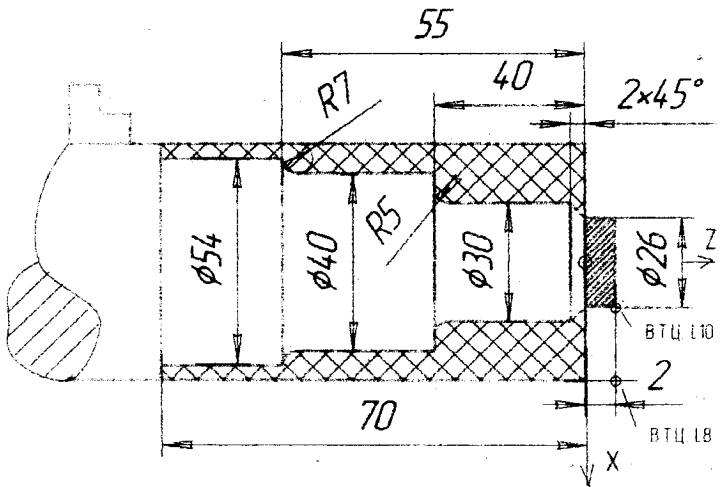


Рисунок 4.13 Креслення деталі для застосування циклу L8

Програма обробки:

N1 G95

N2 T1 S3 350 F 0,25

N3 X 60 Z 2 E M8 вихід на швидкому ході у вихідну точку циклу і ввімкнення ЗОР

N4 L 8 A 1 P 3 заданий цикл із залишенням припуску на чистову обробку 1мм по контуру, глибина різання 3мм

N5 X 26 початок опису креслення деталі

N6 Z 0

N7 Z -2 C 2

N8 Z -40 Q 5

N9 Z -55 Q 7

N10 Z -70 M17 кінець опису креслення деталі

N11 M9

N12 T2 S3 650 F0,07 установили чистовий різець і задали чистові режими різання

N13 X 26 Z 2 E M8 вихід у вихідну точку циклу чистової обробки за описом креслення. Ця точка лежить на початку повітряного циліндра, що ми прибудували до деталі

N14 L 10 B 5 задали цикл чистової обробки за описом креслення. Опис починається у кадрі №5

N15 M9

N16 M02

**Попередження:** виконавши цикл L8 або L9, різець не повернеться у вихідну точку циклу, а залишиться в кінцевій точці опису креслення деталі. Це особливо важливо пам'ятати, якщо програмується обробка отвору.

Нами опущений цикл L7 – нарізання різі мітчиком. Цей цикл у реальних умовах майже не застосовується через імовірність поломки мітчиків в отворі. Структура цього циклу L7 F.....W....., де F – крок різі; W – довжина ходу мітчика з урахуванням довжини різі і зазору по осі Z (2 – 3 кроки різі). Якщо нарізається права різь, то шпindel обертається за годинниковою стрілкою, поки мітчик не пройде шлях W. Після чого відбувається реверс шпинделя і починається вигвинчування мітчика. При цьому супорт рухається назад з подачею F.

### Коментарі до стандартних циклів

Перед програмуванням циклу L02 необхідно запрограмувати вихідну точку циклу. Координата Z вихідної точки повинна збігатися з координатою лівої кромки канавки. Цикл містить переміщення на робочій подачі до координати X, витримку часу (якщо вона не дорівнює 0), відхід на прискореному ході у вихідну точку по осі X, зсув по координаті (у позитивному напрямку на величину P і т. д.) до досягнення ширини канавки A.

Якщо потрібно задати різання з перекриттям, то параметр P задається менше ширини різця. У цьому випадку ширину канавки A треба зменшити на суму перекриттів.

Якщо канавка дорівнює ширині різця, то параметр  $P=A$ .



Цикл закінчується відскакуванням по осі X у вихідну точку; а по осі Z інструмент залишиться в точці останнього проходу.

Цикли L03 і L04 містять переміщення на робочій подачі на величину W з урахуванням знака; відскакування на 1мм від обробленої поверхні, повернення на швидкому ході у вихідну точку циклу.

Цикл L05 містить переміщення на робочій подачі по осі X; відскакування на 1мм по осі Z у позитивний бік і повернення на швидкому ході у вихідну точку.

Цикл L06 містить переміщення на робочій подачі у негативному напрямку на величину P, повернення на швидкому ході у вихідну точку, переміщення на швидкому ході в точку, що віддалена від дна просвердленого отвору на 3мм, переміщення на робочій подачі на величину (P+3)мм і т.д. до досягнення повної глибини заданою величиною W.

Цикл L07 містить переміщення на подачі, що дорівнює кроку різі F, на величину W з урахуванням знака, реверс шпинделя і повернення у вихідну точку на подачі F.

Після виконання циклу шпиндель продовжує обертатися в зворотний бік. Тому перед подальшою роботою треба задати знову обертання шпинделя за адресою S.

Цикл L09 застосовується у випадках, коли форма заготовки нагадує форму деталі. У цьому випадку обробка проводиться паралельно контуру деталі.

Ознакою кінця повторюваної ділянки програми за циклом L11 є команда M18.

Припуск під чистовий цикл L10 по осі Z визначається автоматично шляхом ділення заданого за параметром A на 4.

Процес формування різі за циклом L01 характеризується тим, що в кожному проході різець автоматично зміщується по осі Z, так що різке кожний раз тільки одна різальна кромка. І тільки останній прохід відбувається по середині канавки різі. Величина зсуву розрахована на різь з кутом  $60^{\circ}$ .

**Розділ 5****Методика програмування обробки деталей на вертикально-фрезерних верстатах із системою керування 2С42 – 65****5.1 Деякі особливості програмування для фрезерування кінцевою фрезою**

5.1.1 При програмуванні обробки контуру заготовки кінцевою фрезою у програмі описується не траєкторія руху формотворної кромки, а центра фрези. Ця траєкторія називається *еквідистантою, або еквідистантним контуром, на відміну від заданого контуру деталі.*

Еквідистантний контур віддалений від заданого на відстань радіуса фрези.

Оскільки еквідистантний контур у програмі розраховується на якийсь визначений радіус фрези, а фактичний радіус може відрізнятись від розрахункового, то виникає необхідність введення корекції радіуса фрези.

Крім того, якщо верстат оснащений багатоінструментальною револьверною головкою, то виникає необхідність у корекції довжини інструмента, тому що прив'язку до нуля деталі виконують тільки для одного інструмента.

5.1.2 У пам'яті машини «защитий» алгоритм автоматичного розрахунку еквідистантного контуру по заданому контуру деталі.

5.1.3 На відміну від токарних, вертикально – фрезерні верстати працюють не в площині, а в просторі. Тобто програмується переміщення центра фрези по трьох координатах. Однак просторове переміщення можна задавати тільки у випадку переміщення по прямій (при лінійній інтерполяції). При переміщенні по дузі (кругова інтерполяція) можна програмувати переміщення тільки в площині. Тому при програмуванні фрезерування контуру треба обов'язково зазначати площину, у якій лежать дуги окружності.

**5.2 Адреси та їх призначення**

: - номер програми і підпрограми.

- N- номер кадру.  
 G- службова функція.  
 X,Y,Z - абсолютні координати точки або збільшення по відповідній осі.  
 I,J,K- відстань від початку дуги до її центра.  
 F- робоча подача, мм/хв.  
 S- частота обертання шпинделя, об/хв.  
 T- номер інструмента.  
 M - допоміжна функція.  
 Q,R - додаткові технологічні функції.  
 H - номер коректора на довжину інструмента.  
 D - номер коректора на діаметр інструмента.  
 E - пауза.  
 P- число повторень підпрограми і виклик підпрограми.

### 5.3 Службові функції

- G 00 - прискорене переміщення по координатах.  
 G 01- лінійна інтерполяція.  
 G 02,G 03 - кругова інтерполяція за і проти годинникової стрілки.  
 G 04 - пауза.  
 G 09 - гальмування наприкінці кадру при обході гострих кутів контуру.  
 G 10 - лінійно-колова інтерполяція.  
 G17-XY } задання площини обробки дуг окружності.  
 G 18-XZ }  
 G 19-YZ }  
 G24 - вихід у нуль верстата по трьох координатах на прискореному ході.  
 G40 - скасування корекції на радіус інструмента.  
 G41- корекція на радіус інструмента, фреза у вихідному положенні ліворуч від оброблюваного контуру.  
 G42- те саме, але фреза праворуч від контуру.  
 G43 - корекція на довжину інструмента позитивна.  
 G44 - корекція на довжину інструмента негативна.

G49 - скасування корекції на довжину інструмента.

**Функції, використовувані при безеквідистантному програмуванні у збільшеннях:**

G 45- одинарне зміщення інструмента в «+».

G 46 – одинарне зміщення інструмента в «-».

G 47 - подвоєне зміщення інструмента в «+».

G 48 – подвоєне зміщення інструмента в «-».

### Стандартні цикли

G80-G86

### Інші функції

G92 - задання нуля координат деталі.

G 90 - абсолютна система координат (усі розміри в програмі задаються щодо нуля координат деталі)

G 91- робота зі збільшень (у програмі всі переміщення задаються в збільшеннях).

### 5.4 Допоміжні функції

M 00- безумовний зупинник програми.

M 03-праве обертання шпинделя (за годинниковою стрілкою).

M 04-ліве обертання шпинделя.

M 05- стоп обертання шпинделя.

M02- кінець програми.

M 30 - кінець програми з поверненням у її початок.

M 99- кінець підпрограми.

### 5.5 Програмування головного руху

Головний рух задається частотою обертання шпинделя за адресою S розрахунковим числом обертів. Напрямок обертання шпинделя задається допоміжною функцією: M03 – праве обертання і M04 – ліве обертання. Задана частота обертання шпинделя системою керування запам'ятовується і діє до надходження нової величини.

### 5.6 Програмування подачі

Рух подачі може бути робочим і прискореним. Робоча подача програмується за адресою F розрахунковим числом з розмірністю, мм/хв.

На пульті оператора системи 2С42 є ручний коректор робочої подачі, за допомогою якого можна змінити задану в програмі подачу на 20, 40, 50, 60, 70, 80, 90, 100, 110, 120 відсотків.

Прискорене переміщення задається службовою функцією G00. Робоча подача системою керування запам'ятовується і діє до надходження нової величини. Прискорене переміщення діє тільки в межах того кадру, де воно задано.

### **5.7 Лінійна інтерполяція**

Переміщення інструмента по прямій або кривій лінії називається інтерполяцією. Переміщення інструмента по прямій – це лінійна інтерполяція.

Враховуючи той факт, що система керування сама розраховує еквідистантний контур по введеному контуру деталі, ми надалі будемо в програмі описувати тільки рух інструмента, радіус якого дорівнює нулю.

Приклад: необхідно інструмент по прямій перемістити з точки А в точку В. Схема переміщення, координати точок і програми переміщення показані на рисунку 5.1

Лінійна інтерполяція задається функцією G01. Потім по осях X, Y, Z задаються координати кінцевої точки відрізка, якщо програмування виконується в абсолютній системі координат (G90). При програмуванні в збільшеннях (G91) вони визначаються за формулами:

$$\Delta X = X_k - X_n,$$

$$\Delta Y = Y_k - Y_n,$$

$$\Delta Z = Z_k - Z_n,$$

де  $X_k, Y_k, Z_k$  – координати кінцевої точки відрізка;

$X_n, Y_n, Z_n$  – координати початкової точки відрізка.

### **5.8 Кругова інтерполяція**

Кругова інтерполяція задається функцією G02, якщо дуга обробляється за годинниковою стрілкою, і G03 – якщо проти годинникової стрілки. Потім за адресами X, Y, Z задаються координати кінцевої точки дуги при програмуванні в абсолютних координатах

Розділ 5 Методика програмування обробки деталей на вертикально - фрезерних верстатах із системою керування 2С42 - 65

(G90). При програмуванні в збільшеннях (G91) за цими адресами

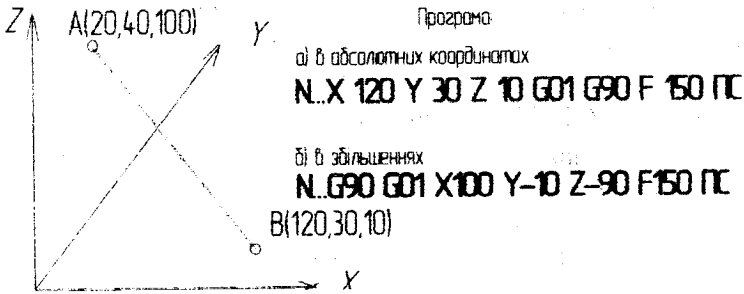


Рисунок 5.1 – Лінійна інтерполяція

задаються відстані від початкової до кінцевої точки дуги. Далі в кадрі записуються відстані від початкової точки дуги до її центра за адресами I,J,K, тому що адреси X,Y,Z уже задіяні. Приклад кругової інтерполяції показаний на рисунку 5.2.

Як бачимо, дуга обходить проти годинникової стрілки, тому G03. Дуга лежить у площині XY, тому G17. Відстань від початкової точки дуги до кінцевої по осі X буде  $20-40=-20$ , по осі Y буде  $40-15=25$ . Відстань від початкової точки дуги до її центра по осі X буде  $I = 15-40=-25$ , по осі Y  $J = 15-15=0$ . Подачі в цих кадрах немає, тому що вона була задана раніше в попередніх кадрах.

Керуюча машина 2С42 дозволяє програмувати обхід дуги  $360^\circ$ .

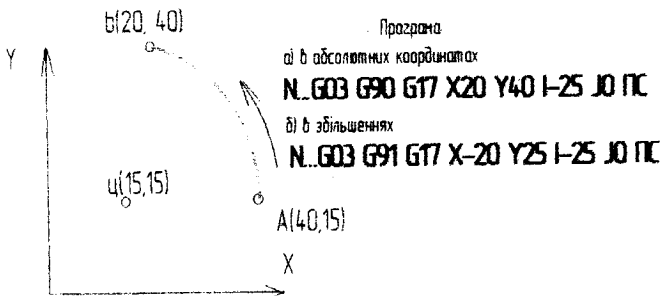


Рисунок 5.2 – Кругова інтерполяція проти годинникової стрілки

У цьому випадку кінцева точка дуги збігається з початковою, і до моменту програмування обходу дуги її координати вже є в пам'яті машини, у кадрі дуги задаються тільки відстані від початкової точки дуги до її центра. Приклад програмування показаний на рисунку 5.3.

Функції G17 і G90 задаються один раз і запам'ятовуються машиною до надходження змінної функції. Аналогічно – функції вигляду інтерполяції (G01, G02, G03). Тому в наведених прикладах вони записані тільки тому, що ми наводимо тільки фрагмент програми.

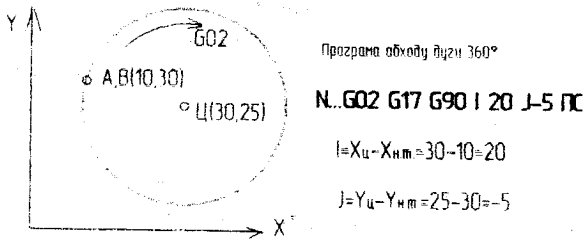


Рисунок 5.3 – Кругова інтерполяція при обході дуги 360°

### 5.9 Лінійно – кругова інтерполяція

Лінійно – кругова інтерполяція (рух по спіралі) програмується функцією G10.

Приклад лінійно-кругової інтерполяції показаний на рисунку 5.4.

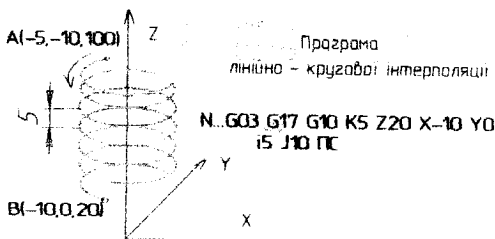


Рисунок 5.4 – Лінійно – кругова інтерполяція

До складу кадру лінійно – кругової інтерполяції входить: напрямок обходу окружності, що формує спіраль (G03); площа, у якій лежить проекція окружності, що формує спіраль (якщо вона не була раніше задана) (G17); код лінійно – кругової інтерполяції (G10); крок спіралі за адресою K (K10); якщо програмується в абсолютних координатах, то координати кінцевої точки дуги по всіх трьох осях (X-10, Y0, Z20); і, нарешті, відстань від початкової точки дуги до її центра в проекції на площину XOY (I 5, J 10).

Якщо програма пишеться в збільшеннях, то зміни будуть тільки в даних по осях X, Y, Z, за якими запишуться різниці координат кінцевої і початкової точок формуючої дуги.

Варто зазначити, що крок спіралі задається за адресою K і він завжди позитивний. Таким чином, спіраль може бути запрограмована тільки в напрямку осі Z.

### **5.10 Приклади безеквідистантного програмування**

Як ми уже відзначали вище, система 2С42 дозволяє автоматично розраховувати еквідистантний контур за заданим у програмі контуром деталі і радіусом фрези. При цьому методика програмування буде відрізнятися тим, у якій системі ми задаємо розміри координат точок контуру.

Тому розглянемо програмування в збільшеннях і в абсолютних координатах.

#### **5.10.1 Програмування в збільшеннях**

При програмуванні в збільшеннях використовуються функції G45 – одне позитивне зміщення, G46 – одне негативне зміщення, G47 – подвійне позитивне зміщення, G48 – подвійне негативне зміщення.

Величина зміщення дорівнює радіусу фрези, що вводиться в пам'ять машини шляхом набору його величини в будь - який коректор D.

Команди G45-G48 діють тільки в межах кадру, де вони задані.



Приклад деталі для програмування в збільшеннях показаний на рисунку 5.5.

Керуюча програма:

%ПС – початок програми

M3 S350 F120 T1 H01 G43 ПС – програма №1, праве обертання шпинделя (M3), 350об/хв – частота обертання шпинделя, робоча подача 120мм/хв (F120), перший інструмент (T1) і його коректор по довжині (H01), корекція на довжину інструмента позитивна (G43).

N1 G91 G17 G01 G00 G46 D05 X5000 Y2000 ПС – задане програмування в збільшеннях (G91), площа XY, на якій лежать дуги на контурі деталі (G17), лінійна інтерполяція (G01), тому що рух буде відбуватися по прямій, прискорене переміщення (G00), при підході до точки 1 потрібно не доходити до неї на розмір радіуса фрези по обох осях, тому задане одне негативне зміщення (G46), а розмір цього зміщення (10мм) був заданий при налагодженні верстата в коректорі D05. А далі по осях X і Y задана відстань від точки 0 до точки 1.

N2 X4000 G47 D05 ПС – задана відстань від точки 1 до точки 2. Але ми в першому кадрі не дійшли до точки 1 на величину зміщення, і для обробки відрізка 2 – 3 треба пройти далі точки 2 на величину зміщення, тому в кадрі задане подвійне позитивне зміщення (G47). Величина цього зміщення записана в коректорі D05.

N3 Y3000 ПС – задана відстань від точки 2 до точки 3. Задавати зміщення тут не потрібно, тому що ми знаходилися від точки 2 нижче на величину зміщення і повинні не дійти до точки 3 на це зміщення.

N4 X 6000 G48 D05 ПС – задана відстань між точками 3 і 4. Але ми не повинні перемістити фрезу на цю відстань, тому що від точки 3 ми уже віддалені вправо на величину зміщення, і до точки 4 не повинні дійти на величину зміщення. Тому в кадрі записане подвійне негативне зміщення (G48).

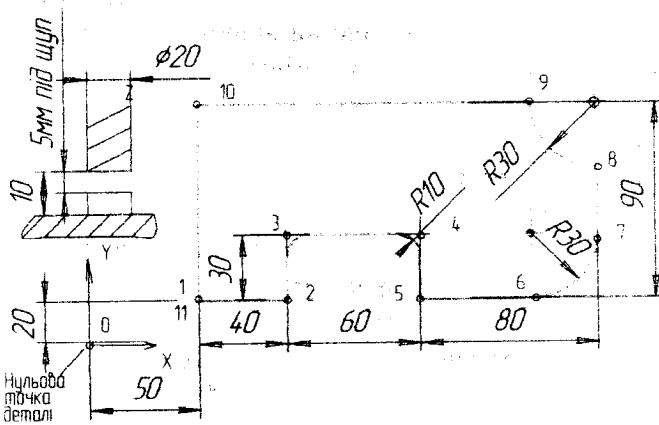


Рисунок 5.5

N5 Y-3000 PC – тут аналогічно до кадру №3, тільки відстань між точками 4 і 5 у зворотний бік осі Y.

N6 X5000 G45 D05 PC – записана відстань між точками 5 – 6, але ми були віддалені від точки 5 уліво на величину зміщення. Отже, це зміщення і потрібно додати (G45).

N7 G03 X3000 Y3000 I0 J3000 G45 D05 PC – заданий обхід по дузі радіусом 30мм. Але насправді фреза буде обходити дугу по радіусу 30 плюс одне позитивне зміщення. Тому в кадрі задана команда G45. Ця команда діє на всі геометричні адреси, записані в кадрі.

N8 G01 Y3000 G45 D05 PC – задана відстань від точки 7 до точки 8. Це вже лінійна інтерполяція, тому задаємо G01. Для того щоб можна було обходити дугу 8 – 9, треба пройти точку 9 на величину одного зміщення. Тому в кадрі знову команда G45.

N9 G02 X-3000 Y3000 I0 J3000 G46 D05 PC – обходимо дугу 8 – 9 за годинниковою стрілкою, тому G02. Задаємо одне негативне зміщення (G46), тому що центр фрези буде рухатися по дузі меншого радіуса, ніж заданий на контурі, на величину одного зміщення.

N10 G01 X-15000 G47 D05 PC – задана відстань між точками 9 – 10. Але ми знаходилися правіше точки 9 на величину

зміщення (при обході дуги), і щоб рухатися по прямій 10 – 11, нам треба пройти точку 10 лівіше на величину зміщення. Тому в кадрі задане подвійне позитивне зміщення (G57).

N11 Y-9000 G47 D05 PC – задані відстань 10 – 11 і подвійне позитивне зміщення. Тому що ми знаходилися вище точки 10 на величину радіуса фрези і щоб обробити поверхню 10-11, потрібно пройти точку 11 на деяку відстань.

N12 G00 G90 M9 X0 Y0 G49 PC – для повернення в нуль деталі ми перейшли на абсолютні координати (G90) і на прискореному ході (G00) задаємо координати точки 0, при цьому скасовуємо корекцію на довжину інструмента (G49).

N13 M02 M30 PC – кінець програми (M02) і повернення до початку програми (M30).

У програмі всі розміри задані з дискретою 0,01мм.

### **5.10.2 Програмування в абсолютних координатах**

При безеквідистантному програмуванні в абсолютних координатах діють функції G41– корекція на радіус інструмента, фреза ліворуч від оброблюваного контуру і G42 - праворуч від контуру. Скасовує ці функції команда G40. Функції G41, G42 задаються один раз на початку програми при програмуванні виходу фрези з нульової точки в напрямку до оброблюваного контуру. І при поверненні в нульову точку після обходу контуру вони скасовуються командою G40.

Приклад деталі для даного методу безеквідистантного програмування наведений на рисунку 5.6. На рисунку суцільною лінією позначений контур деталі, штриховою – контур заготовки. Фреза в нульовій точці розміщена на 10 мм нижче від основи деталі. Товщина деталі 10мм. Матеріал – сталь 45. Мінімальний припуск по контуру – 5мм. Закріплення деталі виконується за допомогою двох технологічних отворів.

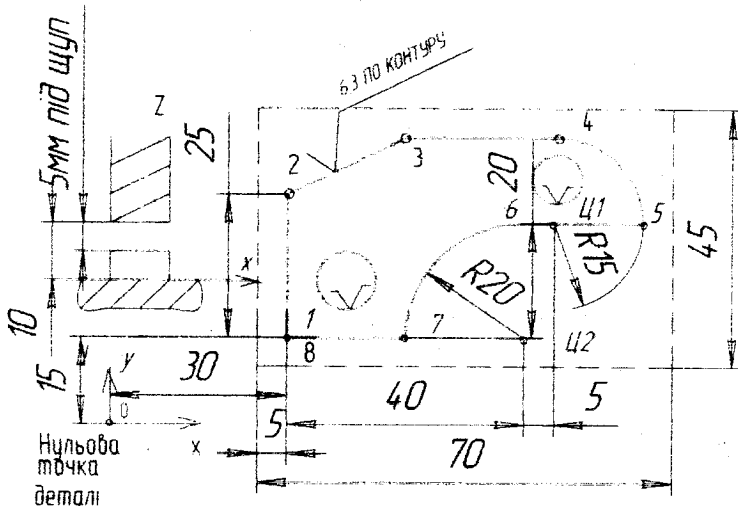


Рисунок 5.6

Програма обробки в абсолютних координатах:

% ПС – початок програми

: 2 G92 X0 Y0 Z0 T1 ПС – прив'язка першого інструмента до нуля деталі. Цю операцію можна робити, як при настроюванні верстата, шляхом задання функції G92 із клавіатури пульта оператора після встановлення фрези в нульове положення. Так було зроблено у вищенаведеній програмі. Або задавати це саме першим кадром програми, за який можна взяти і номер програми, як це зроблено у нас. Тут цифра 2 - це номер програми, а не номер кадру.

N1 G00 G01 G90 X1000 ПС – цим кадром ми зрушили фрезу по осі X на 10мм, щоб потім можна було її опустити на рівень деталі по осі Z. G90 означає, що програмування буде вестися в абсолютних координатах.

N2 M3 S250 F75 G41 D03 X3000 Y1500 Z-1000 ПС – ввімкнули праве обертання шпинделя, установили 250 об/хв, робочу подачу 75мм/хв, зазначили, що фреза в нульовій точці знаходиться ліворуч від деталі (G41), указали, де записаний

радіус фрези (D03), і задали координати точки 1. Фреза одночасно переміщається по трьох координатах на робочій подачі, щоб уникнути удару об заготовку.

N3 Y4000 PC – координата точки 2. Тому що координата по осі X не змінюється, то ми її і не задаємо.

N4 X5500 Y3500 PC – координати точки 3.

N5 X7500 PC – координата точки 4.

N6 G02 X7000 Y3500 I0 J-1500 PC – описуємо дугу 4 – 5, тому кругова інтерполяція за годинниковою стрілкою - G02, задаємо по осях X і Y координати кінцевої точки дуги і за адресами I і J - відстані від початкової точки дуги до її центра Ц1 по тих самих осях.

N7 G01 X7500 PC – задаємо координати точки 6 і вказуємо, що рух буде йти по прямій.

N8 G03 X5000 Y1500 I0 J -2000 PC – обходимо дугу проти годинникової стрілки (G03) і записуємо координати кінцевої точки дуги по осях X і Y. Потім за адресами I і J задаємо відстань від початкової точки дуги до її центра Ц2.

N9 G01 X3000 PC – координата точки 8. Якщо координата Y не змінюється, то її і не записуємо. При цьому переходимо на лінійну інтерполяцію (G01).

N10 G40 G00 X0 Y0 Z0 PC – скасовуємо функцію G41 і даємо команду на швидкому ході повернутися в нульову точку деталі.

N11 M02 M30 PC.

### **5.11 Програмування з використанням стандартних циклів**

Практика обробки деяких поверхонь деталей на вертикально – фрезерних верстатах виробила оптимальні прийоми їх виконання. Ці прийоми виконання робіт запрограмовані і «зашиті» у пам'ять керуючої машини при її виготовленні. Називаються такі прийоми стандартними циклами або технологічними циклами. Виклик технологічних циклів здійснюється функцією G із зазначенням двозначним числом коду циклу. У машину «зашиті» такі стандартні цикли:  
G81 – цикл свердління отвору;

G82 – цикл розточування отвору без зупинення наприкінці роботи;

G83 – цикл глибокого свердління;

G84 – цикл нарізування різі мітчиком;

G85 – цикл розточування отвору із зупиненням в кінцевій точці розточення ( для одержання стабільної глибини отвору);

G86 – цикл свердління з відскакуванням (для руйнування стружки);

G80 – скасування циклу.

### 5.11.1 Цикл свердління отвору

Цикл свердління отвору задається функцією G81 і має таку структуру: G81 U.....Z.....I.....F.....E... (рис. 5.7),

де U – координата кінцевої точки швидкого підведення свердла до початку свердління (G90), або відстань від початку руху на швидкому ході до його кінцевої точки (G91);

Z – координата кінцевої точки свердління отвору (G90), або відстань від точки 1 до точки 2, що свердло пройде на робочій подачі (G91);

I – координата точки 3, у яку свердло виходить із отвору на швидкому ході (G90), або відстань 2 – 3 при G91;

F – величина робочої подачі, мм/хв;

E – величина витримки часу (наприкінці свердління глухого отвору для одержання точної глибини відбувається зупинення подачі при обертовому шпинделі).

Адреси I, F, E не є обов'язковими на відміну від U і Z. Якщо адреса I не задана, то свердло вийде з отвору в точку 1. Якщо нас улаштовує подача, задана до циклу, то усередині циклу її задавати повторно не потрібно. Аналогічно, якщо отвір наскрізний або його глибина не має високої точності, витримку часу задавати не треба.

Приклад застосування циклу свердління показаний на рисунку 5.8.

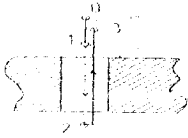


Рисунок 5.7 – Структура рухів циклу свердління

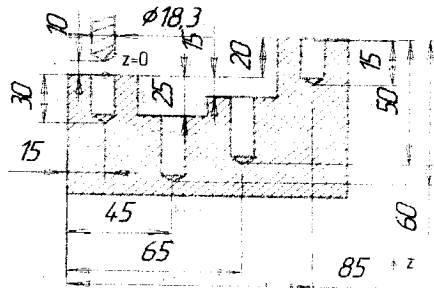


Рисунок 5.8

Для спрощення задачі будемо вважати, що всі отвори розміщуються уздовж осі X. Свердляться отвори під різі M20, їх глибина не має вимог щодо точності. Отже, у циклі свердління витримку часу задавати не будемо. Прив'язку по довжині інструмента здійснюємо по спіральному свердлу, тому для нього корекція по довжині у програмі не задається. Будемо вважати, що нуль координати Z деталі буде на поверхні першого отвору. Це ми зазначимо в першому кадрі програми.

Програма:

% ПС

: 1 ПС

N1 G54 G90 X0 Y0 Z1000 M3 S250 ПС – функцією G54 вказуємо, що це місце в просторі, у якому знаходиться свердло вважати нулем координат по осях X і Y, а нуль по осі Z розміщений на відстані 10мм від положення свердла, тобто на поверхні першого отвору. G90 – вказує, що програмування буде здійснюватися в абсолютних координатах. M3 – праве обертання шпинделя з частотою 250 об/хв.

N2 G81 U0,5 Z-3000 F100 ПС – свердло за алгоритмом циклу на швидкому ході пройде шлях 9,5мм і за 0,5мм зупиниться. Далі до координати - 30мм по осі Z воно буде рухатися на робочій

подачі 100 мм/хв. Після чого на швидкому ході вийде з отвору і зупиниться на відстані 0,5мм від поверхні.

N3 X4500 G00 PC – на швидкому ході свердло переміститься для свердління другого отвору.

N4 G81 U-2450 I=0 Z -4000 PC – свердлиться другий отвір. Оскільки його площина нижче положення свердла, то воно на швидкому ході пройде шлях до координати -24,5мм по осі Z і зупиниться за 0,5 мм від поверхні деталі. Далі свердлить отвір до точки Z= - 40мм. Оскільки площина наступного отвору знаходиться вище, то свердло виходить з отвору в точку I=0 (Z=0).

N5 X6500 G00 PC – свердло переміщаємо в точку третього отвору.

N6 G81 U -1450 Z -3500 I 3000 PC – свердлиться третій отвір до точки Z= -35мм. Перед цим спочатку свердло на швидкому ході підходить до площини отвору, на 0,5мм не доходячи до неї. Після свердління на швидкому ході свердло піднімається в точку Z= 30мм, тому що площина останнього отвору знаходиться вище просвердленого отвору на 20мм.

N7 X7000 G00 PC – координата четвертого отвору.

N8 G81 U2050 Z500 PC – свердлиться четвертий, останній, отвір. Для цього свердло підходить до поверхні деталі за 0,5 мм до неї (U=2050). Точка закінчення свердління знаходиться вище нульової точки по осі Z на 5мм.

N9 G80 PC – скасування циклу свердління.

N10 G00 X0 Y0 Z1000 M5 PC – повернення свердла в початкову точку і зупинення шпинделя.

N11 M02 M30 PC – кінець програми.

### 5.11.2 Цикли розточування

Як уже було сказано вище, у системі є два цикли розточування:

G82 – розточування без зупинення подачі наприкінці розточення і G85 – розточування із зупиненням подачі в кінцевій точці розточення отвору. Останній цикл



використовується для розточення глухих отворів з точною глибиною отвору.

Діаграма рухів розточувальної голівки показана на рисунку 5.9.

Структура циклу G82:

G82 U.....Z.....I...F...E...H...ПС.

Тут всі адреси, крім H, мають той самий зміст, що й у циклі свердління.

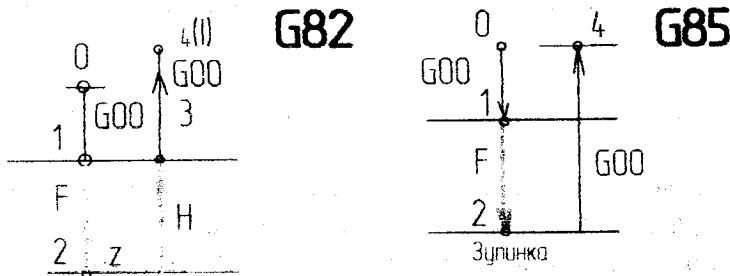


Рисунок 5.9 Діаграма рухів циклів розточування

Адреса H визначає подачу на ділянці 2 – 3 при виході розточувальної голівки. Якщо він не заданий, то вихід буде відбуватися на подачі F. Якщо задана точка 4 (адреса I), то рух до точки 3 буде з подачами H або F, а далі на швидкому ході.

Структура циклу G85:

G85 U...Z.....I.....E.....F.....M...ПС.

Адреса M дає можливість або реверсувати обертання шпинделя, або зупиняти його в точці 2. Якщо ця адреса не задана, то в точці 3 шпиндель обертається в той самий бік, що й у точці 0.

### 5.11.3 Цикл глибокого свердління

Структура циклу G83: U.....Z.....I.....F.....V...W.....ПС.

Як бачимо, тут додалося дві нових адреси, що означають: V – глибина одного проходу свердла; W – зазор між дном раніше просвердленого отвору і підходом свердла на швидкому ході

для виконання наступного проходу. Структура рухів свердла при виконанні циклу G83 показана на рисунку 5.10.

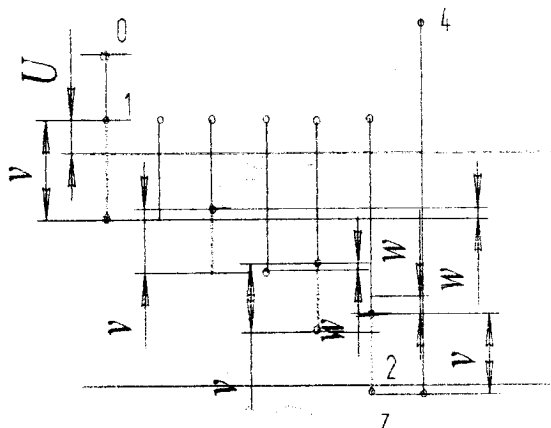


Рисунок 5.10 *Діаграма рухів циклу глибокого свердління*

Приклад: просвердлити отвір діаметром 16 мм на глибину 80 мм.

N.....U50 Z-8000 I1000 W 50 V1000 ПС.

У програмі задано: глибина одного свердління  $V=10$  мм; зазор безпеки при підведенні свердла до дна отвору  $W=0,5$  мм. За один прохід свердлиться отвір глибиною 10 мм, потім на швидкому ході свердло виводиться з отвору в точку 1 і після паузи, що дорівнює 1 с, на швидкому ході вводиться в отвір, не доходячи до його дна на 0,5 мм, і свердлить знову 10 мм і т.д. до точки 2 ( координата Z).

#### 5.11.4 Цикл переривчастого свердління

Структура циклу G86: U.....Z.....I.....F.....V.....W.....ПС.

Тут V – глибина одного свердління, так само, як у попередньому циклі; W – величина відсакування від дна отвору для руйнування стружки.

Діаграма рухів циклу показана на рисунку 5.11.



**Розділ 6****Параметричне програмування для машини 2С42-65****6.1 Алфавіт мови макропрограмування**

Алфавіт містить у собі всі символи мови програмування ISO.

**6.2 Представлення чисел**

Числа представляються в десятковій системі числення у форматі з фіксованою комою. Ціла частина числа відокремлюється від дробової за допомогою крапки.

**6.2.1 Константи**

Константа – це дійсне ціле число, представлене в десятковій системі вирахування в діапазоні від -8388607,99999 до +8388607,99999. Точність представлення чисел у системі  $\pm 0,00001$ , тобто мінімальне позитивне число, що може бути запрограмоване, +0,00001, а максимальне негативне число -0,00001.

Адреси і цифрова інформація в машині при параметричному програмуванні зберігаються у двох видах пам'яті: формальний і системний.

**6.3 Формальна пам'ять**

У системі 2С42-65 передбачено 195 комірок пам'яті з номерами від 1 до 195, що називається формальною пам'яттю. Ця пам'ять може у своїх осередках зберігати будь-яку інформацію. Комірки пам'яті мають адреси #. Ємність кожної комірки формальної пам'яті така, що може вмістити в себе десяткове число від -8388607,99999 до +8388607,99999.

**6.4 Системна пам'ять**

Комірки системної пам'яті, на відміну від формальної, мають строго фіксоване призначення, яке програміст не має права порушувати.

Призначення комірок системної пам'яті показано в таблиці 6.1. Адреса комірки системної пам'яті &.

При перезаписі вмісту комірки системної пам'яті у формальну пам'ять вони записуються як ціле число.

**6.5 Вираження мови макропрограмування**

До складу виражень мови входять:

- десяткові числа;
- номери комірок пам'яті;
- оператори;
- функції.

Числа можуть бути написані безпосередньо або шляхом задання номера комірки пам'яті, у якій вони знаходяться.

До комірок пам'яті при програмуванні можна звертатися двома способами:

- безпосередньо;
- побічно.

*Таблиця 6.1 - Приклади закріплення комірок системної пам'яті*

Номер комірки	Вміст комірки
&2	Стан шпинделя M3     &2=3 M4     &2=4 M5     &2=5
&4	Дискретність індикації геометричних даних: 1мм - &4=0 10мм - &4=1
&23	Номер діючого коректора на довжину інструмента (H)
&24	Те саме на радіус інструмента (D)
&26	Номер інструмента, який потрібно вставити в шпиндель
&27	Подача – що діє в кадрі
Поточні значення координат у момент дотикання:	
&29	по X
&30	по B
&31	по Z
Абсолютне положення у системі координат деталі на початку поточного кадру:	
&33	по X
&34	по B
&35	по Z

### 6.5.1 Безпосередній спосіб звертання до комірки пам'яті

При звертанні до комірки пам'яті безпосередньо розуміється таке звертання, коли дані, записані в ній, будуть використані безпосередньо.

Приклад 1: # 2 означає, що в програмі буде використовуватися та інформація, що знаходиться в комірці № 2 формальної пам'яті.

Приклад 2: &2 означає, що у програмі буде безпосередньо використовуватися та інформація, що знаходиться в комірці № 2 системної пам'яті (стан шпинделя).

### 6.5.2 Непрямий спосіб звертання до комірки пам'яті

Під непрямым звертанням розуміється таке звертання, коли в програмі буде використовуватися не число, записане в комірку, а номер комірки, що записаний в комірці, і в якому в кінцевому підсумку знаходиться потрібне число або інша інформація.

Приклад 1: # # 3 – буде означати, що в програмі використовується число, що знаходиться в комірці номер якої зберігається в комірці № 3 формальної пам'яті.

Приклад 2: & # 4 – цей запис означає, що в програмі при виконанні певної – те операції буде використана інформація, записана в комірці системної пам'яті, номер якої зберігається в комірці № 4 формальної пам'яті.

Припустимо, що в комірці формальної пам'яті записане число 1001. Таким чином, у деякій операції програми буде використана величина, що знаходиться в комірці системної пам'яті 1001 (коректор №1 на довжину інструмента або його діаметр).

Оскільки комірки системної пам'яті за своїм призначенням строго регламентовані, то в них не може бути записаний номер якої-небудь комірки. Тобто запис & # 4 є неприпустимим.

### 6.6 Оператор присвоєння

Оператор присвоєння програмується символом «=» і означає присвоєння комірці пам'яті, що розміщена ліворуч від

Розділ 6 Параметричне програмування для машини 2С42 - 65

знака «=», результату операції, що розміщена праворуч від знака.

Приклад 1: запис  $\#10 = \#26$  означає, що в комірку № 10 буде записана величина, що знаходиться в комірці № 26.

Приклад 2: запис  $\# \# 2 = \& 10$  означає, що в комірку формальної пам'яті, номер якої зберігається в комірці № 2, треба записати вміст комірки системної пам'яті № 10.

Приклад 3:  $\# 3 = 12$  – в комірку № 3 записати число 12.

Приклад 4:  $\# 12 = 0.5$  – в комірку № 12 записати число 0,5.

Приклад 5:  $\# 3 = 3001$  – в комірку № 3 записати число 3001.

Приклад 6:  $\& 6001 = G00$  – в комірку системної пам'яті записати код швидкого ходу.

### **6.7 Неявний оператор присвоєння**

Неявний оператор присвоєння не програмується, а виконується автоматично при розв'язанні якого – небудь арифметичного виразу за таким правилом: *результат обчислення завжди записується в крайню ліву комірку пам'яті з усіх комірок, наведених у виразі*

$\# 10 + \# 25 - \# 45.$

Цей запис читається так: вміст комірки №10 підсумовується з вмістом комірки № 25. Із суми віднімається вміст комірки № 45. Результат обчислення записується в комірку № 10.

Використовуючи неявний оператор присвоєння, потрібно пам'ятати, що вміст комірок системної пам'яті строго регламентований. Тому номер системної комірки не може бути записаний крайнім зліва.

### **6.8 Арифметичні вирази**

Арифметичні вирази програмуються для виконання яких – небудь арифметичних дій.

#### **6.8.1 Оператор додавання**

Оператор додавання програмується символом «+» (плюс). Реалізується таким чином, що результат операції завжди

Розділ 6 Параметричне програмування для машини 2С42 - 65

записується в ліву комірку пам'яті в записаному виразі. Запис #7=100.007 + 22.458 означає, що треба скласти два наведених числа і суму записати в комірку № 7 формальної пам'яті.

Запис # 1 + # 7 означає, що треба скласти два числа, записані в наведених комірках формальної пам'яті, і результат записати в комірку № 1.

Цей вираз можна записати і так: #1 = #1 + # 7.

Запис # 24 = 102.1 + #18 означає, що в комірку № 24 буде записана сума числа 102,1 з величиною, що знаходиться в комірці системної пам'яті, номер якого зберігається в комірці 18 формальної пам'яті.

*Примітка.* Знак + не можна використовувати для вказівки того, що число позитивне. Тобто запис #10= +12.8 машиною не буде зрозумілим і сприйматиметься як помилка. Варто писати #10= 12.8.

### 6.8.2 Оператор віднімання

Оператор віднімання програмується символом " - ", (мінус) і означає арифметичну операцію віднімання.

Запис # 3 = 100.5 - 75.7 означає, що з першого числа треба відняти друге і результат записати в комірку № 3.

# 15 - # 45 означає, що з числа, що знаходиться в комірці № 15, треба відняти число, що знаходиться в комірці № 45, і різницю записати в комірку № 15.

За допомогою знака мінус можна не тільки задавати операцію віднімання, але і зазначити, що певне число має негативне значення. Наприклад, запис # 7 = - 78.5 означає, що в комірку № 7 треба записати число - 78,5.

За допомогою цього знака можна інвертувати знак числа, записаного в комірку. Наприклад, запис # 14 = 45.85 - - # 1 означає, що в комірку № 14 треба записати різницю числа 45,85 і числа, що зберігається в комірці № 1, але з протилежним знаком.



**6.8.3 Оператор множення**

Оператор множення програмується символом " \* " (зірочка) і використовується для задання арифметичної дії множення.

Запис # 34 = # 4 \* 55.75 означає, що в комірку № 34 треба записати добуток числа 55,75 і числа, записаного в комірку № 4.

Запис # 7 \* 67.5 означає, що вміст комірки № 7 треба помножити на 67,5 і добуток записати в комірку № 7.

Запис & # 12 = & 1001 \* # 25 означає, що число, що зберігається в комірці № 1001 системної пам'яті, буде помножене на вміст комірки № 25 і результат буде записаний в комірку системної пам'яті, номер якої зберігається в комірці № 12 формальної пам'яті.

**6.8.4 Оператор ділення**

Оператор ділення програмується символом « / » (скісна риска) і означає арифметичну дію –ділення.

Запис # 75 = 55 / & # 29 означає, що в комірку № 75 треба записати частку від ділення числа 55 на число, що знаходиться в комірці системної пам'яті, номер якої зберігається в комірці № 29 формальної пам'яті.

Запис # 101 = 105/5 означає, що в комірку №101 треба записати частку від ділення числа 105 на число 5.

**ПРИМІТКИ:**

1 Всі арифметичні операції виконуються з точністю до 0,00001.

2 Якщо результат операції буде менше 0,00001, то він у пам'яті буде записаний як 0000000,00000.

3 Результат будь-якої арифметичної операції не повинен за модулем перевищувати значення 8388607, 99999.

4 Машина контролює, щоб не було ділення на нуль. Якщо це буде виявлене, то обчислення припиняються і на екрані монітора висвічується « 0? ».

**6.9 Обчислення функцій**

У мові макропрограмування є можливість обчислювати корінь квадратний, синус, косинус, арктангенс, знаходити модуль числа, а також проводити округлення чисел.

**6.9.1 Обчислення кореня квадратного**

Обчислення кореня квадратного програмується за допомогою оператора **SQR**.

Приклад 1: **SQR 5.2** означає знайти корінь квадратний з числа 5,2.

Приклад 2: **& # 5 = SQR # 1** означає, що потрібно знайти корінь квадратний з числа, що знаходиться в комірці № 1 формальної пам'яті, і результат записати в комірку системної пам'яті, номер якої зберігається в комірці № 5 формальної пам'яті.

При виконанні оператора **SQR** виробляється контроль знака числа під коренем і його величини, що не повинна перевищувати 8388607,99999.

Крім оператора **SQR**, у системі є оператор **DSQ**, за допомогою якого можна знайти корінь квадратний із суми квадратів чисел, що зберігаються в комірках № 88 і № 89 формальної пам'яті.

Приклад 3: **# 1 = DSQ** означає, що треба в комірку 1 формальної пам'яті записати величину кореня квадратного із суми квадратів чисел, що зберігаються в комірках № 88 і № 89.

*Примітки. Функція DSQ працює тільки з комірками № 88 і № 89.*

**6.9.2 Обчислення синуса і косинуса**

Для обчислення синуса або косинуса величина кута програмується десятковим числом у частках градуса.

Величина аргументу повинна бути в межах від  $-181^{\circ}$  до  $+181^{\circ}$ .

Команда на обчислення програмується операторами **SIN** і **COS**.

Розділ 6 Параметричне програмування для машини 2С42 - 65

Приклад 1: # 75 = SIN 30.75 означає, що в комірку № 75 формальної пам'яті треба записати синус кута  $30,75^{\circ}$ . Тут 30 – це цілі градуси, а 0,75 - це 45 хвилин ( $0,75 = 45/60$ ).

**6.9.3 Обчислення арктангенса**

Обчислення арктангенса числа програмується оператором **ATN**. При цьому аргумент не повинен виходити за межі можливості пам'яті (від - 8388607,99999 до +8388607,99999).

У результаті обчислення виходять кути в межах від  $-181^{\circ}$  до  $+181^{\circ}$ .

Приклад 1: # 4 = ATN 1 означає, що арктангенс 1 (тобто  $45^{\circ}$ ) буде записаний в комірку № 4 формальної пам'яті.

Крім операції обчислення арктангенса числа, є додаткова функція **DTN**, за допомогою якої визначається арктангенс частки від розподілу числа, записаного в комірку № 88, на число, записане в комірку № 89, тобто

$$\text{arctg} \frac{\#88}{\#89}$$

Приклад 2: #1 = DTN означає, що буде знайдений арктангенс відношення чисел, записаних в комірках № 88 і № 89, і результат запишеться в комірку № 1 формальної пам'яті.

*Примітка. З іншими комірками ця команда не працює!*

**6.9.4 Округлення чисел**

Округлення чисел у даній системі програмується трьома способами:

- округлення відповідно до загальноприйнятого правила **RND**;
- округлення з недостачею **FIX**;
- округлення з надлишком **FUP**.

**6.9.4.1 Округлення за правилами арифметики**

Якщо дробова частина числа більше 0,5, то округлення виконується до більшого цілого.

Розділ 6 Параметричне програмування для машини ZC42 - 65

Приклад 1:  $\#4 = RND10.2$  - в осередок 4 записати результат округлення числа 10,2. Запишеться число 10, тому що  $0,2 < 0,5$ .

Приклад 2:  $\#20 = RND10.50001$  - в комірку № 20 запишеться число 11, тому що  $0,50001 > 0,5$ .

Приклад 3:  $\#12 = RND - 10.3$  - в комірку № 12 запишеться число -10.

Приклад 4:  $\#45 = -10.5$  - в комірку № 45 запишеться число -11.

**6.9.4.2 Округлення з недостатчею**

Оператор **FIX** завжди округляє до меншого цілого.

Приклад 1:  $\#7 = FIX 8.8$  - в комірку № 7 буде записане число 8,0.

Приклад 2:  $\#3 = -8.8$  - в комірку № 3 запишеться число 9.

**6.9.4.3 Округлення з надлишком**

Оператор **FUP** завжди округляє до більшого цілого.

Приклад 1:  $\#4 = FUP 100.5$  - в комірку № 4 запишеться число 101.

Приклад 2:  $\#5 = FUP - 105.4$  - в комірку № 5 запишеться число -105.

**6.10 Визначення модуля числа**

Оператор визначення модуля числа програмується як **ABS**.

Приклад  $\#3 = ABS \##8$  означає, що в комірку № 3 буде записаний модуль числа, що знаходиться в комірці формальної пам'яті, номер якої зберігається в комірці № 8.

**6.11 Умовні вирази**

Умовними називаються такі вирази, у яких запрограмована перевірка певної умови. Є такі умовні вирази:

- рівності (=);
- нерівності (/=);
- більше (>);

Розділ 6 Параметричне програмування для машини ZC42 - 65

- менше (<);
- не більше (>=);
- не менше (<=).

Умовні вирази можуть починатися з числа. Наприкінці умовного виразу потрібно запрограмувати оператор передачі керування тієї частини програми, яку потрібно виконувати, якщо виконано дану умову (або не виконано).

Програмування оператора передачі керування буде розглянуто нижче, тому в прикладах ми його будемо заміняти символом L.

Приклад 1:  $\#28 == \#17(L)$  - відбувається перевірка рівності вмісту комірок № 28 і № 17. Якщо ці числа рівні, то відбудеться передача керування відповідно до оператора L.

Приклад 2:  $\#1 >= 0(L)+1$  - відбувається порівняння вмісту комірки № 1 з нулем. Після цього до числа, записаного в цій комірці, додається одиниця, і керування передається операторові L.

Що відбудеться, якщо умова не виконується, ми розглянемо далі.

### 6.12 Текстові вирази

Текстові вирази, що необхідні для кращого розуміння програми, формуються за таким форматом:

$\# \& \text{повідомлення} \&$ .

Довжина повідомлення не повинна бути більшою 27 символів. У текстовому виразі припустимі будь-які букви і символи, включаючи букви російського алфавіту.

Стирається повідомлення записом пропуску в текстовому виразі:

$\#7 \text{пропуск} \&$ .

### 6.13 Структура програми

Керуюча програма, написана мовою макропрограмування, складається з послідовності кадрів двох типів:

- технологічних;
- обчислювальних.

**6.13.1 Технологічний кадр**

Це звичайний кадр, що містить завдання на виконання будь-яких технологічних функцій, як у випадку традиційного програмування. Особливістю технологічного кадру при макропрограмуванні є те, що завдання може програмуватися через номер комірки, де зберігається дана команда.

Через номер комірки програмуються:

- код **G** – функції (дивися таблицю 5.1);
- задання переміщення за адресами **X, Y, Z**;
- задання векторів **I, J, K**;
- задання робочої подачі **F**;
- задання частоти обертання шпинделя **S**;
- задання номера інструмента **T**;
- задання номера коректорів **D** і **H** (дивися таблицю 5.1);
- задання величини паузи **E**;
- задання кількості повторень ділянки програми **L**.

Приклад 1: **G & 6005** - запрограмована функція, код якої знаходиться в комірці системної пам'яті № 6005.

Приклад 2: **X #25Y #72F #6** - запрограмоване переміщення по осях **X** і **Y**. Величини цих переміщень знаходяться в комірках № 25 і №72 відповідно. Подача для цього переміщення зберігається в комірці № 6.

При такому методі задання переміщень величини цих переміщень або координат точок повинні бути записані в пам'ять системи раніше, ніж до них буде відбуватися звертання в програмі.

Адреси **M, N, G, P** програмувати через номер комірки пам'яті не можна. Виняток складають команди **M3, M4, M5** і **G32**.

**6.13.2 Обчислювальний кадр**

Це кадр, що містить будь-який вираз мови макропрограмування.

Він може починатися або з номера кадру **N**, або з номера комірки. Обчислювальний кадр не може починатися з цифри. Довжина обчислювального кадру так само, як і технологічного, не повинна бути більше 128 символів, включаючи і **ПС**.

Розділ 6 Параметричне програмування для машини 2С42 - 65

Обчислювальний кадр може містити кілька виразів, розділених пробілом (.....)

В обчислювальному кадрі не можна задавати технологічні команди.

### **6.14 Керування виконанням програми**

Програма, написана мовою макропрограмування, виконується в логічній послідовності заданою нумерацією кадрів, тобто так само, як і звичайна керуюча програма. Але в мові макропрограмування передбачене керування виконанням програми за допомогою оператора передачі керування.

#### **6.14.1 Номер кадру**

Номер кадру задається за адресою N. Перед обчислювальним кадром завдання номера кадру необов'язково, якщо цьому кадрові не буде передаватися керування.

Якщо певному обчислювальному кадрі буде передаватися керування, то завдання номера кадру обов'язково. При цьому його формат повинен містити чотири розряди. Опускати нулі перед значущими цифрами не можна.

Приклад 1: *N1X#24Y#25* - кадр заданий правильно, але такому кадрові передавати керування не можна.

Приклад 2: *N0001X#24Y#25* - такому кадрові можна передавати керування.

#### **6.15 Оператор передачі керування**

Це команда, за допомогою якої передається керування будь-якому кадрові програми (якому за наведеною вище умовою можна передавати).

Задавати цю команду можна тільки в обчислювальному кадрі. Оператор передачі керування може бути безумовним і умовним.

##### **6.15.1 Безумовний оператор передачі керування**

Програмування безумовного оператора виконується за таким форматом:  *#(L),*

Розділ 6 Параметричне програмування для машини 2С42 - 65

де L – номер кадру, якому треба передати керування без перевірки певної умови (нулі перед цифрами можна опускати).

Якщо номер кадру записаний позитивним числом, то передача керування буде зроблена кадрові, що розміщений нижче від виконуваного кадру.

Якщо номер кадру записаний зі знаком мінус, то керування передається кадрові програми, що розміщений раніше ніж виконуваний кадр.

Приклад 1: N0010.....

.....  
 .....  
 # 1 = # 2 + SIN 45 ..... # (-10).

Після виконання обчислення і запису результату в комірку №1 відбудеться передача керування кадру N0010, розміщеному вище від даного кадру.

Приклад 2: N 1 X2000 Y # 79

N0010 Z0

## 44 = & # 1 + 1

# (19)

.....  
 .....  
 N0019 ## 44 = # 7

Після виконання обчислення відбудеться передача керування кадру N0019, що розміщений нижче по тексту програми.

### 6.15.2 Умовний оператор передачі керування

Ця команда задається тільки наприкінці умовного виразу і передає керування кадрові, зазначеному в операторі тільки при виконанні заданої умови. Якщо умова не виконується, то відбувається відпрацювання наступних кадрів.

Приклад 1: # 1 < & # 70 ( - 15) – якщо величина, що знаходиться в комірниці №1, менше від величини, що знаходиться в комірниці системної пам'яті, номер якої зберігається в комірниці №70 формальної пам'яті, то керування передається кадру N0015,



Розділ 6 Параметричне програмування для машини ZC42 - 65

розміщеному вище даного кадру. Якщо нерівність не виконується, то відпрацьовується наступний кадр програми.

Приклад 2:  $N 10 \# 66 - 1 > 0$  (900)

$N 11$ .....

У кадрі №10 з числа, що знаходиться в комірці №66, віднімається одиниця, і число, що залишилося, порівнюється з нулем. Якщо воно ще більше нуля, то керування передається кадру № 0900, що розміщений у тексті програми нижче. Якщо умова не виконується, то відпрацьовується кадр № 11.

**6.15.3 Призначуваний оператор передачі керування**

Ця команда може використовуватися і як умовний, і як безумовний оператор керування. Особливістю команди є те, що кадр, якому передається керування, знаходиться не в тексті програми, а в комірці формальної пам'яті, куди записується попередньо.

Приклад 1:  $\#17 + 5$ ..... $\#$ (#20).

Цей запис читається так: до числа, записаного в комірці №17 додається 5 і відбувається безумовна передача керування кадру, номер якого записаний в комірці 20.

Приклад 2:  $\&1001 == 0$ ..... $\#$ (#1)- якщо величина, записана в комірці системної пам'яті 1001, дорівнює нулю, то виконується кадр, записаний в комірці №1 формальної пам'яті.

**6.16 Послідовність виконання операцій у виразі**

Операції у виразі виконуються в тій послідовності, у якій ми їх пишемо в програмі, тобто зліва направо.

Оператори у виразі виконуються відповідно до їх пріоритету. Оператори, пріоритет яких вище, виконуються першими. Оператори, що мають рівні пріоритети, виконуються в порядку їхнього запису.

Зростання пріоритетів відбувається в такому порядку:

- оператор передачі керування;
- арифметичні, логічні, умовні оператори й оператори присвоєння;
- оператори функцій.

Розділ 6 Параметричне програмування для машини 2С42 - 65

Якщо у виразі записано декілька операторів передачі керування, то виконується тільки останній.

Приклад 1:  $\#1 = 2 + \#2 - 10 * \#3 / \sin 30$ . Даний вираз буде виконуватися в такій послідовності: в комірку № 1 записується число 2, потім підсумовується вміст комірок № 1 і № 2 і результат поміщається в комірку № 1 (неявне присвоєння); потім із вмісту комірки № 1 віднімається число 10, а результат збільшується на вміст комірки № 3, добуток поміщається знову в комірку № 1. Потім знаходять синус  $30^{\circ}$ , тому що пріоритет цієї функції вище, ніж операції розподілу, після цього вміст комірки № 1 ділять на  $\sin 30^{\circ}$  і результат усього виразу буде в комірці № 1.

Оператори множення і ділення перед операторами додавання і віднімання пріоритету не мають.

Приклад 2:  $\#66 = 3.3 * \#25 = RND \#66$ .

У комірку № 66 записується число 3,3, після чого збільшується на вміст комірки № 25. Результат множення округляється до цілого і записується в комірку № 66.

Приклад 3:  $\#(10) \dots \#(20) \dots \#(30)$  - записано три безумовних оператори передачі керування. У результаті буде виконаний тільки останній. Тобто керування буде передано кадру, що знаходиться в комірці № 30.

Приклад 4:  $\#1 == \#2(10) \dots \#(12)$  - при виконанні рівності чисел, що знаходяться в комірках № 1 і № 2, керування буде передано кадру програми, номер якого знаходиться в комірці № 12.

Приклад 5:  $\#2 >= 5.73(100) \vdash \#3 + 1$  - перевіряється умова «більше або дорівнює», і незалежно від результату перевірки до вмісту комірки № 2 додається вміст комірки № 3 і ще додається одиниця. Результат цих обчислень записується в комірку № 2. Передача керування відбудеться тільки тоді, коли після виконання цих операцій буде виконана умова ( $>=$ ).

Цей приклад можна записати ще так:

1) #2 + #3 + 1;

2) # >= 5.73(100).

### **6.17 Розроблення підпрограм**

При програмуванні даною мовою так, як і при звичайному програмуванні, можливе використання підпрограм. Розроблення підпрограм і звертання до них виконуються за тим самими правилами, що і при звичайному програмуванні. Однак тут є деякі особливості, що ми далі розглянемо.

#### **6.17.1 Виклик підпрограм**

Виклик підпрограми задається у звичайному технологічному кадрі і виконується тільки наприкінці виконання кадру.

Приклад 1:  $N \dots X100F \#6Z \#26P03$ .

За програмою відбудеться відпрацьовування по осях X і Z з подачею F, а потім викликається підпрограма № 3.

Після команди виклику підпрограми можна записувати тільки параметри, що належать до цієї підпрограми, і нічого іншого.

Приклад 2:  $P05 \#1 = 10 \#2 = 100 \#3 = 5.5$ .

Тут зазначено для підпрограми № 5, що в комірці № 1 записане число 10, в комірці № 2 – число 100, а в комірці № 3 – 5,5.

*Викликати підпрограму в обчислювальному кадрі заборонено.*

#### **6.17.2 Відпрацьовування підпрограм і повернення в основну програму**

Якщо у підпрограмі не задані оператори передачі керування, то повернення в основну програму відбудеться за командою «кінець підпрограми» (M 99).

За наявності оператора передачі керування треба організувати керування програмою так, щоб у будь-якому випадку кадр із командою M 99 не був пропущений.

Розділ 6 Параметричне програмування для машини 2С42 - 65

Не можна передавати керування кадрам підпрограми без її виклику.

Програмуючи оператори передачі керування, треба намагатися, щоб не було звертання до кадрів, що не входять у підпрограму. Розглянемо це на прикладах, наведених нижче.

Після виконання підпрограми P01 у прикладі 1 повинен виконуватися кадр № 0101. У підпрограмі відбудеться перевірка рівності вмісту комірки № 1 числу 11. Якщо ця умова виконується,

Приклад 1:

$N...P01\#1 = 10\#2 = 100\#3 = 5.5$

$N0101X\#24$

.....

..... $N.....M02$

:01

$\#1 == 11(150)$

$\#25 = \#2 - \#1\#(151)$

$N0150\#24 = \#3$

$N0151M99$

то керування передається кадру № 0151, де записана команда M 99. Якщо умова не виконується, то після розв'язання виразу

$\#25 = \#2 - \#1$  керування безумовно передається кадру

№ 0151. Тобто команда M 99 у будь-якому випадку виконається.

Приклад 2:  $P01\#1 = 10\#2 = 100\#3 = 5.5$

$N0101X\#24$

$N.....$

.....

$N.....M02$

: 01

# 1 = = 11 (152)

# 24 = # 2 - # 1

N0151 M 99

N 0152 # 24 + # 3 # (-151)

У прикладі 2 кадр № 0152 не включений у підпрограму, але безумовний оператор керування (-151) передасть керування кадру № 0151 і команда M 99 буде виконана.

### 6.17.3 Виконання програм

Тепер ми знаємо, що і програма, і підпрограма складаються з двох типів кадрів: обчислювальних і технологічних.

Обчислювальні кадри потрібні для підготовки інформації в технологічні кадри, а на верстат ці кадри не передаються і не відпрацьовуються.

Таким чином, обчислювальні кадри, у яких записані оператори передачі керування програмою, визначають лише логічну послідовність виконання програми, у тому числі і технологічних кадрів.

Приклад:

N 1 G01 X1000 Y5500 F300

N 2 Y1010 Z-50

N 3 #1 = 5000 \* # 3 ..... # 3 = ## 69

N 4 X#1 F#23

Обчислювальний кадр № 3 виконується разом з кадром № 4 і переривання в рухові інструмента не буде.

## 6.18 Приклади програм, виконаних мовою макропрограмування

### 6.18.1 Програма 1

Ескіз деталі для програмування обробки наведений на рисунку 6.1.

Необхідно зацентрувати, просвердлити і зняти фаски у восьми отворах фланця.

Для роботи необхідні такі інструменти: свердло центрове, свердло спіральне 12мм, зенківка з кутом 45°.

Розділ 6 Параметричне програмування для машини 2С42 - 65

Верстат одноінструментний без автоматичної коробки швидкостей, тому зміну інструмента і переключення чисел обертів шпинделя виконуємо вручну. Для цього в програмі передбачаємо команду «безумовний зупинник» (M00).

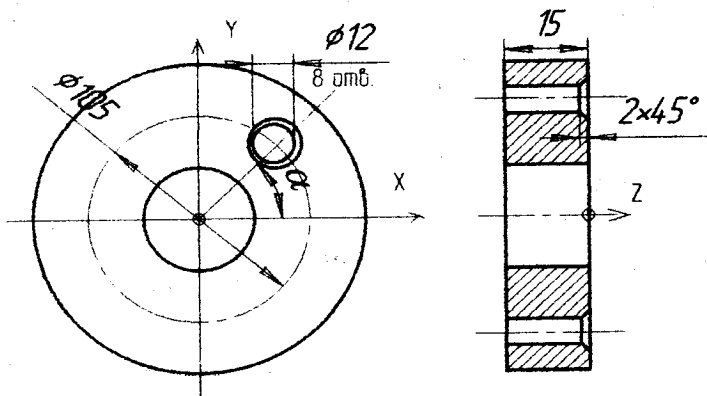


Рисунок 6.1-Ескіз деталі для прикладу макропрограмування

Формальну пам'ять системи використовуємо так:

- # 1 – запишемо діаметр, на якому розміщуються отвори, мм;
- # 2 - запишемо число отворів, розмішених на цьому діаметрі;
- # 3 – кут розміщення першого отвору відносно осі X;
- # 4 – номер коректора на довжину інструмента;
- # 5 – U – позиція початкової точки руху інструмента на робочій подачі;
- # 6 – Z – позиція кінцевої точки руху інструмента на робочій подачі;
- # 7 – I – координата виходу інструмента з отвору на швидкому ході;
- # 8 – F – величина робочої подачі, мм/хв.

001 ПС – номер програми

Розділ 6 Параметричне програмування для машини 2C42 - 65

% PC – початок програми

N1 G 54 G 90 PC - зазначена система координат деталі і спосіб завдання розмірів – абсолютні розміри G90

# 1 = 105 # 2 = 8 # 3 = 30 # 4 = 1

# 5 = 3 # 6 = -3.5 # 7 = 15 # 8 = 50 PC - у формальну пам'ять записані параметри отворів

N2 P 33 PC - виклик підпрограми № 33

N3 M00 PC - команда «безумовний останов» для зміни інструмента (установлюється спіральний свердло)

N4 G 54 G 90 # 6 = -25.5 # 8 = 60 PC - вводяться змінні параметри циклу G81

N5 P 33 PC - знову викликається підпрограма № 33

N6 M 00 PC - безумовний останов для заміни свердла на зенківку

N7 G 54 G 90 # 4 = 3 # 6 = - 2 # 8 = 75 PC - вводяться змінні параметри для зенкування фаски

N8 P 33 PC - знову викликається підпрограма № 33

N9 M 02 PC - кінець програми

N10 M 30 PC - повернення в початок програми

: 33 PC - початок підпрограми №33

# 1 \* 100 # 5 \* 100 # 6 \* 100 # 7 \* 100 PC - перекладання геометричної інформації в дискретність відпрацьовування (0,01)

# 67 = # 1/2 # 66 = # 3 # 68 = 360/# 2 # 69 = 0 PC - знаходимо радіус, на якому розташовані отвори, поділивши діаметр

окружності на 2, результат запишемо в комірку 67. В комірку 66 запишемо кут розташування першого отвору  $30^{\circ}$ . В комірку 68 запишемо дискрету кута між отворами ( $360/8=45^{\circ}$ ). В комірку 69

запишемо 0; ця комірка буде служити лічильником оброблених отворів

N0005 # 64 = # 67 \* cos # 66 PC - в комірці 67 у нас знаходиться радіус, помножуючи його на косинус початкового кута першого

отвору ( $30^{\circ}$ ), знаходимо координату X першого отвору. Результат запишемо в комірку 64

Розділ 6 Параметричне програмування для машини 2С42 - 65

65 = # 67 \* sin # 66 ПС - аналогічно знаходимо координату У першого отвору. Результат помістимо в комірку 65

0006 G01 G09 X # 64 Y # 65 F4500 ПС - прискорений підхід інструмента в координати першого отвору

0007 G09 G43 H # 4 Z # 5 ПС - підхід на швидкому ходу до початкової точки свердління по осі Z із уведенням корекції на довжину інструмента (G43), величина корекції задана при налаштуванні верстата в коректор, адреса якого зберігається в комірці №4

0008 Z # 6 F # 8 ПС - виконується свердління отвору на глибину, величина якої записана в комірку №6, з подачею, що знаходиться в комірці №8

0009 G09 Z # 7 F4500 ПС - вихід свердла з отвору на швидкому ходу на величину, що зберігається в комірці №7

69 + 1 ПС - у лічильник числа оброблених отворів додається одиниця

66 + # 68 ПС - до кута першого отвору ( $30^{\circ}$ ) додається величина розміщення отворів (кутовий крок  $45^{\circ}$ ), результат записується в комірку 66

66 > 180 (10) ПС - якщо при підсумовуванні утворився кут більше  $180^{\circ}$ , то переходимо до кадру № 0010. Якщо ця умова не виконується, то до кадру № 0011

(11) ПС - перехід до кадру № 0011

0010 # 66 = # 66 - 360 ПС - якщо кут став більше  $180^{\circ}$ , то синус і косинус його стають негативними величинами і кадр №0010 дозволяє зробити це реальним. Результат запишемо в комірку №66

0011 # 69 = # 2 (15) - перевірка стану лічильника оброблених отворів. Якщо оброблені всі отвори, записані в комірку №2, то треба переходити до кадру № 0015.

# (-5) - якщо ця умова не виконана (ще не всі отвори оброблені), то переходимо до кадру №0005, де знову обчислюються координати наступного отвору й обробляється наступний отвір (№ 0006)



Розділ 6 Параметричне програмування для машини 2С42 - 65

N0015 G09 G 49 H # 4 Z0 - якщо лічильник просвердлених отворів показав збіг заданого і виконаного чисел отворів, то скасовується корекція на довжину інструмента (G49), величина якої знаходиться в комірці 4, і по осі Z інструмент обертається в нуль з гальмуванням наприкінці кадру (G09)

N0016 M05 - стоп шпindel

N0017 M99 - кінець підпрограми

N0020 M30 - кінець програми з поверненням у початок

Таким чином, суть даної програми зводиться до того, що після центрування відбувається безумовна зупинка (M00). Оператор установлює свердло і знову відбувається свердління за даною програмою. Коли всі отвори просвердлені, відбувається зупинка (M00) і оператор установлює зенківку, потім запускає верстат і програма повторюється третій раз.

Керування відпрацюванням і розрахунком відбувається за такою схемою:

- визначається кут між отворами;
- знаходяться координати отвору;
- обробляється отвір;
- у лічильник додається одиниця;
- до попередньої кутової координати отвору додається кутовий крок;
- перевіряється знак синуса і косинуса;
- перевіряється стан лічильника отворів;
- знаходяться координати наступного отвору;
- обробляється наступний отвір;
- у лічильник додається одиниця, і так доти, поки стан лічильника не стане дорівнювати числу заданих отворів.

### 6.18.2 Програма 2

Ескіз виробу наведений на рисунку 6.2.

Фреза дискова діаметром 40мм, товщиною 2мм. Лунки розміщені в шаховому порядку з кроком по висоті втулки 3,5мм. Число лунок у ряді 8 штук.

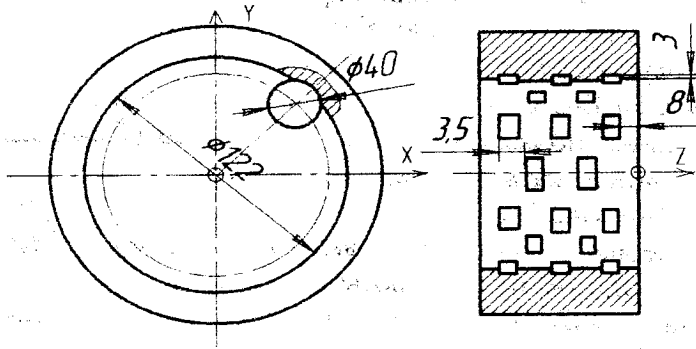


Рисунок 6.2 - Ескіз для прикладу програми фрезерування лунок шляхом радіального врізання дисковою фрезою

Керуюча програма:

002

% ПС

# 1 = 122 – внутрішній діаметр втулки, мм

# 2 = 3.0 – глибина лунки, мм

# 3 = 40 – діаметр дискової фрези

# 4 = 8 – число лунок у ряді

# 5 = 5 – число рядів по висоті втулки

# 6 = 3.5 – крок лунок по висоті втулки

# 7 = 8 – відстань по осі Z від нуля до кромки першої лунки

# 8 = 100 – величина робочої подачі, мм/хв

# 1 \* 100 # 6 \* 100 # 7 \* 100 # 2 \* 100 # 3 \* 100 – перекладення геометричних розмірів у дискрету верстата (0,01 мм)

# 10 = # 1 \* 0.5 - 3 \* 2 - # 3 \* 0.5 – знаходимо радіус окружності для безпечного підходу центра фрези до поверхні втулки, щоб фрезерувати шляхом радіального врізання. Результат запишемо в комірку №10

(122\*0,5=61; 61-3\*2=55, 40\*0,5=20, 55-20=35)

# 11 = # 10 + # 2 + 3 - визначення радіуса окружності дна лунок (40+3+3=46- положення осі обертання фрези наприкінці врізання)

Розділ 6 Параметричне програмування для машини 2С42 - 65

- # 12 + 360 / # 4 - знаходимо кутовий крок лунок ( $360:8=45^{\circ}$ )
- # 13 = 1 - комірка №13 буде лічильником рядів, тому перед виходом у перший ряд у неї запишемо 1
- N0013 G00 G90 G43 G54 G17 X0 Y0 - прискорений вихід фрези в нуль координат по осях X і Y, фіксування нуля (G54), задання площини, у якій лежать дуги окружності (G17)
- N0014 G01 G09 G43 H01 Z # 7 F 2000 M03- опускання фрези на виконання першого ряду лунок і включення правого обертання шпинделя. Одночасно задаються корекція по довжині інструмента (G43) і номер коректора, де ця величина знаходиться (H01)
- N0015 # 30 + # 4 - перевірка числа оброблених лунок у ряді
- # 9 + # 13/2 - в комірку 9 записується частка від розподілу вмісту лічильника рядів на 2
- # 19 = # 9 - в комірку 19 записується вміст комірки 9
- # 9 = RND # 9- вміст комірки 9 округляється до цілого
- # 9 = # 19 (16)- якщо ряд парний, то переходимо до виконання кадру № 0016
- # 20 = 90 - # 12 - якщо ряд непарний, то початковий кут лунки  $90^{\circ} - 45^{\circ}$
- # (17) - і тоді перехід до кадру 0017
- N0016 # 20 = 90 \* 2 - # 12 \* 0.5 - якщо ряд парний, то початковий кут лунки буде  $90^{\circ} - 45^{\circ} \times 0,5$
- N0017 # 20 + # 12 - початковий кут збільшується збільшенням кутового кроку  $-45^{\circ}$
- # 20 > 180 (18) - якщо в результаті підсумовування утворився кут більше  $180^{\circ}$ , то переходимо до виконання кадру № 0018
- # (19) - якщо ні, то виконуємо кадр № 0019
- N0018 # 20 = # 20 - 360 - перевіряємо знак кута. Якщо він більше  $180^{\circ}$ , то тригонометричні функції будуть негативними
- N0019 # 21 = # 10 \* cos # 20 # 22 + # 10 \* sin # 20 # 24 = # 11 \* sin # 20 - визначення координат першої лунки (дивися рисунок 6.3)
- N0020 G01 G09 X # 21 Y # 22 F 2000 - підхід до лунки в точку 1
- N0021 X # 23 Y # 24 F # 8 - фрезерування лунки

Розділ 6 Параметричне програмування для машини 2С42 - 65

N 0022 G09 X # 21 Y # 22 F 1200 – вихід з лунки на великій подачі

# 30 – 1 – число лунок зменшується на одиницю

# 30 == (23) – якщо зроблені всі лунки ряду, то переходимо до кадру № .0023

# (17) – якщо ні, то виконуємо кадр № 0017

N0023 # 5 – 1 – число рядів лунок зменшуємо на одиницю

# 5 == 0 (25) – якщо оброблені всі ряди, то переходимо до кадру № 0025.

# 13 + 1 – встановлюємо номер наступного ряду

# 7 + # 6 – знаходимо координату Z наступного ряду

# (-15) – переходимо до відпрацювання кадру № 0015

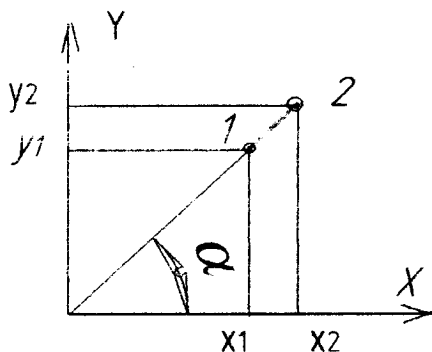
N0025 G 01 G09 X0 Y0 F2000 – відвід фрези в нуль деталі

N 0026 G09 Z5000 M05 – підйом фрези у вихідне положення

N0027 G09 G49 Z – 1000 – скасування корекції на довжину інструмента

N0028 M02

N0029 M30



*Рисунок 6.3 – Схема розрахунку координат точки 1 підходу фрези до стінки отвору втулки і точки 2 – кінця фрезерування лунки*

Алгоритм розрахунків і виконання керуючої програми виконується за такою схемою:

- знаходиться координата точки 1 кожної лунки (початок фрезерування);
- знаходяться координати точки 2 – кінці фрезерування лунки;

Розділ 6 Параметричне програмування для машини 2С42 - 65

- відбуваються фрезерування лунки і швидкий вихід фрези з лунки;
- знаходиться кутове положення наступної лунки, а в лічильник лунок у ряді додається 1;
- перевіряється знак синуса і косинуса;
- перевіряється число в лічильнику лунок на відповідність заданому;
- перевіряється число виконаних рядів;
- якщо ще немає відповідності, то знаходяться координати наступної лунки і відбувається її фрезерування;
- якщо число виконаних лунок у ряді збіглося із заданим, то відбувається визначення парності ряду і перехід на наступний ряд із зміщенням кута першої лунки на половину кутового кроку;
- якщо всі ряди виконані, то відбувається вихід фрези у вихідне положення.

**Розділ 7****Програмування обробки деталей на верстатах із системою керування WL4T****7.1 Призначення і характеристика машини**

Система керування WL4T є сучасною контурною системою програмного керування і призначена для установки на двох - і три - координатних верстатах (токарних , токарно-карусельних та фрезерних).

7.1.1 Число керованих координат – 2 + 1 (план – шайба) або 3.

7.1.2 Способи інтерполяції – лінійна і кругова.

7.1.3 Дискретність переміщення виконавчого механізму – 0,001мм.

7.1.4 Швидкість переміщення виконавчого механізму від 0 до 32м/хв. Граничне значення подачі встановлюється параметром, який оператор вводить з робочого терміналу.

7.1.5 Величина корекції радіуса інструмента до 32мм.

7.1.6 Програмоносії – твердотільний диск флеш – пам'яті від 3Мб до 16Мб.

7.1.7 Візуальний контроль відпрацьовування програми – на екрані монітора.

7.1.8 Геометрична інформація вводиться в абсолютних координатах і збільшеннях. Ціла частина числа відокремлюється від дробової крапкою.

7.1.9 Геометрична і технологічна інформація записується в програмі в тому самому вигляді, як на кресленні, або в технологічному процесі.

**7.2 Адреси та їх призначення**

% - ім'я програми, може позначатися цифрами 0 – 9 або буквами від A до Z.

N – номер кадру від 1 до 65535.

D – номер коректора, де записана величина корекції на радіус інструмента (0 – 127).

G – службова функція – визначає вид руху виконавчого механізму.

Розділ 7 Програмування обробки деталей на верстатах із системою керування WLS/T

X, Z – абсолютні координати точок.

U, W – збільшення за цими координатами. Програмувальні межі геометричної інформації від – 9999.999 до + 9999.999.

I, K – параметри радіуса вектора при круговій інтерполяції. Програмувальні межі від – 4999.999 до + 4999.999.

L – команда виклику підпрограм (00 – 99).

E – пауза, тривалість якої знаходиться у межах від 0 до 99999.999с.

F – величина робочої подачі. Програмувальні межі від 0.001 до 99999.99. Розмірність мм/хв або мм/об.

P – початкова фаза синхронізації шпинделя при нарізуванні різі. Програмувальні межі від 0 до 359.999 град.

S – задання частоти обертання шпинделя або швидкості різання.

T – номер інструмента, який необхідно установити в робоче положення.

M – допоміжна функція.

### **7.3 Структура кадру програми**

Кадр програми починається з адреси N і закінчується знаком кінець кадру - LF. У межах цих знаків інформація в кадрі програми може розміщуватися в будь-якому порядку. Всі адреси в кадрі можна задати тільки один раз. Виняток становлять адреси G і M, що у межах кадру можуть бути використані кілька разів.

Довжина кадру не повинна перевищувати 80 символів.

Нумерація кадрів необов'язково повинна бути підряд, можна її вести і з пропусками. Адреси N і номер кадру можна опускати, але тоді цей кадр не можна використовувати в режимі «пошук кадру».

У програмі можуть бути кадри з коментарями. Це обов'язково окремі кадри.

Такий кадр не має номера, а починається або з точки, або з зірочки (\*).

**7.4 Службові функції**

Службові функції задаються за адресою G числом від 0 до 99. Перелік функцій, використовуваних у системі WL4T, наведений у таблиці 7.1.

Функції G0, G40, G53, G90, G95, G97 встановлюються автоматично при ввімкненні живлення системи ЧПК.

У кадрі програми можуть бути задані кілька службових функцій з різних груп. Записувати їх потрібно в порядку зростання.

Службові функції однієї групи скасовують одна одну.

Функції групи 0 діють тільки в межах того кадру, де вони задані.

Якщо в кадрі програми помилково записано кілька функцій однієї групи, то буде виконана остання.

*Таблиця 7.1 - Службові функції*

Код функції	Група	Призначення функції
0	0	Швидкий рух
1	1	Лінійна інтерполяція
2		Кругова інтерполяція за годинниковою стрілкою
3		Те саме, але проти годинникової стрілки
4	0	Пауза
9	0	Гальмування в кінці кадру
31	0	Богатопрхідний цикл нарізування різі
32		Різьбовий рух
33		Нарізування різі мітчиком або плашкою
70 - 75	0	Технологічні цикли



*Продовження таблиці 7.1*

1	2	3
40	2	Відміна корекції на радіус
41		Корекція на радіус інструмента – інструмент ліворуч від контуру деталі
42	3	Корекція на радіус інструмента – інструмент праворуч від контуру деталі
53		Повернення до системи координат верстата
54		Вибір координатної системи заготовки 1
55		Вибір координатної системи заготовки 2
92	4	Задання координатної системи заготовки від програми
90		Задання розмірів в абсолютних координатах
91	5	Задання розмірів у збільшеннях
94		Розмірність подачі, мм/хв
95		Розмірність подачі, мм/об
96	6	Умова сталості швидкості різання, розмірність, м/хв
97		Частота обертання шпинделя задана у об/хв
80-89	7	Постійні цикли

**7.5 Допоміжні функції**

Допоміжні функції програмуються за адресою M. Перелік функцій, зона їх дії, початок дії і групи функцій наведені в таблиці 7.2

Якщо в кадрі програми помилково записані кілька функцій однієї групи, то буде виконана остання.

Розділ 7 Програмування обробки деталей на верстатах із системою керування W/LAT

Функції різних груп можуть бути задані в одному кадрі програми.

Таблиця 7.2 - Допоміжні функції

Код функції	Група	Початок дії		Час дії		Найменування
		до початку руху	після руху	до вистання такої, що відміняє	в межах кадру	
M0	1		+		+	Безумовне зупинення
M1			+		+	Зупинка з підтвердженням
M2				+		+
M3	2			+		Праве обертання шпинделя
M4		+		+		Ліве обертання шпинделя
M5			+	+		Зупинка шпинделя
M6	3	+			+	Зміна інструменту
M7 - 8	4	+		+		Вимкнуті охолодження
M9		+		+		Вимкнуті охолодження
M30	1		+		+	Кінець програми з поверненням на початок
M41 - M48	5 +			+		Діапазони регулювання частоти обертання шпинделя №1 - 8
M99	1		+		+	Кінець підпрограми

**7.6 Програмування подачі**

Робоча подача програмується за адресою F у тій розмірності, що задається функціями G94 або G95. Причому функція G95 встановлюється автоматично при включенні живлення. У цьому випадку подача має розмірність мм/об. Якщо потрібно задавати подачу в мм/хв, то треба задати функцію G94.

Розділ 7 Програмування обробки деталей на верстатах із системою керування WLAT

Наприклад, G94 F35.5 – подача 35,5мм/хв;

G95 F0.5 – подача 0,5мм/об.

За допомогою регулятора на пульті машини оператор може корегувати величину подачі, задану в програмі від 0% до 120%.

**7.7 Програмування головного руху**

Головний рух залежно від заданої функції G96 або G97 програмується за адресою S або об/хв (G97), або м/хв (G96). При цьому, задаючи функцію G96, ми забезпечуємо сталість швидкості різання, що відповідає в даний момент фактичному діаметрові обробки.

Оператор з пульта може корегувати задану швидкість різання в межах від 0% до 120%.

Функції G96 і G97 системою керування запам'ятовуються і діють до настання змінної функції.

Увага! Записуючи нове значення функції G96 або G97, потрібно у цьому самому кадрі задати величину обертів або швидкість різання за адресою S.

**7.8 Програмування паузи**

Пауза програмується функцією G4, а час паузи задається в секундах за адресою X. Величина паузи може бути задана в межах від 0,001с до 99999,999 с.

Функція G4 діє тільки в тому кадрі, де вона задана.

**7.9 Програмування гальмування наприкінці кадру**

Завданням функції G9 можна забезпечити точне підведення інструменту до заданої точки контуру деталі. При цьому швидкість руху зменшується до нуля.

У кадрах G0 цю функцію задавати не потрібно, тому що в алгоритмі прискореного переміщення закладений розгін до швидкості холостого ходу і гальмування до нуля при підході до заданої точки.

Функція G9 діє тільки в кадрі, де вона записана.

**7.10 Програмування геометричної інформації**

Геометрична інформація задається в абсолютних координатах за адресами X і Z або в збільшеннях за адресами U

Розділ 7 Програмування обробки деталей на верстатах із системою керування WLAT

і W. Інформація з адрес I і K задається завжди тільки в збільшеннях. Вказівками на спосіб задання інформації є функція G90 або G91. Причому функція G90 встановлюється автоматично при ввімкненні живлення.

Інформація з адрес X і U завжди задається на діаметр.

Функції G90 і G91 системою запам'ятовуються і діють до настання змінної функції.

Завдання розмірів деталі в абсолютних координатах полегшує процес корегування програми. А розміри деталі в збільшеннях рекомендується задавати в підпрограмах, коли виклик їх може здійснюватися в будь-якому місці робочої програми.

### **7.11 Програмування зміни інструменту**

Команда «зміна інструменту» задається номером інструменту за адресою T і функцією M6. Залежно від того, яка службова функція задана в програмі (G41 або G42 або G40), може включатися або не включатися корекція на радіус інструменту. Зміна інструменту повинна виконуватися у такому місці робочого простору, щоб не пошкодити деталь, верстат або інструмент.

Якщо інструмент змінювати не потрібно, а потрібно тільки поміняти коректори на довжину і радіус інструменту, то в програмі записується номер відповідного інструменту за адресою T без запису функції M6.

При записі в програмі команди T0 зміна інструменту не відбувається, але скасовуються всі зсуви і коректори, і переміщення інструменту відбувається в системі координат верстата ( $dXs=0$ ,  $dZs=0$ ,  $R=0$ ).

### **7.12 Лінійна інтерполяція**

Лінійна інтерполяція програмується функцією G1, після якої за адресами геометричної інформації (X,Z,U,W) задаються координати кінцевої точки відрізка прямої або збільшення за координатами. Спосіб задання геометричної інформації визначається функціями G90 або G91. При цьому треба пам'ятати про те, що величини, записані за адресами X і U,

Розділ 7 Програмування обробки деталей на верстатах із системою керування WLAT  
діють по діаметру. Наприклад: заданий діаметр деталі 45мм, у програмі запишемо X45; задане переміщення в радіальному напрямку 5мм, у програмі запишемо U10.

При лінійній інтерполяції результуюча подача пов'язана з подачами за координатами таким співвідношенням:

$$F = \sqrt{F_x^2 + F_z^2},$$

а величини перемішень за координатами пов'язані зі швидкостями рівнянням:

$$\frac{L_x}{F_x} = \frac{L_z}{F_z}$$

Приклад лінійної інтерполяції показаний на рисунку 7.1

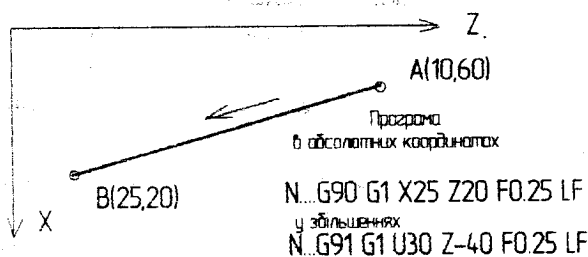


Рисунок 7.1 – Лінійна інтерполяція

### 7.13 Кругова інтерполяція

Кругова інтерполяція задається функціями G2 у випадку обходу дуги за годинниковою стрілкою, або G3 – у випадку обходу дуги проти годинникової стрілки. Далі в кадрі записуються адреси X і Z, при заданій функції G90 або U і W, якщо задано функцію G91. За даними адресам задаються координати кінцевої точки дуги або збільшення – відстані від початкової до кінцевої точки дуги. Потім у кадрі за адресами I і K, завжди в збільшеннях, задаються відстані від початкової точки дуги до її центра по осях X і Z. Якщо програмується обхід по дузі 360°, то адреси X,Z,U,W не задаються. У кадрі кругової

Розділ 7 Програмування обробки деталей на верстатах із системою керування W14T

інтерполяції залишаються тільки відстані від початку дуги до її центра, тобто I і K. Приклад кругової інтерполяції наведений на рисунку 7.2

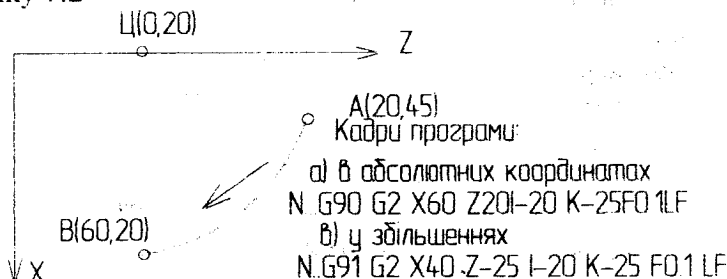


Рисунок 7.2 – Приклад кругової інтерполяції

### 7.14 Програмування технологічних циклів

Технологічні цикли на відміну від постійних (G80-G89) виконуються тільки один раз і в тому кадрі, де вони задані. Тоді як постійні, задані один раз будуть повторюватися до настання команди, що їх скасовує (G80).

#### 7.14.1 Багатопрохідний цикл нарізування різі різцем

Багатопрохідний цикл нарізування різі різцем програмується функцією G31. За цим циклом можна нарізати зовнішню, внутрішню, циліндричну, конічну різі будь-якого профілю з постійним і змінним кроком різі.

Вихідна точка циклу по осі X повинна бути на відстані від фактичного діаметра заготовки під різцю на 8 – 10мм, а по осі Z від початку різі на 1 – 3 кроки різі. Структура кадру із заданим циклом має такий вигляд:

G31 X... (U...) Z... (W...) F... P... I... #1... #2... #3... #4... #5...

де X (U) - фактичний діаметр заготовки під різцю або відстань до нього від вихідної точки циклу;

Z (W) – координата кінцевої точки різі або величина ходу різця до кінцевої точки різі;

F – крок різі, мм;

P – фаза початкової синхронізації (від 0° до 359.999°);

Розділ 7 Програмування обробки деталей на верстатах із системою керування WLAT

1 – збільшення кроку різі на один оберт при нарізуванні різі із змінним кроком;

#1 – глибина різі, завжди позитивна, береться на радіус у збільшеннях;

#2 – глибина різання, задається на радіус у збільшеннях;

#3 – конусність різі (різниця діаметрів) задається в збільшеннях на діаметр;

#4 – кут вривання у градусах (завжди позитивна величина);

#5 – модуль катета кінцевого збігу (величина позитивна).

Схема рухів циклу показана на рисунку 7.3.

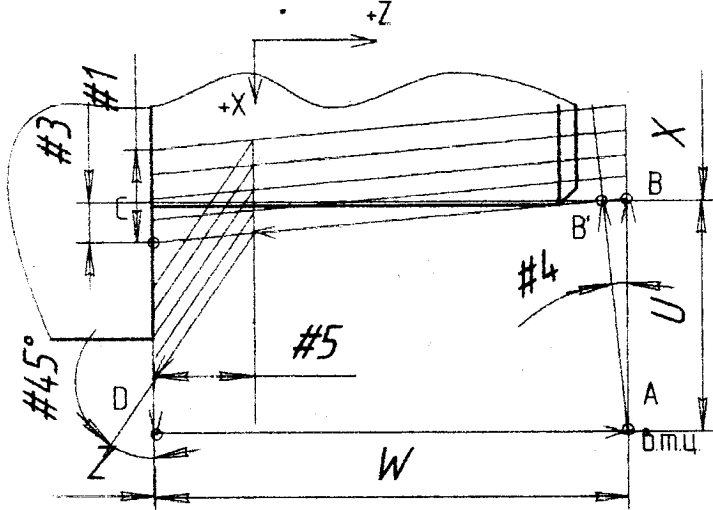


Рисунок 7.3 – Схема рухів при виконанні циклу G31

Рух починається з точки А, що є вихідною точкою циклу. Різець на швидкому ході переміщається в точку В при нульовому куті вривання або в точку В', якщо задано за адресою #4 необхідний кут вривання. У точці В при збігу фази шпинделя з фазою, заданою параметром Р, починається різальний рух до точки С. (Збіг різі також є нарізним рухом). По закінченні збігу різець прискорено відходить у точку D і повертається у вихідну точку циклу. У програмі циклу крок різі F можна не задавати, якщо задана раніше подача відповідала цьому крокові.

Розділ 7 Програмування обробки деталей на верстатах із системою керування W14T

Якщо у програмі циклу опущений параметр P, то машина розуміє це так, що фаза синхронізації дорівнює нулю.

Якщо опущено параметр I, то це означає, що він дорівнює нулю і нарізуються різі з постійним кроком.

Якщо опущено параметр #1, то це означає, що різь нарізується за один прохід. Тоді за адресою X треба задавати внутрішній діаметр різі. У цьому випадку і параметр #2 можна опускати. Але якщо задано глибину різі (#1), то опускати параметр #2 (глибину різання) не можна.

При нарізуванні циліндричної різі параметр #3 (конусність нарізки) у кадрі циклу не задається.

Значення параметра #4 (кут врізання) задається в межах від  $0^{\circ}$  до  $79.999^{\circ}$ . Опускати цей параметр не можна.

Значення параметра #5, яке дорівнює нулю, означає, що збіг різі відсутній (різь закінчується канавкою для виходу різця). Однак задання нульового значення цього параметра не рекомендується.

Якщо різь конічна, а кут конуса  $45^{\circ}$  і більше, то параметр #5 машиною ігнорується і при досягненні кінцевої точки різі різець на швидкому ході відходить у точку D.

Якщо в програмі заданий фактичний діаметр заготовки під різь менше, ніж розмір X вихідної точки циклу, то машина розуміє це так, що нарізується внутрішня різь. У цьому випадку збіг різі буде виконуватися у бік осі шпинделя, а задання конусності приводить до зменшення діаметра різі при рухові різця уздовж різі.

Розподіл припуску за проходами показано на рисунку 7.4.

Глибина різання, що задається в циклі параметром #2, залишається постійною, поки припуск, що залишився, не стане таким, що дорівнює або менше подвоєної глибини різання. Потім робиться прохід із глибиною різання t, що менше ніж #2, і дорівнює A, де A повинна бути сумою чотирьох проходів із глибинами різання:



$$A = \frac{\#2}{2} + \frac{\#2}{4} + \frac{\#2}{8} + \frac{\#2}{8}$$

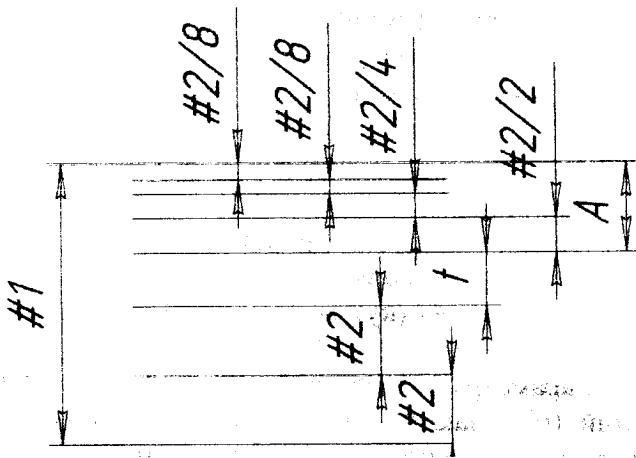


Рисунок 7.4 – Розподіл припуску за проходами

Увага! При виконанні нарізних рухів ручні регулятори процентної зміни подачі і частоти обертання шпинделя не діють. При цьому їх значення беруться такими, що дорівнюють 100%. Команди «СТОП ПОДАЧІ» і «СТОП ПРОГРАМА» на пульті оператора також заблоковані до кінця нарізного руху.

Приклад програмування нарізування різі різцем показаний на рисунку 7.5

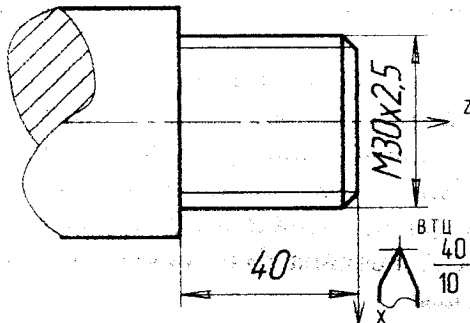


Рисунок 7.5

Фрагмент програми:

N.....G00 G1 X40 Z10 LF

N.....G31 X 29.8 Z-40 F2.5 P0 I0 #1=2 #2=0.15 #3=0 #4=15 #5=2.5

### 7.14.2 Різальний рух

Різальний рух програмується функцією G32 і має те саме призначення, як і в системі «Електроніка НЦ – 31». Цією командою ми можемо запрограмувати один або кілька проходів щодо формування повного профілю різі. Відмінною рисою цієї команди в даній керуючій машині є те, що функцію G32 ми записуємо тільки один раз, а потім задаємо тільки два її параметри X і Z. Діє функція G32 до настання однієї з функцій інтерполяції (G0, G1, G2, G3), що її скасовують.

Формат команди G32 для першого кадру

G32 X.....(U.....) Z.....(W.....) I..... F..... P.....

Для наступних кадрів тільки X.....(U.....)Z.....(W.....),

де X – координата діаметра різі, що буде отриманий у результаті даного проходу (або U – зсув до цього діаметра);

Z – координата кінцевої точки різі (або W – хід різця до цієї точки);

I – збільшення кроку різі на один її оберт, якщо нарізується різь із змінним кроком, що поступово збільшується або зменшується. Цей параметр задається зі знаком. Плюс – якщо крок збільшується по довжині нарізування, мінус – якщо зменшується. Знак плюс опускається. Якщо різь має постійний крок, цей параметр не задається.

F – крок різі також може бути опущений, якщо раніше був заданий у вигляді подачі;

P – фаза початкової синхронізації задається від  $0^0$  до  $359.999^0$ .

Вихідна точка циклу вибирається аналогічно до циклу G31.

*Останній прохід повинен містити обов'язково команду G9.*

### 7.14.3 Зворотно – поступальний різальний рух

Цей рух задається функцією G33 і має той самий формат, що і попередній рух. Використовується ця функція при нарізуванні різі мітчиком або плашкою. Рух характерний тим, що після досягнення кінцевої точки по Z відбуваються автоматичний реверс шпинделя і повернення інструменту у вихідну точку циклу.

### 7.14.4 Однопрохідний поздовжній цикл

Однопрохідний поздовжній цикл кодується функцією G70 і має таку саму структуру рухів, як і в системі «Електроніка НЦ – 31», що докладно розглянуто в розділі 2. Цикл призначений для чистової обробки циліндра, конуса і підрізування торця за командою одного кадру.

Цикл має такий формат:

G70 X...(U...)Z...W...)F...#1=...#2=...,

де X – величина початкового діаметра (або U – зсуву до нього);

Z – координата кінцевої точки по цій осі (або W – величина ходу різця до неї);

F – робоча подача по контуру руху;

#1= - довжина конічної частини траєкторії руху (завжди задається в збільшеннях, величина позитивна);

#2= - відстань по осі X від кінцевої точки конічної частини руху до вихідної точки. Ця величина береться в збільшеннях на діаметр і завжди позитивна. Якщо у деталі немає конічної частини, то ці два параметри не задаються, тому відбувається рух по циліндру до початку торця, потім підрізається торець і відбувається повернення у вихідне положення.

Величина подачі також не є обов'язковим параметром даного циклу.

Схема рухів циклу і положення вихідної точки циклу показані на рисунку 7.6.

**7.14.5 Однопрохідний поперечний цикл**

Цикл абсолютно ідентичний до розглянутого в розділі 2, лише відрізняється, як і попередній, адресами параметрів циклу (замість адреси P записуємо #1= або #2=).

Формат циклу

G71 X...(U...) Z...(W...) F... #1=... #2=...

Призначення параметрів циклу те саме, що й у попередньому. Структура рухів показана на рисунку 7.7.

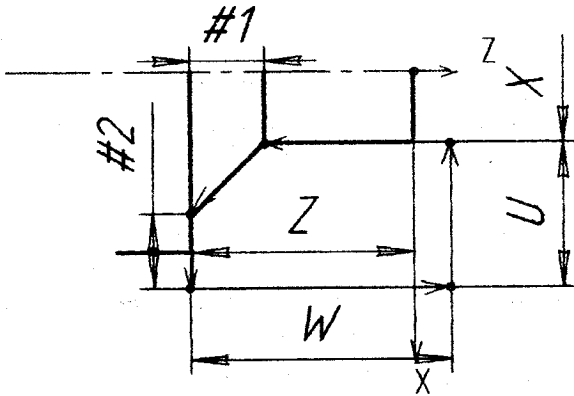


Рисунок 7.6 – Структура рухів циклу G70

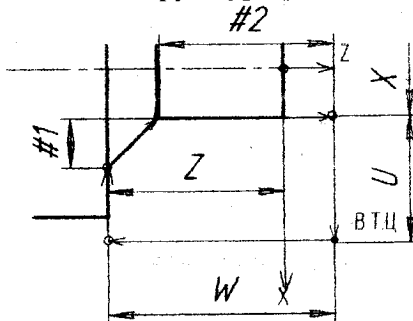


Рисунок 7.7 – Структура рухів циклу G71

**7.14.6 Цикл глибокого свердління**

Цикл глибокого свердління G73 призначений для програмування свердління отвору, що збігається з віссю Z. Цикл має таку структуру кадру:

Розділ 7 Програмування обробки деталей на верстатах із системою керування WLAT

G73 X...(U...)Z...(W...)F...#1=...#2=...,

де X.....(U...)–координата X або збільшення по цій осі, куди зміщується свердло після виконання циклу;

Z.....(W...)–координата кінця свердління отвору або величина ходу свердла у збільшеннях до цієї точки;

F... - величина робочої подачі;

#1=... - величина одного проходу, завжди позитивна і задається в збільшеннях;

#2=... - зазор безпеки при введенні свердла в отвір.

Задання робочої подачі не є обов'язковим, якщо раніше задана подача придатна і при виконанні циклу.

Якщо в програмі даний цикл повторюється, то при повторних заданнях його параметри #1, #2 можна не задавати.

#### 7.14.7 Багатопрорізний цикл прорізання канавок на торцевій поверхні деталі G74

За даним циклом можна прорізати канавки на торцевій поверхні деталі, якщо крок канавок однаковий. Так само можна проточити широку канавку, якщо крок канавки, що задається в циклі, зробити рівним або менше ширини різальної кромки канавкового різця.

Структура кадру, у якому заданий цикл, має такий формат:

G74 X...(U...)Z...(W...)F...#1=...,

де X.....(U...)–діаметр, на якому розміщений останній прохід при проточуванні канавки або зсув до нього;

Z.....(W.....) – координата дна канавки або збільшення до неї;

F – робоча подача;

#1= - крок канавки, задається в збільшеннях на діаметр, завжди позитивний.

Тут також задання подачі (F) і кроку канавки (#1) не є обов'язковими.

Структура рухів показана на рисунку 7.8.

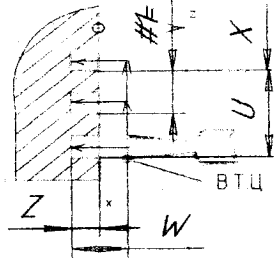


Рисунок 7.8 – Структура рухів циклу G74

### 7.14.8 Багатопрхідний цикл прорізування циліндричних канавок

Багатопрхідний цикл G75 забезпечує прорізування канавок на циліндричній поверхні з рівним кроком або дає можливість прорізати широку канавку вузьким різцем, якщо крок канавки задати рівним або менше, ніж ширина канавкового різця.

Формат циклу G75 X..(U..) Z..(W..) F...#1–... Призначення параметрів циклу абсолютно аналогічне попередньому, відмінність полягає лише в тому, що тут припуск знімається рухом по осі X і крок канавки також береться по X. Вихідним положенням інструменту перед заданням циклу також є координата першої канавки або першого проходу, якщо прорізаємо широку канавку.

### 7.14.9 Багатопрхідний цикл чорнової обробки циліндричної поверхні

Цикл G77 характерний тим, що припуск знімається при переміщенні різця на робочій подачі уздовж осі Z. За цим циклом можна обробляти зовнішню поверхню і розточити отвір.

Відмінність цього циклу в порівнянні з аналогічним, розглянутим нами в розділі 2, полягає в тому, що до складу параметрів цього циклу входить і подача для останнього чистового проходу циклу. А в системі «Електроніка НЦ – 31» вона повинна бути задана до початку циклу.

Структура циклу G77

N.... G77 X...(U...)Z...(W...)F...#1=...#2=...#3=...,

$Z...(W...)$ —координата кінцевої точки обробки або збільшення до неї від вихідної точки циклу;

$F$  – робоча подача для всіх чорнових проходів щодо знімання припуску;

$\#1=$  - глибина різання задається у збільшеннях на діаметр і завжди є позитивною величиною;

$\#2=$  - величина скосу по осі  $Z$  задається у збільшеннях і є тільки позитивною;

$\#3=$  - подача для останнього чистового проходу циклу, що виконується по контуру деталі.

Останній чистовий прохід машина виконує тоді, коли залишився припуск на поверхні деталі меншим або таким, що дорівнює заданій за параметром  $\#1=$  глибині різання.

Якщо в програмі обробки деталі даний цикл використовується повторно і його параметри ( $\#1=, \#2=, \#3=$ ) не змінюються, то їх при повторному заданні циклу можна не задавати. Схема рухів циклу показана на рисунку 7.9.

#### 7.14.10 Багатопробіжний цикл чорнового підрізування торця

Даний цикл за своїм складом і призначенням подібний до розглянутого вище. Відмінність полягає тільки в напрямку знімання припуску, тут він знімається рухом різця по осі  $X$ .

Структура циклу G78

$N...G78...X...(U... )Z...(W... )F... \#1=... \#2=... \#3=.....$

Призначення параметрів і структура рухів показані на рисунку 7.10.

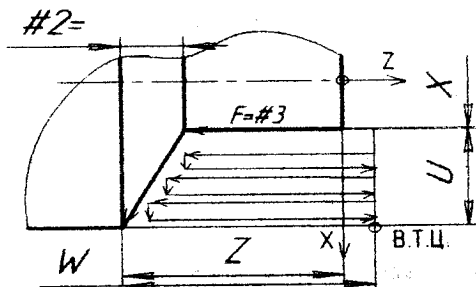


Рисунок 7.9 – Структура рухів циклу G77

У циклах G77 і G78, якщо скіс відсутній, його параметр (#2= ) у циклі не задається.

Якщо в одній програмі даний цикл задається кілька разів і його параметри залишаються незмінними, то їх повторно задавати не слід. Тоді задається тільки команда циклу й адреси X і Z з їх розмірами.

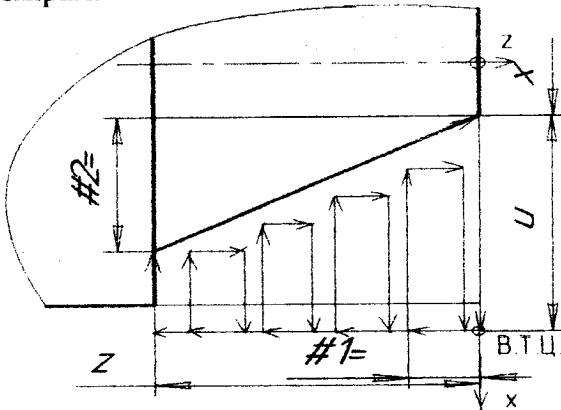


Рисунок 7.10 – Структури рухів циклу G78

### 7.15 Методика програмування контурної обробки

Програмування обробки контуру кінцевою фрезою припускає введення величини радіуса фрези і корекцію його з урахуванням фактичного розміру інструменту. При цьому в програмі ми описуємо координати опорних точок заданого контуру, а машина веде автоматичне перерахування координат еквідистантного контуру.

При програмуванні обробки контуру використовуються службові функції: G41- корекція радіуса фрези, фреза ліворуч від оброблюваного контуру; G42 – корекція радіуса фрези, фреза праворуч від оброблюваного контуру; G40 –відміна функцій корекції G41 і G42.

Дані функції машиною запам'ятовуються і діють до настання функції, що їх відмінняє. При цьому їх можна задавати окремим кадром або в тому кадрі, де запрограмований рух. Однак у всіх випадках їх можна задавати тільки тоді, коли діють



Розділ 7 Програмування обробки деталей на верстатах із системою керування WLAT  
 функції G0 або G1. У середині програми можлива зміна функції G41 на G42 і навпаки.

Якщо задано функцію корекції радіуса і контур деталі не гладкий, то кадри без перемішень задавати не рекомендується, тому що відбудеться збій корекції. Контур вважається гладким, якщо виконується умова  $d \leq \Delta L$ . Величина  $\Delta L$  вводиться оператором з пульта машини перед початком роботи з програми. Якщо контур гладкий, то еквідистанта проходить через точки 1', 2', 3' (рисунок 7.11 а). Якщо величина  $d > \Delta L$ , система формує на еквідистанті додаткову дугу 2' - 2'' радіусом, який дорівнює радіусу фрези (рисунок 7.11 б).

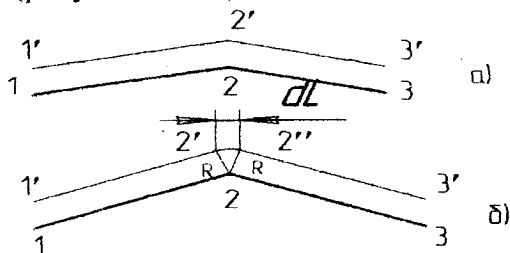


Рисунок 7.11

Приклади застосування функцій G41, G42 показані на рисунку 7.12

При зламах контуру деталі корекцію треба застосовувати залежно від того, з якого боку контуру відбувається обробка – зсередини чи ззовні. Приклади корекції наведені на рисунку 7.13.

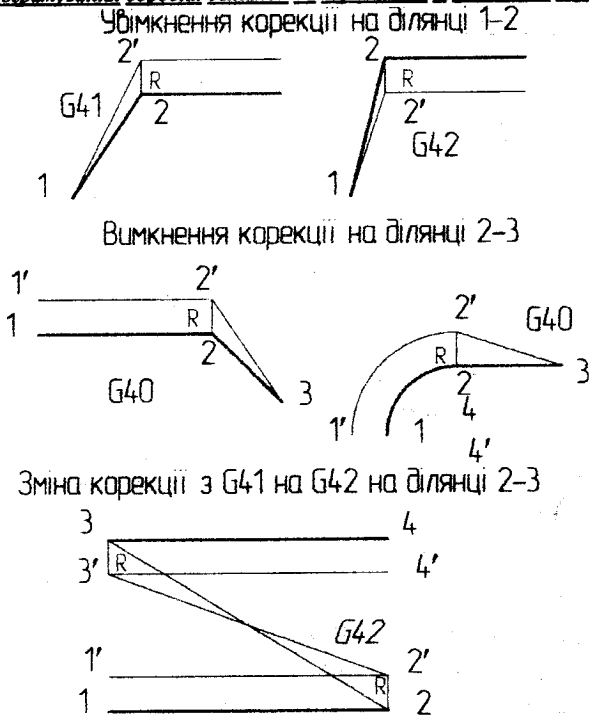
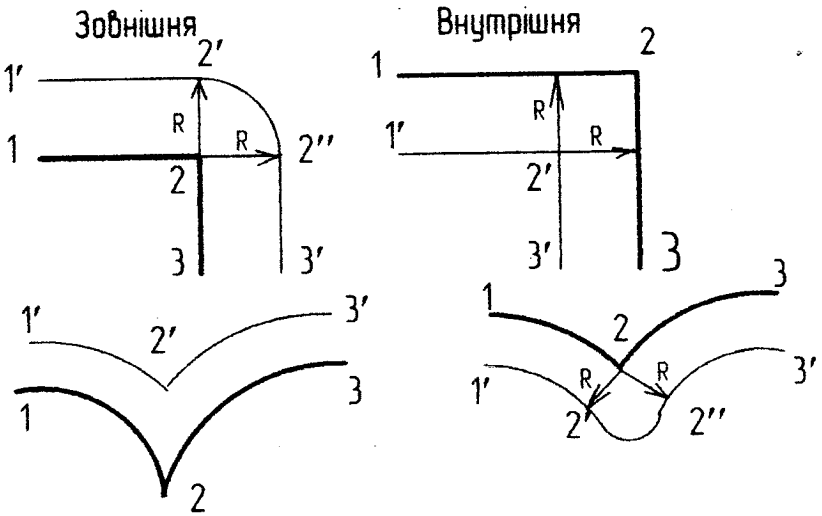
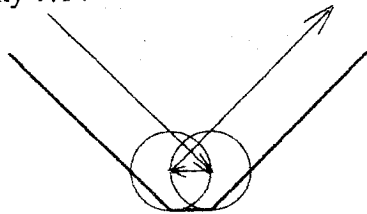


Рисунок 7.12 – Приклади використання корекції радіуса фрези при підході до контуру деталі і відведенні від контуру, а також перехід від внутрішньої обробки до зовнішньої



*Рисунок 7.13 – Корекція при зламах траєкторії*

Оскільки система керування для обчислення точки еквідистантного контуру завжди використовує дані наступного за відпрацьованим кадром, то можуть з'явитися помилки розрахунку. Це призведе до появи браку через вріз контуру, як показано на рисунку 7.14



*Рисунок 7.14 – Вріз внутрішнього контуру*

Якщо внутрішній контур у куті описаний радіусом, меншим за радіус фрези, то може виникнути односторонній вріз контуру, як показано на рисунку 7.15.

У цьому випадку інструмент зупиниться в точці 2', відпрацьовування програми припиниться і на екрані індукуються слова: «Негативний радіус при корекції».

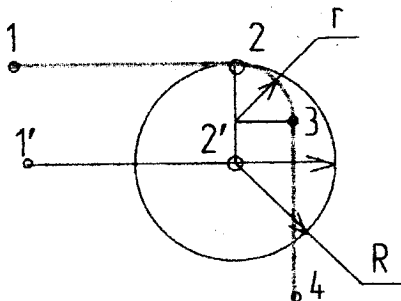


Рисунок 7.15

### 7.16 Зміщення нульової точки деталі

Зміщення нульової точки деталі здійснюється за командами G54-G59, а скасування зміщення – командою G53. Зміщення нульової точки діє тільки у випадку активних функцій G0 або G1. Зміщення може задаватися із командами переміщення виконавчого механізму або без цих команд. Зміщення задається з пульта оператора окремо за кожною з координат. Він може бути позитивним або негативним.

### 7.17 Постійні цикли

Постійні цикли задаються службовими функціями G81-G89. Скасування дії тієї або іншої функції циклу виробляється командою G80. Алгоритм постійного циклу містить у собі такі дії механізмів верстата:

- 1) швидкий вихід інструменту в координату отвору;
- 2) швидкий підхід до площини обробки;
- 3) рух з робочою подачею до координати Z;
- 4) зупинка на дні отвору;
- 5) повернення інструменту на швидкому ході або робочій подачі в точку початку руху або іншу.

Характеристики постійних циклів наведені в таблиці 7.3.

Таблиця 7.3 – Характеристики постійних циклів

Найменування циклу	Вид руху у циклі	Функції на дні отвору		Повернення
		витримка часу	функція шпинделя	
G81 свердління	робоча подача	немає	робоча швидкість	швидкий хід
G82 розточування	робоча подача	є	робоча швидкість	швидкий хід
G83 глибоке свердління з руйнуванням стружки	переривчаста робоча подача	немає	робоча швидкість	швидкий хід
G84 нарізування різі мітчиком	робоча подача	немає	реверс	робоча подача
G85 розсвердлювання	робоча подача	немає	робоча швидкість	робоча подача
G86 розгортання	робоча подача	немає	останов	швидкий хід
G89 розгортання	робоча подача	є	робоча швидкість	робоча подача

### Приклад керуючої програми для вертикально – фрезерного верстата

Верстат одноінструментний, тому для зміни інструменту передбачені зупинки програми за функцією M00. Ескіз деталі і траєкторія руху інструментів показані на рисунку 7.16.

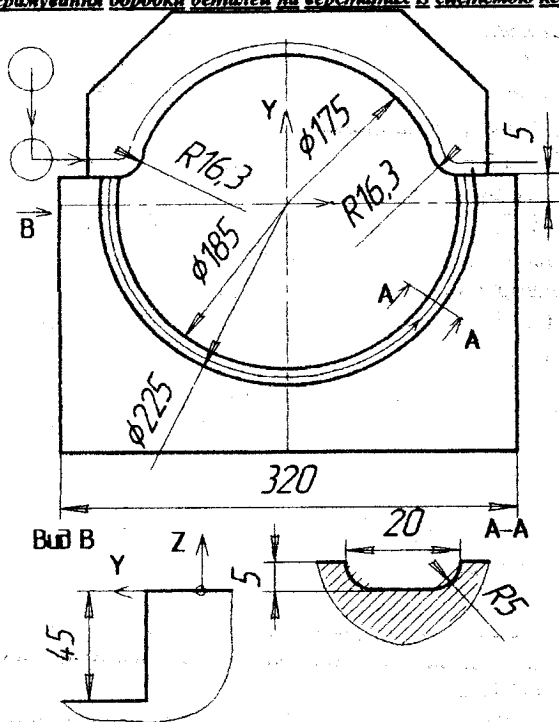


Рисунок 7.16

Керуюча програма:

% F 2715A - знак відсоток – це початок програми і далі пояснювальний текст латинськими літерами. Перше - це модель верстата  
 \$ ZAKAZ 21063 – номер замовлення  
 \$ H05.157.117.01 – номер деталі  
 \$ KORPYS – найменування деталі  
 \$USTANOWKA 1 – номер установки  
 \$FREZA D32 (H01, D02) – зазначений інструмент і його коректори  
 \$ kontur d175, r16 – зазначено, що буде обробляти обраний інструмент – контур деталі діаметром 175мм і радіуси 16мм  
 N1 G00 G17 G90 G54 X - 180 Y 25 – на швидкому ході інструмент виходить у точку з координатами X= - 180, Y= 25, одночасно зазначається, що ця точка знаходиться відносно початку координат деталі (G54) і що працювати будемо в абсолютній системі координат (G90), усі дуги лежать у площині XY (G17)

Розділ 7 Програмування обробки деталей на верстатах із системою керування WL4T

- N2 G43 H01 Z 60 – вводиться позитивна корекція на довжину інструменту і зазначається початкова точка щодо нуля деталі
- N3 M03 – праве обертання шпинделя
- N4 G01 G09 Z – 45 F 1400 M07 – фреза опускається на глибину уступу, контур якого буде оброблятися
- N5 G41 D02 Y 5 F252 – переміщаємося по осі Y до поверхні обробки і задаємо положення інструменту щодо початкової точки оброблюваного контуру – фреза ліворуч (G41), коректор на діаметр інструменту 02
- N6 X – 101.591 F 52 – обробляємо лінійну частину контуру
- N7G03 X – 85.638 Y 17.955 I 0 J 16.3 – обробляємо контур по дузі радіусом 16,3мм проти годинникової стрілки
- N8 G02 X 85.638 Y 17.955 I 85.638 J – 17.955 – обробляємо контур по дузі діаметром 175мм
- N9 G03 X 101.591 Y 5.115 I 15.953 J 3.345 – обробляємо другу ділянку контуру по дузі радіусом 16,3мм
- N10 G01 X 180 – обробляємо прямолінійну частину контуру з іншого боку деталі
- N11 G40 Y 25. F300 – скасовуємо корекцію на радіус інструменту (G40). Ця команда повинна задаватися з яким-небудь рухом, тому переміщаємося по осі Y в точку B=25мм
- N12 G00 Z 50. M05 – прискорено піднімаємо фрезу і зупиняємо шпиндель
- N13 G49 Z0- скасовуємо корекцію на довжину інструменту
- N14 M09 – вимикаємо охолодження (було ввімкнено в кадрі №4)
- N15 G00 X 300 Y 0 – відводимо інструмент у зону ручної зміни
- N16 M00 – безумовний останов
- \$ FREZA D20 S R5 – установлюємо фрезу діаметром 20мм зі скругленими вершинами зуба з радіусом 5мм
- \$ po centru kanawki – зробимо прохід по центру канавки шириною 20мм
- N17 G00 G17 G90 G54 X – 102.5 Y 20 – повторюються команди перших трьох кадрів
- N18 G00 G43 H03 Z 50
- N19 M03
- N20 G01 G09 Z – 5 F 1363 M07 – опускаємо фрезу на глибину канавки і вмикаємо охолодження
- N21 Y0. F40 – по осі B підходимо до початкової канавки. Оскільки рух відбувається по центру канавки, то корекцію на радіус інструменту не включаємо

Розділ 7 Програмування обробки деталей на верстатах із системою керування W14T

N22 G03 X 102.5 Y 0. I 102.5 J 0. – обробляємо канавку по центральній дузі

N23 G01 Y 20- виходимо з канавки по осі Y

N24 G00 X 50 M05 – зупиняємо шпиндель і відходимо по осі X

N25 G49 Z0 –відміняємо корекцію на довжину інструменту

N26 M09

N27 G00 X 300 Y 0 – відводимо інструмент у зону зміни

N28 M00

\$ FREZA D20 S R5 (H03, D04) – встановлюємо аналогічну фрезу, але тепер вона пройде по контуру внутрішньої дуги канавки діаметром 185мм

\$ kontur po d 185

N29 G00 G17 G90 G54 X 102.5 Y 20. – повторюємо дії щодо прив'язування інструменту до нульової точки деталі і введення корекції на довжину фрези

N30 G00 G43 H03 Z 50

N31 M03

N32 G01 G09 Z – 5 F1405 M07

N33 G41 D04 X 92.5 F245 – задаємо переміщення в напрямку контуру обробки і показуємо, де знаходиться фреза щодо першої точки контуру обробки (ліворуч - G41)

N34 Y0 F65 – прорізаємо пряму частину канавки

N35 G02 X – 92.5 Y 0 I – 92.5 J 0– фрезеруємо дугову частину канавки

N36 G01 Y 20- фрезеруємо вихід з канавки по прямій

N37 G40 X – 102.5 – скасовуємо корекцію на радіус фрези

N38 G00 Z 50 M05

N39 G49 Z 0 – скасовуємо корекцію на довжину фрези

N40 M09

N41 G00 X 300 Y 0 – повертаємося в зону зміни інструменту

N42 M30 – кінець програми

%% - дані знаки обов'язкові



**Розділ 8****Параметричне програмування для системи WL4T**

Методика параметричного програмування для даної машини аналогічна розглянутій нами в розділі 6. Тому тут будуть докладно розглянуті тільки ті відмінності, що виникли в даній системі керування.

**8.1 Види пам'яті машини**

Для збереження проміжних результатів обчислення, неявного задання значень за керуючими адресами і передачі параметрів у цикли і підпрограми в ході виконання програм використовуються комірки формальної пам'яті з номерами 1 – 99. Комірки з номерами від 100 і вище закріплені за системною пам'яттю. Вміст цих комірок можна тільки прочитати, але не можна в них заносити результати розрахунків.

Комірки формальної пам'яті з 1-ї по 29-ту є локальними. Тут зберігаються копії змінних викликаних програм або підпрограм. Модифікація змінних у викликаній програмі не робить впливу на вміст цих комірок пам'яті. Це дозволяє програмісту не піклуватися про відновлення параметрів при повторному виклику програм або підпрограм.

Комірки з 30-ї по 99-ту є універсальними і загальними для всіх рівнів вкладення.

Звертання до комірки формальної і системної пам'яті однакове і складається із символу # і номера комірки. Наприклад, #2 означає звертання до комірки №2, а запис ##2 означає звертання до комірки пам'яті, номер якої зберігається в комірці №2. Або запис, де # ABS (#3 - #4 \*2) означає, що треба взяти абсолютне значення, що одержується при розв'язанні вираження, поміщеного в дужках. (Із вмісту комірки №2 відняти вміст комірки №3 і різницю помножити на 2).

Системна пам'ять зберігає в собі такі команди і параметри:

- # 101 – поточне значення функцій (G0 – G3, G32 < G33);
- # 102 – поточне значення функцій (G4- - G42);
- # 103 – поточне значення функцій (G53 – G59, G92);
- # 104 – поточне значення функцій (G90, G91);

- # 105 – поточне значення функцій (G94 , G95);
- # 106 – поточне значення функцій (G96, G97);
- # 107 – поточне значення функцій (G80 – G89);
- # 108 – поточне значення функцій (G31, G70 – G78);
- # 112 – поточне значення функцій (M3 - M5);
- # 114 – поточне значення функцій (M7 – M9);
- # 117 – поточне значення функцій (M41 – M48);
- # 120 – останнє задане значення S;
- # 121 – останнє задане значення T;
- # 130 – розрахункове значення X в абсолютних координатах;
- # 131 – збільшення по X в попередньому кадрі;
- # 132 – розрахункове значення Y в абсолютних координатах;
- # 133 – збільшення по Y в попередньому кадрі;
- # 134 – розрахункове значення Z в абсолютних координатах;
- # 135 – збільшення по Z у попередньому кадрі;
- # 136 – останнє задане I;
- # 137 – останнє задане J;
- # 138 – останнє задане K;
- # 139 – останнє задане F;
- # 140 – останнє задане R;
- # 141 – останнє задане D;
- # 200 – установлений по команді M6 інструмент.

Це не всі функції, але ті, що найбільш часто трапляються і їх місце збереження.

## **8.2 Вирази**

Комбінація знаків операцій і операндів, яка подана в круглих дужках, її результатом є реальне значення, що називається виразом. Знаки операцій визначають дії, що повинні бути виконані над операндами. Кожен операнд у виразі, у свою чергу, може бути виразом, що являє собою реальне значення. Значення виразу залежить від розміщення знаків операцій, а також від пріоритетів операцій.

Операнд – це реальне число, змінна або вираз.

Операції поділяються на унарні і бінарні.

### 8.2.1 Унарні операції

Унарні операції складаються з операнда, який записаний в круглі дужки, і команди операції.

AB (операнд) – абсолютне значення;

COS (операнд) – косинус;

ACOS (операнд) – арккосинус операнда від -1 до 1;

SIN (операнд) – синус;

ASIN (операнд) – арксинус операнда від -1 до 1;

TAN (операнд) – тангенс;

ATAN (операнд 1) / (операнд 2) – арктангенс  $\frac{\text{операнд}1}{\text{операнд}2}$ ;

EXP (операнд) – експонента  $e^{\text{операнд}}$ ;

FIX (операнд) – округлення до меншого цілого;

FUP (операнд) – округлення до більшого цілого;

LN (операнд) – натуральний логарифм;

ROUND (операнд) – округлення до найближчого цілого;

SQTR (операнд) – корінь квадратний.

Операнди тригонометричних функцій беруться тільки в градусах і десяткових частках градуса.

### 8.2.2 Бінарні операції

Бінарні операції складаються з двох операндів, розділених знаком бінарної операції. Бінарні операції бувають такі:

\*\* - піднесення до степеня;

\* - множення;

/ - ділення;

MOD – залишок від ділення  $f = x \text{MOD} y$ , де  $x = ae + f$

для цілого  $a$  і  $0 < f < y$ ;

+ - додавання;

- - віднімання;

= - присвоєння;

AND – логічне І;

OR – логічне АБО;

XOR – логічне ЩО ВИКЛЮЧАЄ АБО;

== - логічне ДОРІВНЮЄ;

< > - логічне НЕ ДОРІВНЮЄ;

< - логічне МЕНШЕ;

< = - логічне НЕ БІЛЬШЕ;

> - логічне БІЛЬШЕ;

> = - логічне НЕ БІЛЬШЕ.

Результат логічних операцій завжди має значення 0 або 1.

### 8.2.3 Порядок обчислення виразів

Арифметичні операції у виразі виконуються зліва направо з урахуванням пріоритету операції. У першу чергу виконуються множення, ділення, піднесення до степеня і одержання залишку від ділення. Якщо операнд у своєму складі містить який-небудь вираз, то спочатку обчислюється цей вираз.

### 8.3 Керування ходом виконання програми

Для керування ходом виконання програм і підпрограм використовуються умовні і безумовні переходи.

#### 8.3.1 Команда безумовного переходу

Ця команда виконується шляхом запису в кадрі адреси N із цифрою, що визначає номер кадру, до якого треба перейти для подальшого виконання програми. Щоб була можливість перейти до якого-небудь кадру програми, він повинен бути заданий за визначеним правилом.

Це правило полягає в тому, що номер такого кадру починається з нуля.

Наприклад, N032 ...- до цього кадру можна перейти, а N32... – не можна.

У кожній програмі або підпрограмі повинне бути не більш 127 кадрів – позначок, до яких можна переходити при виконанні програми. Команда переходу не може бути в кадрі першою, тому що першим йде номер кадру.

Приклад:

N5 G1 X10 Z30 F 50

N010 Z – 10.5

N15 Z0 N17 – після переміщення в точку Z0 відбувається перехід

до кадру №17

N16 - кадр буде пропущений

N20 U5 N10 – після відпрацювання збільшення U5 буде перехід до кадру №10

N25 M2

У даному прикладі кадр № 25 ніколи не буде виконуватися, тому що задано нескінченний цикл від кадру № 10 до кадру № 20.

### 8.3.2 Команда умовного переходу

Команда умовного переходу застосовується у тому випадку, якщо виконується який-небудь логічний вираз. У цьому випадку в кадрі записується слово IF – якщо.

Наприклад:

N10 IF (#1 > #2) #1-0.5 N1 - якщо вміст комірки № 1 більше числа, записаного в комірці № 2, то із вмісту комірки № 1 відняти 0.5 і перейти в кадр № 1;

IF (#30 < > 0) #30-1 – якщо вміст комірки № 30 не дорівнює нулю, то це число треба зменшити на 1;

N20 G1 X0 Z100 F500 IF (#1 - #2) > 0) Z0 – якщо різниця вмісту комірок № 1 і № 2 не дорівнює нулю, то виконується лінійна інтерполяція в точку X0 Z0, інакше – у точку X0 Z100.

**Розділ 9****Методика підготовки керуючих програм для верстатів із системою керування серії NC – 110 (200)****9.1 Загальна характеристика системи керування**

Система керування належить до класу оперативних. Призначена для керування будь-якими типами металорізальних верстатів аж до багатоцільових. Система має 8 координат керування при лінійній інтерполяції і дві – при круговій. Точність інтерполяції в межах одного мікрона на один метр радіуса. Керуюча програма може бути введена в пам'ять машини вручну з пульта оператора або з периферійного пристрою будь-якого типу. Інформація про переміщення може бути задана як в абсолютних координатах, так і в збільшеннях. Розміри програмуються без яких-небудь перетворень так, як вони задані на кресленні. Десяткова частина числа відокремлюється від цілої - крапкою.

Наприклад: X – 20.532. Контроль за введенням програми і її відпрацюванням здійснюється на екрані монітора. Геометрична інформація і робоча подача можуть задаватися в міліметрах або дюймах. Система має функції контролю за справністю інструменту і його терміном служби, а також автоматичного контролю розмірів обробки за допомогою електронного щупа.

**9.2 Характеристики програмування****9.2.1 Система вимірювання**

Задаючи функцію G71, ми всі розміри задаємо в міліметрах. При заданні функції G70 розміри задаються в дюймах. Функція G90 визначає задання розмірів в абсолютній системі координат, а функція G91 – у збільшеннях. Якщо задати функцію G79, то всі переміщення задаються щодо нуля верстата, а не деталі.

**9.2.2 Програмування головного руху**

Головний рух задається за адресою S або числом обертів шпинделя за хвилину (функція G97), або швидкістю різання, м/хв (функція G96).

### 9.2.3 Програмування подачі

Прискорені переміщення виконавчих механізмів верстата програмуються функцією G0, що діє тільки в межах цього кадру.

Робоча подача програмується за адресою F, а її розмірність визначається функціями G93, G94, G95. При заданні функції G94 подача задається в мм/хв або дюйм/хв. Функція G95 визначає розмірність подачі мм/об або дюйм/об. Задаючи функцію G93, ми маємо можливість задати так званий «зворотний час» (відношення швидкості подачі до величини шляху). Функція F у кадрі з G93 діє тільки в межах одного кадру, де вона задана, і машиною не запам'ятовується на відміну від G94 і G95.

### 9.2.4 Програмування зміни інструменту

Зміну інструменту програмуємо заданням за адресою T номера інструменту і номера його коректора. При цьому цифри перед десятковою крапкою – це номер інструменту, а після крапки – його коректор. Наприклад:

T 5.25. У кадрі зміни інструменту потрібно обов'язково задати функцію M6.

### 9.2.5 Службові функції та їх призначення

Службові функції задаються за адресою G і мають таке призначення:

G00 – прискорене переміщення;

G01 – лінійна інтерполяція;

G02 – кругова інтерполяція за годинниковою стрілкою;

G03 – кругова інтерполяція проти годинникової стрілки;

G04 – витримка часу в секундах;

G09 – уповільнення наприкінці кадру;

G17 – площина XY при круговій інтерполяції;

G18 – площина XZ при круговій інтерполяції;

G19 – площина YZ при круговій інтерполяції;

G20 – закриває середовище програмування мовою GTL;

G21 – відкриває середовище програмування мовою GTL;

- G27 - обробка контуру за програмою з автоматичним уповільненням при обході кутів контуру деталі;
- G28 – те саме, але без затримки на кутах контуру;
- G29 – позиціонування «від точки до точки»;
- G33 – цикл нарізування різі різцем з постійним або змінним кроком;
- G40 – скасування введення радіуса фрези при автоматичному розрахунку еквідистантного контуру по заданому контуру деталі;
- G41 – введення радіуса фрези при автоматичному розрахунку еквідистантного контуру, фреза ліворуч від контуру деталі;
- G42 – те саме, але фреза праворуч;
- G70 – програмування в дюймах;
- G71 – програмування в міліметрах;
- G72 – вимірювання положення точки контуру з компенсацією радіуса фрези;
- G73 – вимірювання розмірів отвору;
- G74 – вимірювання відхилення від заданої точки контуру без компенсації радіуса фрези;
- G79 – програмування щодо нуля верстата (діє тільки в цьому кадрі);
- G80 – скасування постійних циклів;
- G81 – цикл свердління;
- G82 – цикл розточування;
- G83 – цикл глибокого свердління;
- G84 – цикл нарізування різі мітчиком;
- G85 – цикл розсвердлювання;
- G86 – цикл розгортання;
- G89 – цикл розгортання із зупиненням;
- G90 – програмування в абсолютних координатах;
- G91 – програмування в збільшеннях;
- G93 – величина подачі, задана як функція часу обробки;
- G94 – розмірність робочої подачі, мм/хв або дюйм/хв;
- G95 – розмірність робочої подачі, мм/об або дюйм/об;
- G96 – умова сталості швидкості різання, розмірність м/хв або дюйм/хв;
- G97 – завдання частоти обертання шпинделя, об/хв;

### 9.2.6 Допоміжні функції

Допоміжні функції кодуються за адресою M і мають таке призначення:



- M00 – безумовна зупинка відпрацьовування програми. Ця команда зупиняє виконання програми після виконання всієї інформації, записаної в кадрі програми. При цьому шпindel зупиняється, ЗОР відключається, але вся подальша програма в пам'яті зберігається;
- M01 – зупинка програми з підтвердженням. Щоб ця команда спрацювала аналогічно до M00, треба перед пуском програми з пульта оператора ввести код USO=1. Якщо код USO=0, машина команду пропускає без уваги;
- M02 – кінець програми без повернення в її початок;
- M03 – обертання шпинделя за годинниковою стрілкою;
- M04 – обертання шпинделя проти годинникової стрілки;
- M05 – стоп шпindel. Команда виконується після всіх операцій, заданих у кадрі програми. При цьому подача ЗОР відключається;
- M06 – команда на зміну інструменту. За цією командою відбувається зупинка обертання шпинделя, відключається подача ЗОР і припиняється відпрацьовування програми. Не стирає команди M03, M04, M08, M13, M14;
- M07 – увімкнення допоміжного охолодження;
- M08 – увімкнення основного охолодження;
- M09 – вимкнення охолодження. Команда виконується після повного відпрацьовування інформації кадру;
- M10 – блокування лінійних і кругових осей координат. За допомогою цієї функції здійснюється блокування осей, що не беруть участі у процесі обробки;
- M11 – скасовує M10;
- M12 – блокування кругових осей координат;
- M13 – обертання шпинделя за годинниковою стрілкою і увімкнення охолодження;
- M14 – обертання шпинделя проти годинникової стрілки і увімкнення охолодження;
- M19 – зупинка обертання шпинделя з кутовою орієнтацією. Команда виконується після відпрацьовування всієї інформації, заданої в кадрі. При цьому стираються в пам'яті команди M03, M04, M13, M14;
- M30 – кінець програми і повернення до першого кадру;
- M40 – скасування діапазону обертання шпинделя;
- M41 –
- M42 –
- M43 – } вибір діапазону чисел обертів шпинделя;
- M44 – }
- M45 – автоматична заміна діапазону чисел обертів шпинделя;
- M60 – заміна деталі.

### 9.2.7 Початок і кінець програми

У першому кадрі звичайно програмується команда про заміну інструменту (T... M06). Наприкінці обробки треба зупинити усі виконавчі механізми верстата в такому положенні, щоб було зручно зняти деталь і установити заготовку. Потім треба зупинити обертання шпинделя і подачу ЗОР. І далі задати автоматичне закінчення програми і повернення на початок функцією M30. У тексті програми можна вставляти будь-які повідомлення, задаючи код DIS і текст повідомлення в лапках. Текст повідомлення повинен бути не більше 32 символів.

N1 { DIS, "....." }

N2 T1.1 M6 S800

N3 G00 Z80 X80 M13

.....  
.....  
N236 G00 Z250 X50 M5

N237 M30

### 9.2.8 Деякі особливості застосування адрес

Адреса **R** використовується в стандартних циклах. За ним задається координата точки початку руху на робочій подачі або величина збільшення до цієї точки. Крім того, ця адреса задіяна при нарізуванні багатозахідної різі як зрушення початку синхронізації на величину ходу різьблення при нарізуванні чергового витка.

Адреси **I** і **J** використовуються при круговій інтерполяції незалежно від площини обробки і за ними задаються відстані від початкової точки дуги до її центра. При цьому

**I** - завжди абсциса, а **J** - ордината. Крім того, адреса **I** використовується в постійному циклі **G33** - нарізування різі з кроком, що змінюється (для задання величини зміни кроку). Якщо в циклі задані **I+** і розмір, то крок буде поступово збільшуватися на величину, задану за цією адресою. Якщо **I** з мінусом, то крок буде зменшуватися.

Адреса **K** в циклі **G33** - нарізування різі й у циклі **G34** - нарізування різі мітчиком - означає крок різі. А в циклі **G83** -

постійний цикл глибокого свердління - визначає коефіцієнт множення для обробки глибини отвору.

### 9.3 Види рухів виконавчих механізмів

Вид руху виконавчого механізму визначається завданням функцій **G**.

**G00** – прискорене переміщення по осях, записаних у кадрі програми. Діє тільки в межах цього кадру.

**G01** – лінійна інтерполяція. Команда діє до настання альтернативної функції.

**G02** – кругова інтерполяція за годинниковою стрілкою.

**G03** – кругова інтерполяція проти годинникової стрілки. Обидві функції машиною запам'ятовуються і діють до настання альтернативної функції.

**G33** – нарізування різі різцем з постійним або змінним кроком.

#### 9.3.1 Програмування прискореного переміщення

Прискорене переміщення визначає лінійний тип руху виконавчих механізмів зі швидкістю, скоординованою по всіх осях, записаних у кадрі.

Структура такого кадру:

**N...G00**[інші функції **G**] [координати точки виходу інструмента або збільшення] [операнди корегування] [швидкість подачі] [допоміжні функції],

де [інші функції] – усі функції, які необхідно задати і які сумісні з **G00**;

[координати точки або збільшення по осях] – числове значення в явній або неявній формі;

[операнди корегування] – коефіцієнти корекції на площині (u,v,w);

[швидкість подачі] – робоча подача. Вона машиною запам'ятовується, але в цьому кадрі не діє;

[допоміжні функції] – функції **M**, **S**, **T**. В одному кадрі можна задати до чотирьох функцій **M** та по одній **S**.

Приклад: N23 G00 G90 G17 G71 G94 X80 Z45 Y120 F75 S450  
T5.25 M06 M3 M42 LF.

### 9.3.2 Лінійна інтерполяція

Лінійна інтерполяція задається функцією G01 і визначає прямолінійний рух по всіх осях, записаних у кадрі зі швидкостями, скоординованими таким чином, щоб результуюча швидкість руху була задана робочій подачі.

Структура кадру:

N...G01[інші функції G] [координати або збільшення по осях] [операнди корегування] [швидкість подачі], де [швидкість подачі] – робоча подача по результуючій прямій. Якщо в кадрі подача не задана рух буде відбуватися на подачі, заданій в раніше записаних кадрах програми.

Інші складові кадру описані вище. Приклад лінійної інтерполяції наведений на рисунку 9.1.

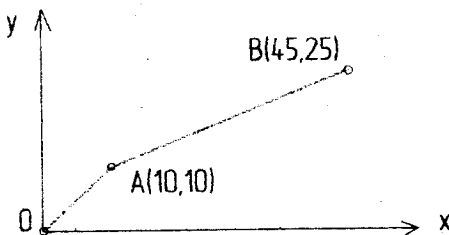


Рисунок 9.1 – Приклад лінійної інтерполяції

Фрагмент програми руху по прямих O-A-B:

N65 G01 G90 X10 Y10 F50 LF

N66 X45 Y25 LF.

### 9.3.3 Кругова інтерполяція

Кругова інтерполяція програмується функціями G02 або G03 залежно від напрямку обходу дуги. Максимально можна запрограмувати дугу  $360^{\circ}$ . Дуги менше  $360^{\circ}$  можна запрограмувати двома способами. За першим способом задаються координати центра дуги, а за другим – радіус дуги.

Формат команди кругової інтерполяції першим способом:

{G02, G03} {ІНШІ G} {ОСІ} I J {ШВИДКІСТЬ ПОДАЧІ} {ОПЕРАНДИ КОРЕГУВАННЯ} {ДОПОМІЖНІ ФУНКЦІЇ}, де {ОСІ} – представлені адресою і величиною координати або збільшенням по даній осі. Якщо в кадрі жодна вісь не задана, отже, запрограмована дуга розміром  $360^{\circ}$ . Цифрове значення за адресою може бути задано і у неявній формі;

I і J – у даній системі керування означають координати центра дуги, у якій би площині вона не лежала. При цьому I завжди належить до першої осі, а J – до другої. При параметричному програмуванні вони можуть бути задані в явній і неявній формах. Якщо програмування здійснюється за збільшеннями (G91), то за цими адресами задається відстань від початкової точки дуги окружності до її центра.

Приклад кругової інтерполяції наведений на рисунку 9.2.

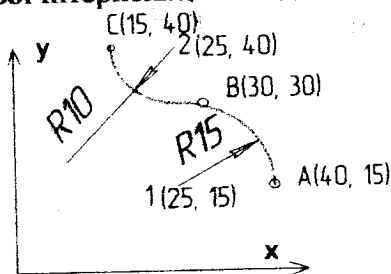


Рисунок 9.2 – Приклад кругової інтерполяції

Фрагмент програми до рисунка 9.2 (запрограмоване переміщення по дугах ABC):

**A) варіант програми в абсолютних координатах:**

N10 G03 G90 G17 X 30 Y 30 I 25 J 15 F 200 LF

N11 G02 X15 Y 40 I 25 J 40 LF

**B) варіант програми в збільшеннях:**

N10 G03 G17 G91 X- 10 Y 15 I- 15 J 0LF

N11 G02 X- 15 Y 10 I-5 J 10 LF

Напрямок кругової інтерполяції (G02 або G03) визначається відповідно до рисунка 9.3. При цьому треба дивитися на площину з боку позитивної осі, перпендикулярної до цієї площини.

За другим способом дугу менше  $360^{\circ}$  у даній системі можна запрограмувати через координати кінцевої точки дуги і її радіус. Формат такого кадру буде:

{G02, G03} {ІНШІ G} {ОСІ} R± {ШВИДКІСТЬ ПОДАЧІ};  
{ОПЕРАНДИ КОРЕГУВАННЯ} {ДОПОМІЖНІ ФУНКЦІЇ};

Програмування дуги менше  $360^{\circ}$  за другим способом покажемо на прикладі рисунка 9.2:

N10 G90 G17 G03 X 30 Y 30 R + 20 F200 LF

N11 G02 X 15 Y 10 R + 10 LF

Знак радіуса визначається за таким правилом: якщо дуга менш  $179,999^{\circ}$ , то перед значенням радіуса ставиться знак плюс (+), якщо дуга має значення від  $180^{\circ}$  до  $359,999^{\circ}$ , то знак мінус (-).

### 9.3.4 Гвинтова інтерполяція

Щоб одержати гвинтовий рух інструменту щодо оброблюваної поверхні, треба в одному кадрі запрограмувати одночасно кругову інтерполяцію на площині і лінійну інтерполяцію уздовж осі, перпендикулярної до цієї площини. При цьому саме задання лінійної інтерполяції (G01) у кадрі відсутнє, а ознакою її є крок спіралі, заданий за адресою K, і довжина гвинта, задана за адресою Z.

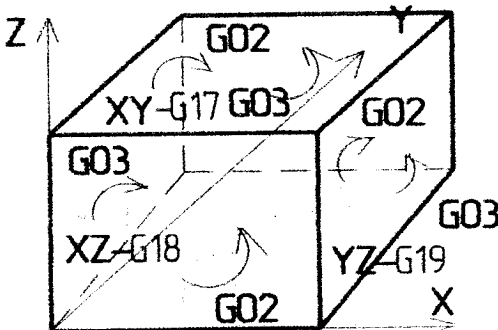


Рисунок 9.3 – Визначення напрямку обходу дуги при круговій інтерполяції

Якщо довжина гвинта менше кроку, то крок гвинта можна не задавати. Але між довжиною гвинта і його кроком повинне виконуватися співвідношення

$$NC = 110 (200)$$

$$Z = n \cdot K,$$

де  $n$  - ціле або десяткове число.

Якщо  $n$  - ціле число, то повинна бути задана повна окружність, тобто  $360^\circ$ . Якщо це число дробове, то дуга повинна бути запрограмована пропорційно до дробової частини числа  $n$ .

Наприклад:  $Z=2,7K$ .

Повинна бути запрограмована дуга -  $360 \cdot 0,7 = 252^\circ$ .

Функції G02 і G03 визначають напрямок спіралі - правий або лівий. Для прикладу розглянемо ліву спіраль із кроком 5мм, що показана на рисунку 9.4.

### 9.3.5 Нарізування різі різцем

Нарізування різі різцем можна віднести до ще одного виду інтерполяції. При цьому можна програмувати нарізування циліндричної, конічної різі, з постійним і змінним кроком, а також багатозахідні різі. Нарізування різі програмується функцією G33.

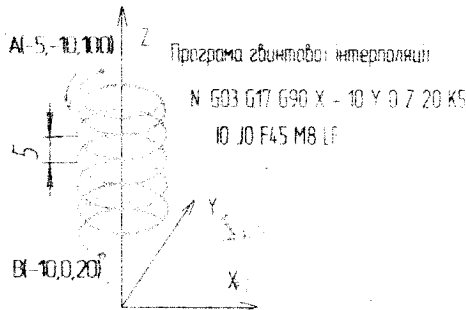


Рисунок 9.4 - Приклад гвинтової інтерполяції

Структура кадру програми має вигляд

$N...G33 \{ \text{РОЗМІРИ РІЗИ ПО ОСЯХ} \} K... I... R...LF,$

де ДО - крок різі, якщо різь із змінним кроком, то це початковий крок. Ця величина не може бути опущена.

I - величина зміни кроку; якщо різь з кроком, що збільшується, то ця величина позитивна, при зменшуваному кроці - негативна;

R - адреса, що задається при нарізуванні багатозахідних різей. За ним задається відхилення стосовно кутової позиції

нуля шпинделя (у градусах). У цьому випадку вихідна точка нарізного різця залишається постійною, а змінюється кут синхронізації початку нарізного руху. Наприклад, при нарізуванні двозахідної різі буде заданий R 180, для тризахідної – R 120, потім R 240.

Приклад програмування тризахідної різі з кроком 6мм:

N35 G33 Z – 45 K6 LF – перший прохід;

N45 G33 Z – 45 K6 R120 LF – другий прохід;

N55 G33 Z – 45 K6 R 240 LF – третій прохід.

Усі параметри різьблення можуть бути задані в явній і неявній формах.

Під час дії функції G33 корекції швидкості подачі й обертів шпинделя, а також команда «стоп» не діють.

Приклади нарізування різі різцем з постійним кроком за функцією **G33** показані на рисунку 9.5, із змінним кроком – на рисунку 9.6.

Примітки:

- 1 Вісь U діаметральна.
- 2 Усі параметри різі можуть бути задані у цифровому вигляді й у неявній формі.
- 3 Під час нарізування різі із зменшуваним кроком треба стежити за тим, щоб наприкінці різі крок не дорівнював нулеві. Перевірку роблять за формулою

$$I \leq \frac{2K}{L},$$

де I – максимальна зміна кроку різі;

K – початковий крок різі;

L – довжина різі.

### 9.3.6 Програмування обертання деталі

Ряд сучасних широкоуніверсальних верстатів мають поворотні столи. Програмування обертального руху здійснюється шляхом задання відповідної кругової осі і швидкості руху по цій осі, у градусах за хвилину, яка задається



за адресою F. Наприклад, у кадрі заданий F75.5. Поворот стола буде відбуватися зі швидкістю 75,5 град/хв. Швидкість обертання стола при відомій подачі фрезерування визначиться за формулою

$$F = \frac{360}{\pi} \cdot \frac{A}{D} = 144.64 \cdot \frac{A}{D},$$

де F- кутова швидкість, град/хв;

A – лінійна швидкість подачі, мм/хв;

D – діаметр, на якому виконується фрезерування.

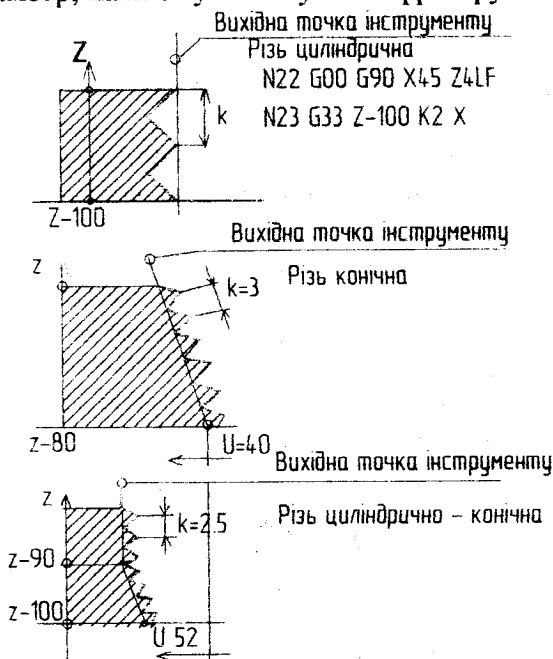


Рисунок 9.5 Приклади нарізування різі з постійним кроком

Якщо необхідний поворот стола на швидкому ході, то задається функція G00. Поворот стола на робочій подачі здійснюється функцією G01.

У тих випадках, коли разом з обертанням стола відбувається переміщення по лінійних осях, швидкість подачі варто виконувати за формулою

NC = 110 (200)

$$F = A \cdot \frac{\sqrt{X^2 + Y^2 + Z^2 + B^2 + C^2}}{L}$$

де  $F$  – результуюча швидкість подачі, мм/хв;

$A$  – швидкість подачі, що потрібна для обробки деталі.

$X, Y, Z, B, C$  – величини переміщень по осях;

$L$  – сумарна довжина шляху інструменту.

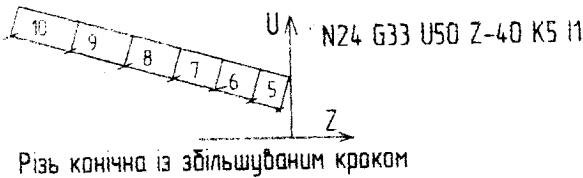
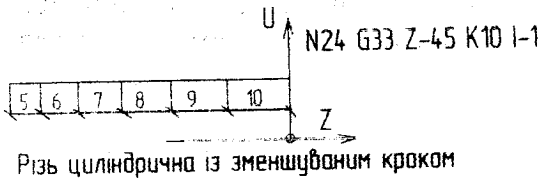
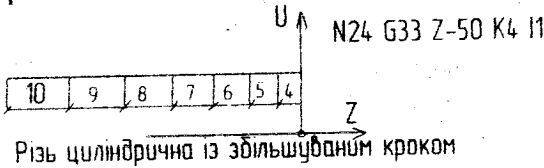


Рисунок 9.6 – Приклади програмування нарізування різи із змінним кроком

У цьому випадку можуть бути кілька видів траєкторій. Якщо рух відбувається тільки по круговій осі, то буде дуга окружності. Якщо відбувається одночасне обертання і переміщення по лінійних осях, то може бути спіраль Архімеда, циліндрична спіраль або складна крива.

### 9.4 Безеквідистантне програмування обробки контуру

Для безеквідистантного програмування обробки контуру використовуються функції:

G41 – фреза ліворуч від контуру деталі;

G42 – фреза праворуч від контуру деталі;

G40 – відміна функцій G41 і G42.

Перш ніж задати положення фрези щодо оброблюваного контуру, необхідно обов'язково визначити площину інтерполяції функціями G17, G18, G19. При цьому функція G17 встановлюється автоматично при ввімкненні машини.

Інструмент до початкової точки контуру повинен підходити тільки по прямій лінії. А початкова ділянка контуру може бути і лінійною, і круговою. Протягом обробки усього контуру деталі вектор радіуса фрези повинен залишатися перпендикулярним до оброблюваного контуру. При цьому забезпечується точне позиціонування інструменту в точках перетинання прямих і окружностей, змішених щодо профілю обробки (рисунок 9.7).

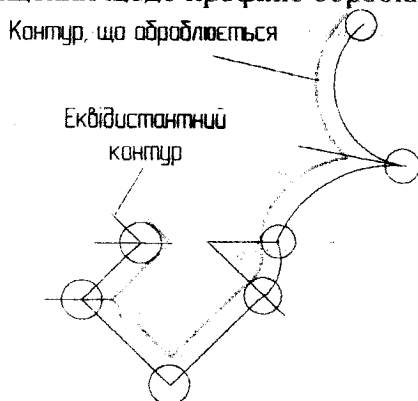


Рисунок 9.7 – Положення інструмента в точках з'єднання контуру

На рисунках 9.8 і 9.9 показані різні варіанти початкової ділянки профілю деталі і положення фрези залежно від заданої функції компенсації радіуса фрези. Для відміни компенсації радіуса фрези в останньому кадрі програми, що описує профіль деталі, задається функція G40. На рисунках 9.10 і 9.11 показано,

як діє функція, що відмінняє, при різних варіантах кінцевої ділянки профілю деталі.

Компенсація радіуса фрези діє й у кадрах з G00.

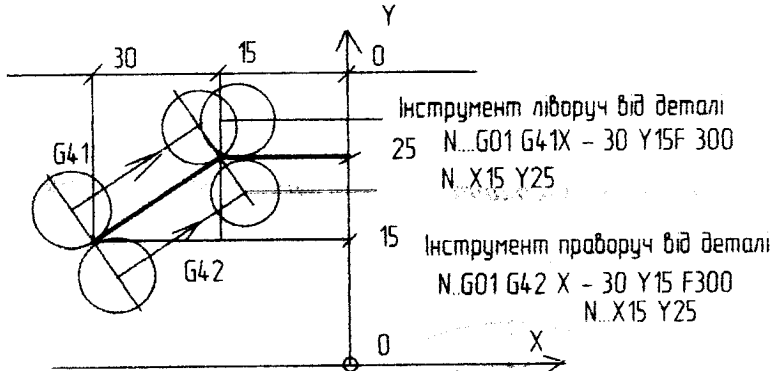


Рисунок 9.8 – Перший елемент профілю – лінійний

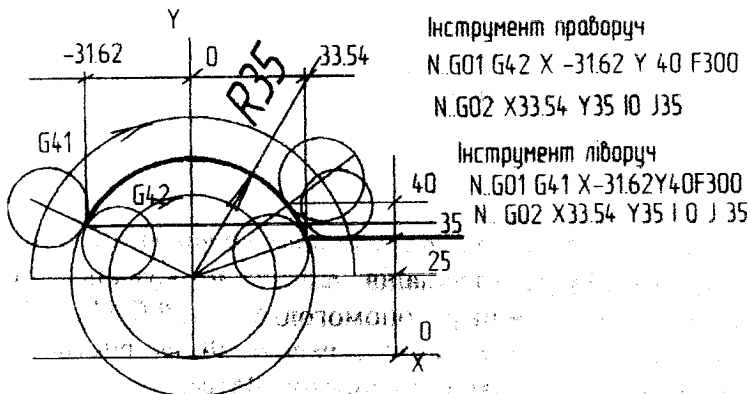


Рисунок 9.9 – Перший елемент профілю – круговий

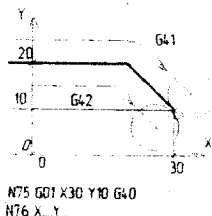


Рисунок 9.10 - Останній елемент профілю - лінійний

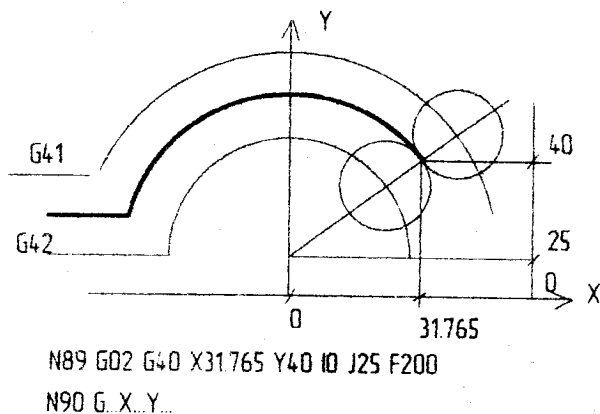


Рисунок 9.11 - Останній елемент профілю - дуга

Розглянемо кілька прикладів щодо застосування методу компенсації радіуса фрези за допомогою функцій G41 і G42.

Приклад 1 В деталі «корпус», наведеній на рисунку 9.12, необхідно вифрезувати два пази висотою 45мм.

Керуюча програма:

(DIS, "KORPUS 1.1650.49.150.52")

(DIS, "FREZA KONCEWAJA D50")

(UAO, 1)

(UCG, 2, X-200 X200, Y-200 Y200)

G17

T1.1 M06

N1 G00 G90 X-65 Y-162

NC - 110 (200)

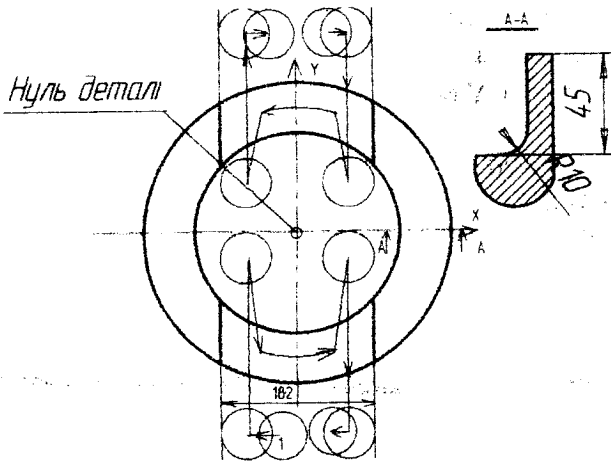


Рисунок 9.12

N2 Z100

N3 M03

N4 Z-54 M08

N5 G01 G42 X-91 Y-162 F100

X - 91 Y - 80 F 40

N6 G40 X - 91 Y - 66 F 40

N7 X - 57.231 Y - 82.78 F 200

N8 X - 40 Y - 114.197 F 40

N9 G03 X 40 Y - 114.197 I 0 J 0

N10 G01 G42 X 91 Y - 66

N11 X 91 Y - 113

N12 G40 X 91 Y - 162 F 300

N13 X 65 F 100

N14 G00 Z 100

N15 Y 162

N16 Z - 45

N17 G01 G42 X 91 Y 162 F 100

N18 X 91 Y 80 F 40

G40 X 91 Y 66 F 40

N19 X 57.231 Y 82.78 F 200

N20 X 40 Y 114.197 F 40  
 N21 G03 X - 40 Y 114.197 I 0 J 0  
 N22 G01 G42 X - 91 Y 66  
 N23 Y 113  
 N24 G40 Y 162 F 300  
 N25 X - 65 F 100  
 N26 G00 Z 100  
 48 M05  
 N49 Z 150 M09  
 N50 M02

Як другий приклад розглянемо обробку паза, показаного на рисунку 9.13

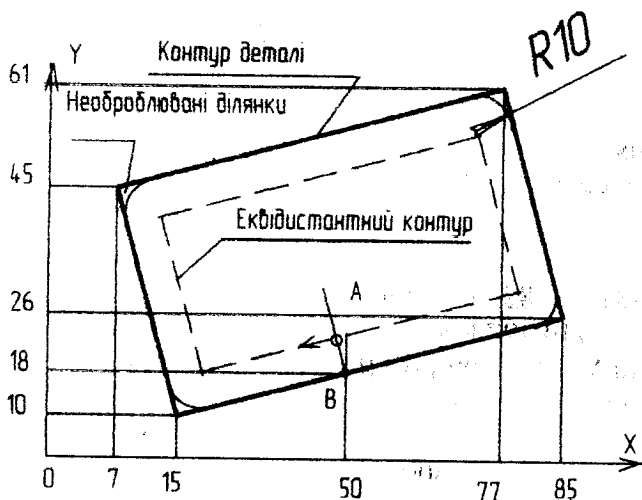


Рисунок 9.13 – Приклад обробки паза з використанням функцій G42, G40

Керуюча програма:

(DIS, "FREZA KONCEWAJA D20")

N10 S110 F170 T9.6 M06

N20 G00 X 50 Y 32 M03

N30 Z - 305

N40 G01 G42 X50 Y18

N50 X15 Y10

N60 X7 Y45

N70 X77 Y61

N80 X85 Y26

N90 G40 X50 Y18

N100 G00 X50 Y32

N110 G00 Z305 M05

N120 M30

### 9.5 Програмування з використанням стандартних циклів

Стандартні цикли програмуються за допомогою функцій G81-G89 і дозволяють виконувати ряд операцій без повтору кожного разу їх розмірів.

Структура кадру стандартного циклу має такий вигляд:  
G8X [ІНШІ G] [R1, R2] КООРДИНАТА ЦИКЛУ [ДОДАТКОВІ ОПЕРАНДИ] [ШВИДКІСТЬ ПОДАЧІ] [ДОПОМІЖНІ ФУНКЦІЇ],

де - [ІНШІ G] – це службові функції, які можна програмувати в кадрі стандартного циклу;

[R1, R2] – перше значення це координата точки, у яку інструмент підводиться на швидкому ході для початку робочого руху в циклі; друге значення - це координата точки, у яку інструмент виходить з отвору на швидкому ході після виконання циклу. Якщо другий параметр не заданий, інструмент зупиниться в точці R1;

КООРДИНАТА ЦИКЛУ – це кінцева точка руху інструменту на робочій подачі при виконанні циклу;

[ШВИДКІСТЬ ПОДАЧІ] – задається за адресою F і визначає величину робочої подачі в циклі в прийнятій розмірності;

[ДОДАТКОВІ ОПЕРАНДИ] – це параметри, що задаються в деяких стандартних циклах (наприклад, I, J, K для циклу глибокого свердління);



[ДОПОМІЖНІ ФУНКЦІЇ] - це функції S,M,T,H, що визначають послідовність рухів при виконанні стандартного циклу.

Структура рухів при виконанні стандартних циклів:

- швидкий підхід у точку розташування отвору;
- швидкий підхід у точку R1;
- рух на робочій подачі до точки Z;
- функції циклу на дні отвору;
- повернення на швидкому ході в точку R1 або R2.

Характеристики стандартних циклів наведені в таблиці 9.1.

Таблиця 9.1 - Характеристики стандартних циклів

Назва циклу	Характер руху	Функція на дні отвору		Повернення
		витримка часу	реверс шпинделя	
G81 свердління	робоча подача	немає	немає	прискорений рух у R1 або R2
G82 розточування	робоча подача	немає	немає	те саме
G83 глибоке свердління	рух на робочій подачі, швидко відведення, швидко повернення	так/немає	немає	те саме
G84 нарізування різьничком	робоча подача, початок обертання	немає	с	робоча подача в R1 і прискорене переміщення в R2, якщо задано
G85 розсвердлювання	робоча подача	немає	немає	те саме
G86 розгортання	робоча подача, початок обертання	немає	зупинення	прискорене переміщення
G89 розгортання або розточування	робоча подача	с	немає	як у G84
G80 відміна циклів				

#### Примітки:

1 Якщо стандартний цикл повторюється кілька разів поспіль, то його треба задати один раз після виходу в координати першого отвору. Для наступних отворів треба задавати тільки їх координати, а цикл при виході в ці координати буде повторюватися автоматично. Після обробки останнього отвору треба задати скасування циклу функцією G80.

2 Тривалість витримки часу програмується кодом TMR.

3 Стандартні цикли не можна програмувати усередині програми з функціями G41, G42.

### 9.5.1 Програмування стандартного циклу свердління

Структура кадру N...G81 R1...(R2...)Z...

Приклад програмування циклу наведений на рисунку 9.14.

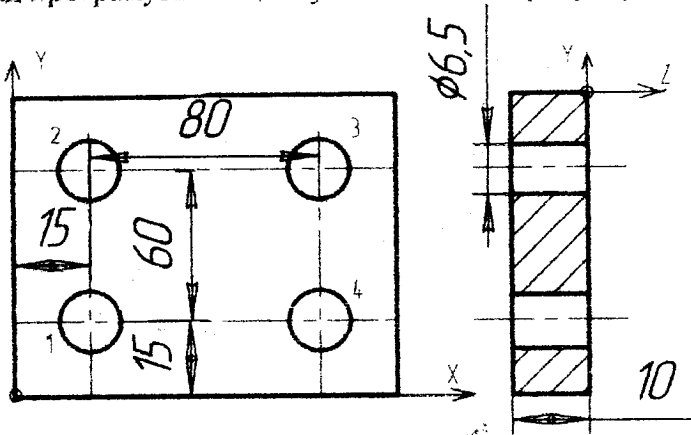


Рисунок 9.14

Фрагмент керуючої програми:

```

N.....
N25 S800 F95 T3.3 M06 G90 LF
N26 G81 R2.5 Z -15 M3 LF
N27 X15 Y15 G00 LF
N28 X60 G00 LF
N29 X80G00 LF
N30 Y15 G00 LF
N31 G80 Z50 G00 M5 LF
N32.....
  
```

Стандартний цикл G81 можна використовувати для центрування, зенкерування і розточування отворів.

Програмування інших стандартних циклів абсолютно ідентичне наведеному прикладу. За необхідності введення витримки часу вона задається попередньо через команду TMR  
Наприклад: N33 TMR=3 LF

NC-110 (200)

N34 G82 R3 Z -200 T6.6 M13 LF

N35 X35 Y290 G00 LF

N36 G80 LF

У даному прикладі витримка часу 3с.

Приклад програмування циклу з двома параметрами R показаний на рисунку 9.15.

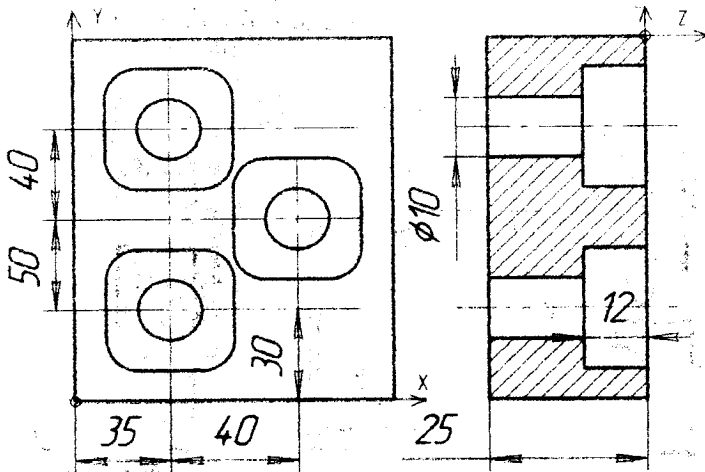


Рисунок 9.15

Фрагмент керуючої програми:

N40 S850 F100 T4.4 M6 LF

N41 G81 R -10 R2 Z -30 LF

N42 X35 Y30 G00 LF

N43 X80 Y80 G00 LF

N44 X35 Y120 G00 LF

N45 G80 Z50 G00 M5 LF

### 9.5.2 Програмування циклу глибокого свердління

Структура циклу: G83 R1.....(R2....)Z...I...(K...)(J...),

де R1, R2, Z мають те саме значення, що й у циклі G81.

I – глибина одного свердління;

J – збільшення величини I;

K – коефіцієнт зменшення параметра I (до досягнення величини J).

Наявність або відсутність у кадрі циклу цих параметрів формують два різних цикли.

Перший випадок - запрограмовані параметри I, J, K.

Цикл виконується за такою схемою:

- швидкий підхід по координатах X і Y в точку заданого отвору.
- швидкий підхід до точки R1;
- свердління на глибину R1+I;
- повернення в точку R1 на швидкому ході;
- обчислення нового значення  $R1=R1+I-1$ ;
- обчислення нового значення I;  $I=I*K$ , якщо  $I*K \geq J$ , і  $I=J$ , якщо  $I*K < J$ .

Кроки, починаючи з другого, виконуються один за одним до одержання повної глибини отвору.

Щоб глибина одного свердління була постійна, треба задати  $K=1$ , а J не задавати.

Другий випадок: K і J не задані.

Цикл виконується за такою схемою:

- швидкий підхід до центра отвору;
- швидкий підхід у точку R1;
- свердління на глибину R1+I;
- витримка часу, запрограмована за TMR;
- свердління на глибину I і т.д.

Таким чином, свердло з отвору не виймається і розвантаження стружки не виконується. Дроблення стружки відбувається за рахунок витримки, коли є обертання шпинделя і немає подачі.

Після досягнення розміру Z відбувається швидкий відвід у точку R1 або в точку R2, якщо вона необхідна і задана в кадрі.

Приклад: необхідно просвердлити в деталі товщиною 50 мм три отвори діаметром 6 мм. Координати отворів: 1) X20, Y50; 2) X40, Y50; 3) X80, Y120.

Фрагмент програми:

N35 S930 F65 T5.5 M6 LF — задання режимів різання й установка інструменту (інструмент № 5 і коректор на його розмір № 5)

N36 G83 R3 Z-.55 I20 K0.8 J6 M13 LF — задання циклу

N37 X20 Y50 LF – вихід у координати першого отвору і його свердління за циклом.

N38 X40 LF – так само обробка другого отвору

N39 X80 Y120 LF – обробка третього отвору

N40 G80 Z50 G00 M5 LF – скасування циклу і відведення свердла по осі Z

### 9.5.3 Програмування циклу нарізування різі мітчиком

Стандартний цикл нарізування різі мітчиком задається функцією G84 і може виконуватися двома способами: перший спосіб, коли шпиндель не має датчика, другий спосіб – шпиндель, на якому є датчик положення.

1) Шпиндель без датчика. Формат циклу: G84 R1...R2... Z...

Параметри в циклі мають той самий зміст, що й у розглянутих прикладах.

Однак при програмуванні цього циклу треба враховувати такі особливості:

- координата точки R1 повинна вибиратися з урахуванням довжини різі. Якщо довжина різі менше або дорівнює трьом діаметрам, то треба зупинитися на відстані, яка дорівнює п'ятьом крокам різі. А якщо довжина різі більше трьох діаметрів зупинення повинне відбуватися на відстані, яка дорівнює семи крокам різі;

- швидкість подачі за адресою F треба програмувати, обчислюючи за формулою  $F = S \cdot p \cdot 0,9$ ,

де  $S$  - частота обертання шпинделя;

$p$  - крок різі;

0,9 - коефіцієнт зменшення швидкості для збереження пружності пружинного компенсатора оправлення для мітчика.

За допомогою коду *RMS* задається в програмі або вводиться з пульта оператора, можна змінювати швидкість повернення інструменту. Ця зміна визначається у відсотках. Остаточний розмір Z повинен бути зменшений на 10% від фактичного робочого ходу мітчика.

Приклад деталі для програмування циклу наведений на рисунку 9.16.

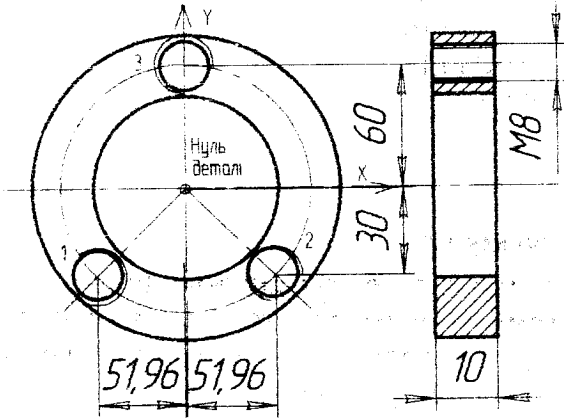


Рисунок 9.16

Фрагмент керуючої програми:

N30 S280 F315 T8.8 M6 M13 LF

N31 G84 R7 Z - 15 LF

N32 X-51.96 Y-30 LF— вихід у точку і нарізування різи в першому отворі

N33 X51.96 LF — вихід у точку і нарізування різи у другому отворі

N34 X0 Y60 LF — вихід у точку і нарізування різи у третьому отворі

N35G80 Z50 G00 M5 LF

Кадр № 31 забезпечує швидке підведення інструменту в точку R1=7, нарізування різи на довжині 15мм (нуль по осі Z знаходиться на верхній площині деталі), повернення в точку R1 на робочій подачі. Ця програма забезпечує нарізування правої різи при правому обертанні шпинделя, що задано командою M13. Якщо треба нарізати ліву різь, то треба задавати команду M14.

Випадок другий – шпиндель має датчик кутового положення. У цьому випадку є два варіанти програмування функції G84.

За першим варіантом програмують так само, як і в першому випадку, тобто програмують швидкість подачі за адресою F.

За другим варіантом в кадрі циклу подача, яка дорівнює кроку різи, задається за адресою K. Тоді формат циклу буде

G84 R1...(R2...) Z...K...

Фрагмент керуючої програми за другим варіантом для деталі, наведеної на рисунку 9.16:

N30 S280 T8.8 M6 M3 LF

N31 G84 R7 X-15 K1 LF

N32 X – 51.96 Y30 LF

N33 X 51.96 LF

N34 X0 Y60 LF

N35 G80 Z50 G00 M5 LF

### **Коментарі до стандартних циклів**

Кадри програми, починаючи з того, де заданий стандартний цикл, і закінчуючи кадром перед його відміною, входять до складу стандартного циклу. Тому зміну параметрів стандартного циклу можна вводити в будь-якому кадрі усередині цієї групи. Наприклад, свердління отворів в деталі, що лежать не в одній площині (рисунок 9.17)

Фрагмент керуючої програми:

N25 (DIS, “SWERLO D10”) LF

N26 S560 F100 T5.5 M6 LF

N27 G81 R3 Z-35 M3 LF — у циклі залишений зазор безпеки до площини першого отвору 3мм, з цим зазором обробимо і другий отвір

N28 X10 Y10 LF

N29 X30 LF

N30 Y35 R-12 LF — третій і четвертий отвори розташовані нижче площини нуля деталі на 15мм, тому, залишаючи зазор, який дорівнює 3мм, опускаємо точку R1 на 12мм

N31 X10 LF

N32 G80 Z50 G00 M5 LF

Якщо в циклі була задана точка R2, то її також можна корегувати в тому місці групи кадрів, де її потрібно змінити. Але в цьому кадрі повинна бути обов'язково задана і є точка R1, навіть якщо вона не змінюється, тому що машина розуміє ці точки тільки в одному порядку: спочатку R1, потім R2. Якщо задано тільки один розмір R, то машина його розуміє тільки як R1.

Якщо глибина свердління отворів (Z) змінюється так само, то і її можна корегувати шляхом задання нового значення в тому місці, де вона змінюється.

Приклад деталі для ілюстрації цих правил наведений на рисунку 9.18.

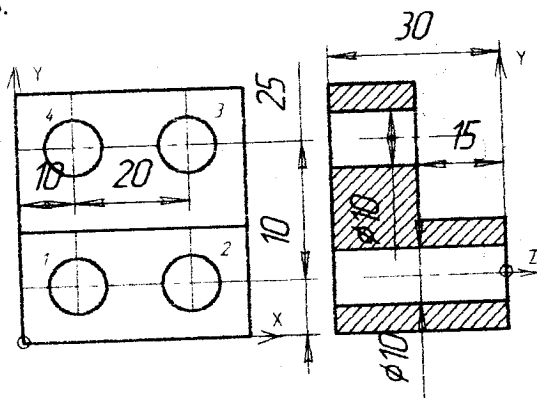


Рисунок 9.17

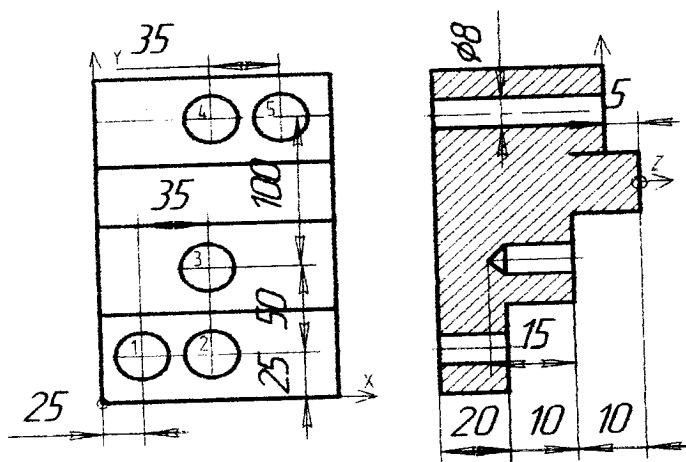


Рисунок 9.18

Фрагмент керуючої програми до рисунка 9.18:

N42 (DIS, "SWERLO D8") LF

N43 S550 F25 T4.4 M6 LF

N44 G81 R-18 Z-46 M13 LF — заданий зазор безпеки 2мм, з цим зазором свердлимо отвори 1 і 2

N45 X25 Y25 LF



**N46 X60 R-18 R-8 LF** — після просвердленого другого отвору свердло треба підняти вище, тому що третій отвір піднятий на 10мм щодо перших двох. Тому задаємо другу точку **R**

**N47 Y75 R-8 R2 Z-25 LF** — при свердлінні третього отвору знову задаємо зазор безпеки 2мм, а щоб перейти в четвертий отвір, свердел треба підняти вище нульової площини **Z** на 2мм, тому задаємо друге **R=2**. Глибина свердління третього отвору відрізняється від перших двох, тому тут задаємо цю глибину **Z**,

**N48 Y175 R-3 Z-46 LF** — четвертий і п'ятий отвори розташовані в одній площині, тому задаємо тільки один розмір **R** і змінюємо глибину свердління по **Z**

**N49 X95 LF**

**N50 G80 Z50 G00 M5 LF**

**Розділ 10****Програмування із використанням мови GTL**

Система ЧПК „НЦ – 110” (200) дозволяє описувати контур деталі не тільки стандартною мовою програмування, як описано вище, але і мовою програмування високого рівня GTL. Ця мова дозволяє описувати профілі деталі, складених з прямих ліній і дуг окружностей, використовуючи тільки інформацію креслення. Сама система обчислює точки перетину і точки торкання геометричних елементів. Мови GTL і стандартна можуть бути використані в одній програмі, але тільки не для того самого контуру.

Мова GTL може бути використана тільки при програмуванні в абсолютних координатах.

**10.1 Векторна геометрія**

Опис профілю з використанням мови GTL засновано на використанні чотирьох типів геометричних елементів:

- точка початку відліку;
- точка;
- пряма;
- окружність.

Оскільки профіль деталі визначається не тільки геометричними елементами, але і напрямком обходу контуру, то в мові GTL використовується векторна геометрія.

Наприклад: пряма знаходиться між точками А і В. Обробити цю ділянку профілю можна, рухаючись від А до В і навпаки. У цьому випадку у векторній геометрії ми маємо дві різні прямі. Тобто для опису прямої недостатньо знати її довжину, треба знати ще і напрямок обходу.

Припустимо, що напрямок обходу будемо визначати кутом, який пряма утворить з позитивним напрямком осі Х. Причому кут будемо визначати, обертаючи вісь Х, до збігу з описуваною прямою. Цей кут буде позитивним, якщо поворот осі Х відбувається проти годинникової стрілки і навпаки, негативним, якщо поворот відбувається за годинниковою стрілкою.

Розділ 10 Програмування із використанням мови GPL

Так само робимо і при описі окружностей. Якщо обхід окружності відбувається проти годинникової стрілки, то це буде позитивний рух, і навпаки.

Радіус окружності також задається позитивним при обході окружності проти годинникової стрілки.

**10.2 Збереження в пам'яті машини геометричних елементів**

Для збереження в пам'яті геометричних елементів використовуються малі літери і цифра, яка визначає порядковий номер елемента. Усі вони закріплені за визначеними видами елементів:

a – кут;

l – пряма;

c – окружність;

d – відстань;

m – модуль;

o – точка початку відліку;

r – радіус;

p – номер перетинання (дискримінатор);

s – скіс.

Елементи можуть мати цифровий індекс або параметр E. Функції G20 і G21 визначають профіль, тобто ряд геометричних елементів, з'єднаних певним чином. Профіль може бути відкритим і закритим. Відкритим вважається такий профіль, що починається з однієї точки, а закінчується іншою. Закритий профіль починається і закінчується однією і тією самою точкою.

Список можливого визначення геометричних елементів наведений у таблиці 10.1.

**10.3 Визначення точок початку відліку**

Функція визначення точок початку відліку дає можливість визначити точки початку відліку явно.

Формат команди: `on= X.Y..a..`,

де `on` – визначає назву точки відліку;

`X, Y` - координати нової початкової точки; `a` – кут обертання (позитивний, проти годинникової стрілки).

Геометрія GTL може бути визначена при будь-якій системі відліку. Формат команди і приклад показані на рисунку 10.1.

Тут  $o4$  – це назва початку точки відліку;  $X20$ ,  $У30$  – координати нової початкової точки;

$a$  – кут обертання (він позитивний, тому що поворот відбувається проти годинникової стрілки).

Таблиця 10.1 – Геометричні елементи

Елемент	Позначення	Опис
Точки початку відліку	$On=XYa$	
Точки	$pn=(om)XY$	точка в декартових координатах
	$pn=(om)ma$	точка в полярних координатах
	$pn=+lm,+lp$	точка перетинання двох прямих
	$pn=+lm,+cp,(s2)$	точка перетинання прямої і окружності
	$pn=+cm,+lp,(s2)$	точка перетинання окружності і прямої
	$pn=+cm,+cp,(s2)$	точка перетинання двох окружностей
Прямі лінії	$ln=(om)lJr,(op)lJr$	лінія, дотична до двох окружностей
	$ln=(om)XY,(op)XY$	лінія, яка проходить через дві точки
	$ln=(om)lJr,(op)XY$	лінія, дотична до окружності і проходить через точку
	$ln=(om)XY,(op)lJr$	лінія, що проходить через точку і дотична до окружності
	$ln=(om)lJr,a$	лінія, дотична до окружності і утворює кут $a$ с абсцисою
	$ln=(om)XY,a$	лінія, що проходить через точку і утворює кут $a$ с абсцисою
	$ln=+om,+cp$	лінія, дотична до двох окружностей
	$ln=-cp,pm$	лінія, дотична до окружності і проходить через точку
	$ln=pm,+cp$	лінія, що проходить через точку і дотична до окружності
	$ln=pm,pg$	лінія, що проходить через дві точки
	$ln=+cm,a$	лінія, дотична до одної окружності і утворює кут $a$ с абсцисою
	$ln=pm,a$	лінія, що проходить через точку і утворює кут $a$ с абсцисою
	$ln=lm,d$	лінія, рівнобіжна іншій лінії, яка розміщена на відстані $d$
Окружності	$cp=(om)lJr$	окружність в декартових координатах
	$cp=(om)ma$	окружність в полярних координатах
	$cp=+lm,+lp,r$	окружність даного радіуса і дотична до двох прямих
	$cp=+lm,+cp,r$	окружність, дотична до прямої і окружності даного радіуса
	$cp=+cp,+lm,r$	окружність даного радіуса, дотична до окружності і прямої

## Продовження таблиці 10.1

Окружності	$sp = pm, +1p, r$	окружність даного радіуса, що проходить через точку і дотична до прямої
	$sp = +1p, pm, r$	окружність даного радіуса, дотична до прямої і проходить через точку
	$sp = +sp, +sp, r$	окружність даного радіуса, дотична до двох окружностей
	$sp = pm, +sp, r$	окружність даного радіуса, яка проходить через точку і дотична к другій окружності
	$sp = +sp, pm, r$	окружність даного радіуса, дотична к другій окружності і проходить через точку
	$sp = pm, pg, r$	окружності даного радіуса, яка проходить через дві точки
	$sp = pm, +1p$	окружність с центром в даной точке і дотична к прямої
	$sp = pm, pa, pr$	окружність, яка проходить через три точки
	$sp = pm, r$	окружність даного радіуса із центром в даній точці
	$sp = +sp, +d$	концентричні окружності з відомими відстанями
	$sp = pm, +sp(s2)$	окружність з центром у даній точці і дотична до іншої окружності

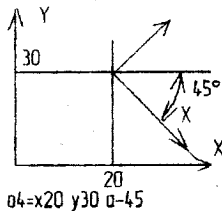


Рисунок 10.1

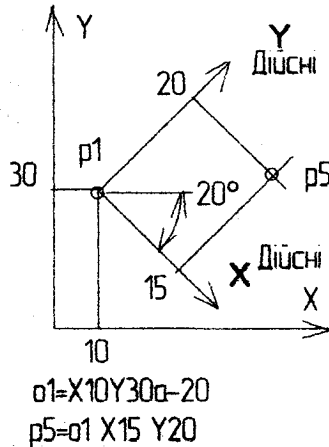


Рисунок 10.2 - Декартові координати

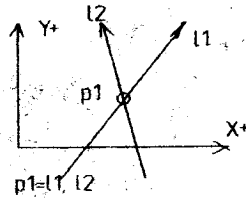
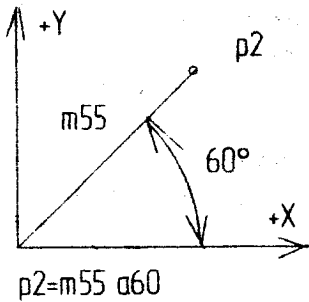


Рисунок 10.4

Рисунок 10.3 – Полярні координати

### 10.4 Визначення точок

Функція визначення точок дозволяє в явній або неявній формі визначати точки контуру деталі. Визначення точок може бути дане як у декартових, так і в полярних системах координат. Формати команд видно з прикладів, наведених на рисунках 10.2 і 10.3.

Це пряме задання точок. Крім того, має місце непряме задання точок як результат перетинання двох прямих (рисунок 10.4); прямої і окружності (рисунок 10.5) і двох окружностей (рисунок 10.6).

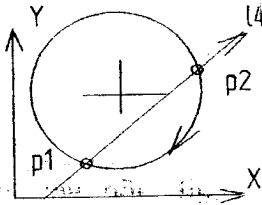
У випадку перетину прямої з окружністю і навпаки існують два розв'язки (рисунок 10.5): перший - окружність  $s1$  і пряма  $l4$  перетинаються в точках  $p1$  і  $p2$ . Проходячи пряму  $l4$ , спостерігаючи за її напрямком, спочатку натикаємося на точку  $p1$  (1-ше перетинання), а потім - точку  $p2$  (2-ге перетинання). Для вибору номера перетину треба задавати модифікатор  $sp$ . Якщо визначаємо точку  $p2$ , то записуємо в кадрі  $s2$ . Якщо модифікатор не заданий, то машина визначає точку  $p1$ .

Аналогічна ситуація виникає при перетині двох окружностей (рисунок 10.6). Тут окружності  $s1$  і  $s2$  перетинаються в точках  $p1$  і  $p2$ . Для визначення потрібної точки проводимо орієнтовну пряму, яка з'єднує центр 1-ї окружності з центром 2-ї

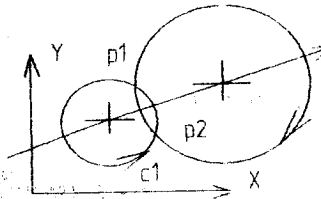
окружності. Ця пряма ділить площину на дві напівплощини. Для вибору точки  $p2$ , що лежить у правій напівплощині, треба в кадр дописати модифікатор  $s2$ . Якщо він опущений, то автоматично вибирається точка  $p1$ , що лежить у лівій напівплощині.

### 10.5 Визначення прямої лінії

Функція визначення прямої лінії дозволяє визначити її в явній і неявній формах.



$p1=l4, c3$   
 $p2=l4, c3, s2$



$p1=c1, c2$   
 $p2=c1, c2, s2$   
 $p1=c2, c1, s2$

Рисунок 10.5

Рисунок 10.6

Напрямок прямої лінії завжди спрямовується від першого до другого серед елементів, що визначаються. У випадку, якщо пряма належить окружності, можливі два розв'язки, тому що пряма може належати окружності з одного або з іншого боку. Для вибору необхідного розв'язання необхідно переконатися в тому, що в точці дотику окружність і пряма мають той самий напрямок. Якщо цього немає, розв'язок неможливий. На рисунку 10.7 показані несумісні (а) і сумісні (в) напрямки руху.

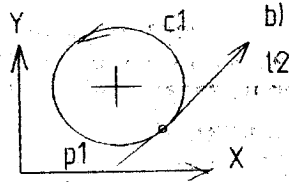
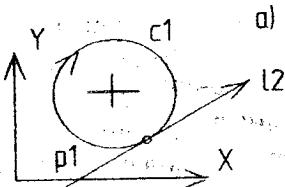


Рисунок 10.7

Задання прямих ліній у явном вигляді показано на рисунках 10.8 – 10.21

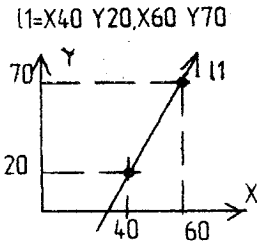


Рисунок 10.8

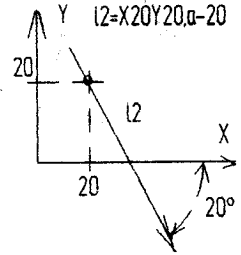


Рисунок 10.9

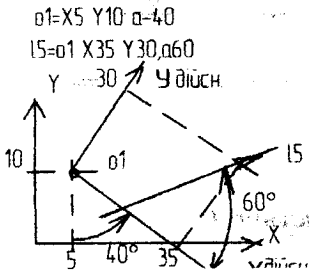


Рисунок 10.10

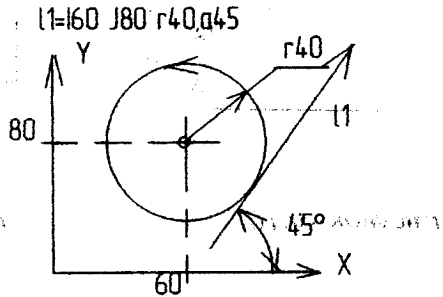


Рисунок 10.11

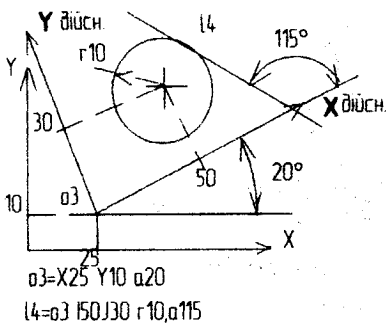


Рисунок 10.12

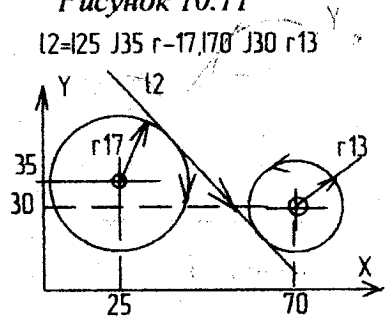


Рисунок 10.13



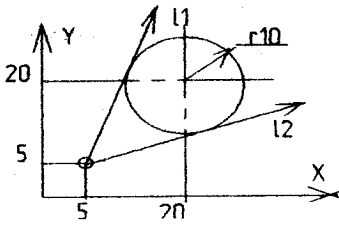
$l1=X5 Y5 l20 J20 r10$ 
 $l12=X5 Y5 l20 J20 r10$ 


Рисунок 10.14

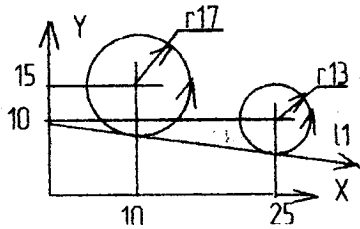
 $l1=110 J15 r17 l25 J10 r13$ 


Рисунок 10.15

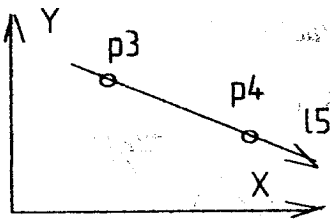
 $l5=p3, p4$ 


Рисунок 10.16

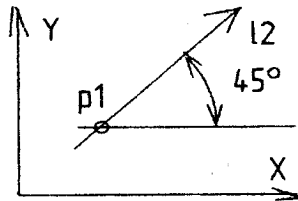
 $l2=p1, a45$ 


Рисунок 10.17

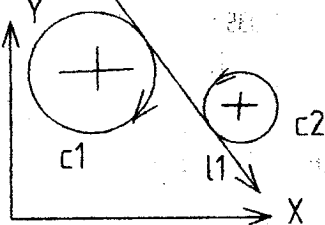
 $l1=c1, c2$ 


Рисунок 10.18

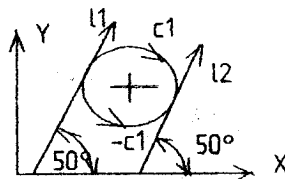
 $l2=-c1, a50$ 
 $l1=c1, a50$ 


Рисунок 10.19

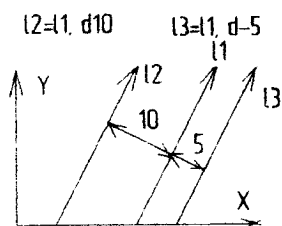


Рисунок 10.20

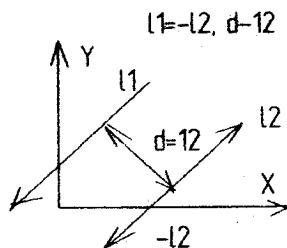


Рисунок 10.21

### 10.6 Визначення окружностей

Мова GTL дозволяє визначити окружність у прямій (явній) або в непрякій (неявній) формах.

Визначаючи окружність у непрякій формі, потрібно дуже уважно стежити за сумісністю напрямків елементів, оскільки знак мінус може змінити напрямок описуваного елемента, і він стає несумісним, тому що, рухаючись по ньому, не можна далі описувати контур деталі. Якщо не враховувати напрямок елементів, то, задаючи окружність відомого радіуса і пряму, дотичну до цієї окружності, можна одержати 8 варіантів положення окружності. А якщо враховувати напрямок елементів, то число варіантів зводиться до двох.

#### 10.6.1 Визначення окружностей у явній формі

На рисунках 10.22 – 10.24 наведені приклади визначення окружностей у явній формі.

Рисунок 10.22 і 10.23 – це визначення окружності по координатах центра і радіуса окружності, при цьому на рисунку 10.23 ми маємо повернені осі координат.

На рисунку 10.24 наведено визначення окружності в полярних координатах з відомим радіусом окружності.

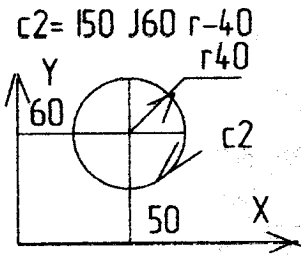


Рисунок 10.22

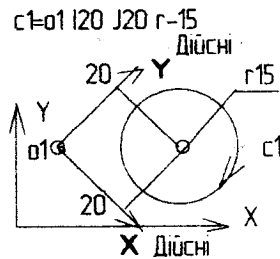


Рисунок 10.23

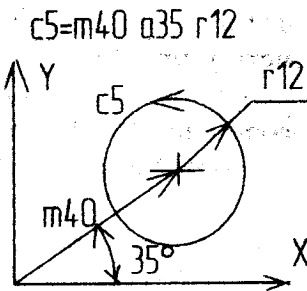


Рисунок 10.24

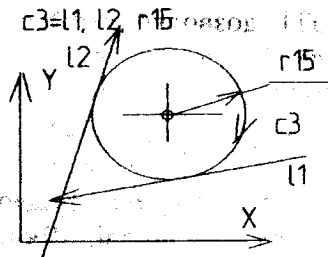


Рисунок 10.25

### 10.6.2 Визначення окружностей у неявній формі

Приклади визначення окружностей у неявній формі наведені на рисунках 10.25 – 10.38. При цьому рисунок 10.25 – окружність даного радіуса, дотична до двох відомих прямих; рисунок 10.26 – 10.28 – окружність, радіус якої відомий, і дотична до прямої й окружності, що вже описана; рисунок 10.19 – окружність даного радіуса, що проходить через точку, і дотична до відомої прямої, рисунок 10.30 – 10.31 – окружність даного радіуса, дотична до двох відомих окружностей; рисунок 10.32 – окружність даного радіуса, що проходить через відому точку, і дотична до відомої окружності; рисунок 10.33 – окружність даного радіуса, що проходить через дві відомі точки; рисунок 10.34 – окружність з центром у відомій точці і дотична до відомої прямої; рисунок 10.35 – те саме, але дотична до

відомої окружності; рисунок 10.36 – окружність, що проходить через три відомі точки; рисунок 10.37 – окружність даного радіуса із центром у відомій точці; рисунок 10.38 – окружність, концентрична до відомої окружності і віддалена від неї на дану величину.

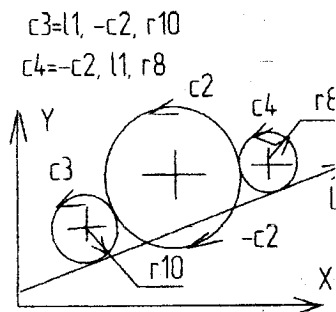


Рисунок 10.26

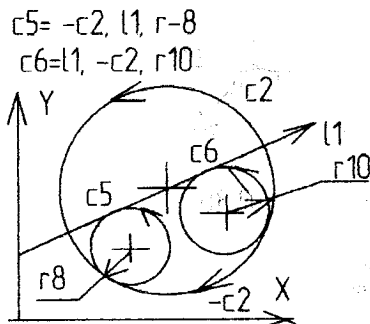


Рисунок 10.27

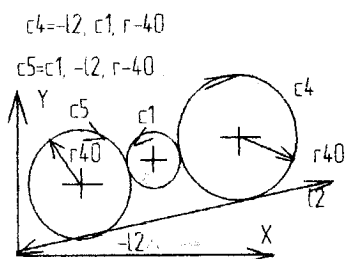


Рисунок 10.28

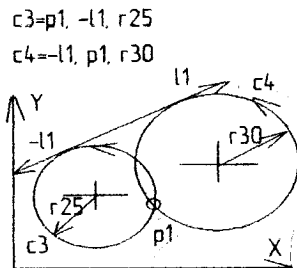


Рисунок 10.29

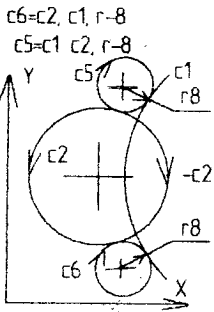


Рисунок 10.30

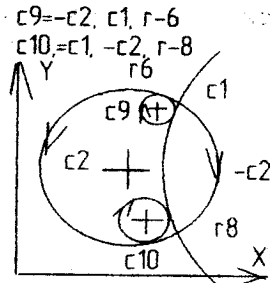


Рисунок 10.31

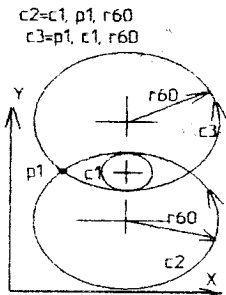


Рисунок 10.32

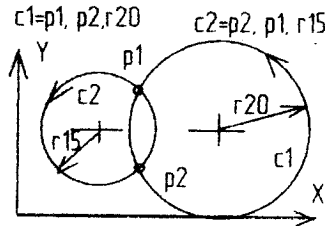


Рисунок 10.33

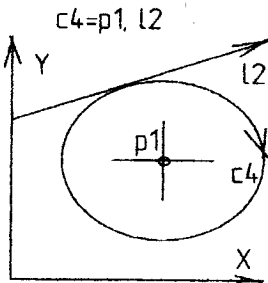


Рисунок 10.34

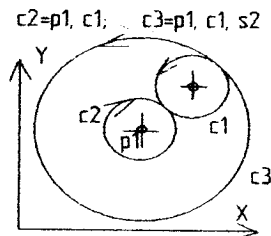


Рисунок 10.35

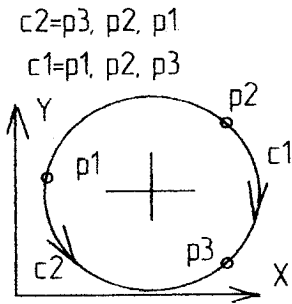


Рисунок 10.36

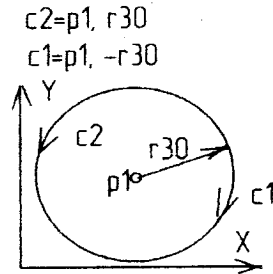


Рисунок 10.37

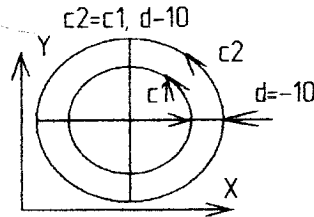


Рисунок 10.38

### 10.7 Приклад програмування мовою GTL

Контур деталі показаний на рисунку 10.39. Заготовка з листа товщиною 10 мм вирізана по шаблону з припуском по контуру 5мм.

Керуюча програма і коментарі:

N1 (DIS, "DETAL UCHEBNA")

N2 I1=X70, Y40, a150 -лінія 1 проходить через точку з координатами X70, Y40 під кутом  $150^{\circ}(90+60)$  до осі X

N3 I2=X8, Y8, a-95 — лінія 2 проходить через точку X8, Y8 під кутом  $-95^{\circ}$  до осі X; знак мінус свідчить, що поворот до збігу з позитивним напрямком осі X на зазначений кут треба робити проти годинникової стрілки, на відміну від першої лінії

N4 I1=I1, I2 — точка 1 — це точка перетину двох ліній - першої і другої

N5 I3=X8 Y8, X70 Y15 — лінія 3 проходить через точки з координатами X8, Y8 і X70, Y15

N6 I4=X50 Y0, a90 — лінія 4 проходить через точку з координатами X50, Y0 під кутом  $90^{\circ}$  до осі X. Кут позитивний, тому що до збігу з позитивним напрямком осі X пряму треба повертати за годинниковою стрілкою

## Розділ 10 Програмування із застосуванням мови GTL

N7 c1=I70 J40 r-25 — окружність 1 задана координатами центра і її радіусом  
 N8 c2=p1, r-25 — окружність 2 задана точкою центра і радіусом. Таким чином, положення всіх прямих, точок і окружностей, що описують заданий контур, визначене і можна задавати обхід контуру кінцевою фрезою №1 з коректором №1 і зазначеними режимами

N9 T1.1 M6 S800 F250 M3 M7

N10 G00 X0 Y0 — вихід у нуль деталі

N11 Z-10 — опускання фрези на 10 мм нижче нульової точки. Нагадаємо, що мовою GTL можна працювати тільки в абсолютних координатах

N12 G21 G41 I2 — вхід у програму GTL, включення корекції радіуса фрези — фреза ліворуч від оброблюваного контуру і перша лінія, перпендикулярно до якої треба підійти, це лінія 2 — остання лінія контуру

N13 I3 —

N14 r3

N15 I4

N16 r3

N17 c1

N18 r5

N19 I1

N20 r5

N21 c2 s2

N22 I2

N23 G20 G40 I3 — відміняємо програму GTL, відміняємо корекцію радіуса і зазначаємо, перпендикулярно до якої лінії скасовується корекція (лінія 3)

N24 G00 Z20 M9 } повернення в нульову точку і зупинення програми

N25 G00 X0 Y0 M30 }

далі починається перелік елементів контуру, які треба обійти, обробляючи заданий контур

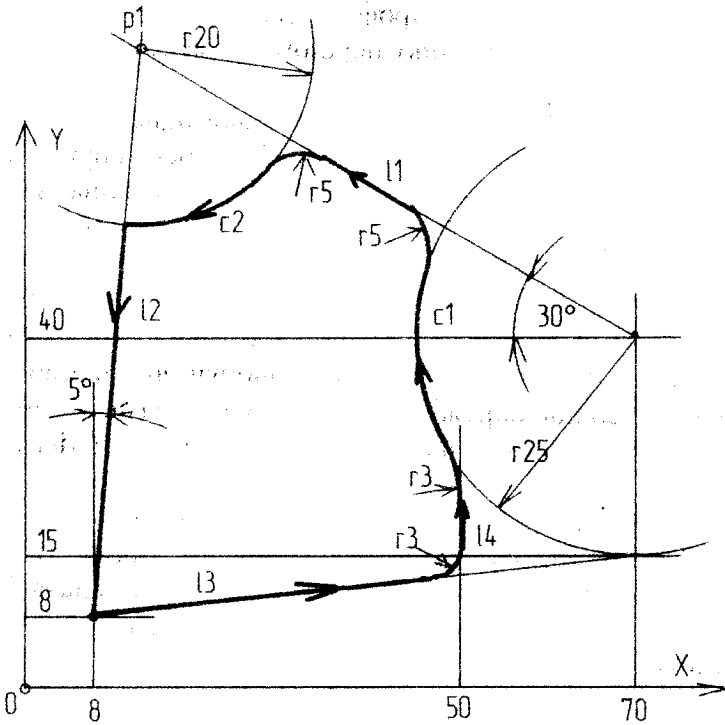


Рисунок 10.39

### 10.8 Деякі правила програмування мовою GTL

1 Якщо профіль відкритий, то він повинен починатися з точки і закінчуватися цією самою точкою, яка не збігається з першою. Компенсація радіуса інструменту, що задається функціями G41, G42, діє перпендикулярно до першого елемента контуру в початковій точці і відмінюється за функцією G40, в останній точці - перпендикулярно до останнього елемента.

2 Якщо профіль закритий, то спочатку потрібно запрограмувати останній елемент (див. кадр № 12), а потім задати перший елемент профілю. Компенсація радіуса інструмента повинна бути задана на початку профілю при



заданні останнього елемента профілю (кадр № 12) і скасована наприкінці профілю при виклику першого елемента профілю (кадр № 23).

3 У випадку перетину двох прямих є тільки одне розв'язання. Якщо перетинаються пряма – окружність або окружність – окружність, то можливі два розв'язки. Але машина завжди вибирає перше. Якщо треба вибрати друге – треба задати дискримінатор S2 (кадр № 21). При перетині прямої і окружності перше і друге перетинання визначаються напрямком прямої. При перетинанні двох окружностей першим перетинанням є те, що знаходиться ліворуч від прямої, що з'єднує центр першої окружності з центром другої. А другим перетинанням буде те, що знаходиться праворуч від тієї самої прямої.

4 Якщо елементи (прямі або окружності) перетинаються, то з'єднання між ними програмується заданням радіуса r. При цьому в кадрах з функціями G21 або G20 не можна задавати цю адресу. Якщо є гострий кут, а радіуса скруглення фактично немає, то можна при цьому задати r0.

Навчальне видання

**Сєдінкін Леонід Михайлович**

**Програмування обробки деталей на верстатах з  
оперативними системами керування**

Навчальний посібник

Редактори: С.М. Симоненко, Т.Г. Чернишова  
Комп'ютерне верстання Л.М. Сєдінкіна

Підп. до друку 16.02.2006.

Формат 60x84/16. Папір офс. Друк офс.

Ум. друк. арк. 12,32. Обл.-вид. арк. 12,69. Ум. фарбовідб. 13,13.

Тираж 300 пр. Вид. № 157.

Зам. № 112.

Видавництво СумДУ при Сумському державному університеті

40007, Суми, вул. Р.-Корсакова, 2

Свідоцтво про внесення суб'єкта видавничої справи до

Державного реєстру

ДК № 2365 від 08.12.05р.

Надруковано у друкарні СумДУ

40007, Суми, вул. Р.-Корсакова, 2.