

Рис. I01:

1 - окончательная траектория режущего инструмента во время цикла черновой обработки, 2 - окончательный контур

е) Последостигения точки В резание производится вдоль окончательного контура цикла черновой обработки вплоть до точки с координатой X, равной $X_b + D$. Если код G80, обозначающий окончание определения контура, находится перед указанной точкой, то резание вдоль контура при окончательном чистовом резании выполняется вплоть до точки, указанной в кадре, который предшествует кадру с кодом G80.

Величина подачи при таком резании определяется словом E, которое указывается в программе определения контура. При отсутствии кода E в соответствующей программе определения контура включается предшествующее слово. Если слово E не указано, то скорость подачи назначается при вызове цикла чернового резания.

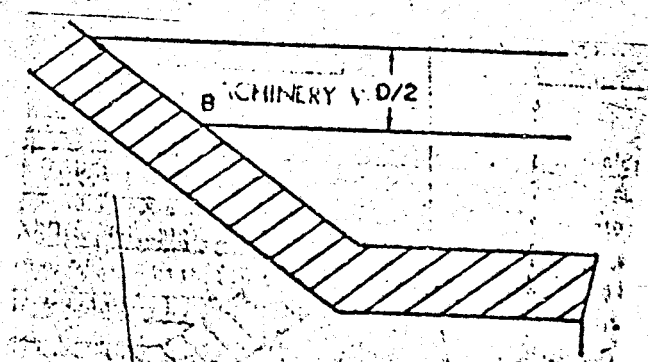


Рис. I02

ж) После завершения резания, указанного выше, режущий инструмент отводится от заготовки в направлении, противоположном направлению подачи, на врезание вдоль оси X и в сторону оси Z. Отвод происходит на величину 0,1 мм по соответствующим осям

(по диаметру в случае оси X).

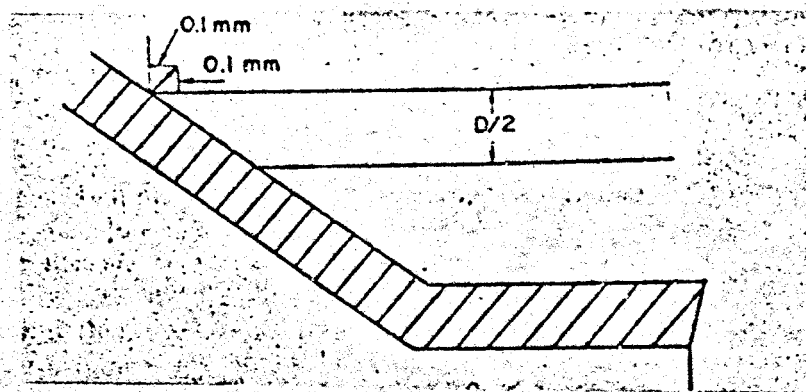


Рис. I03

э) На этом завершается первый цикл черновой обработки. По оси Z возврат осуществляется в точку Z_p , определяемую в п. "е", или происходит ускоренное перемещение вдоль оси X до точки X_p

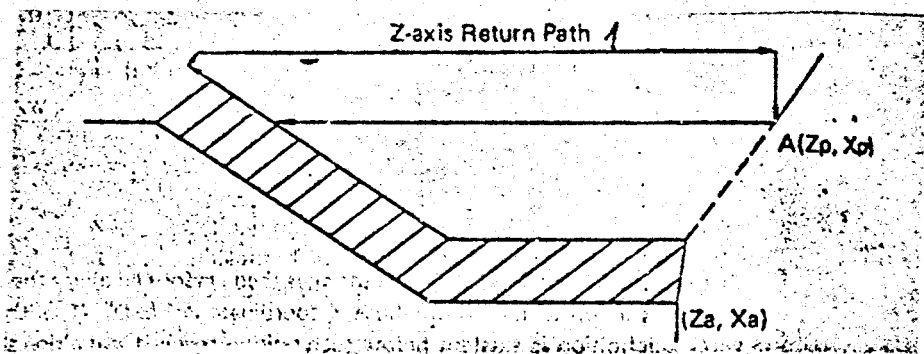


Рис. I04:

I - траектория возврата по указанной оси

и) Повторить операции "е - з" до точки изменения режимов резания. В этой точке повторяется тот же цикл, но при измененной глубине резания Φ и скорости подачи F .

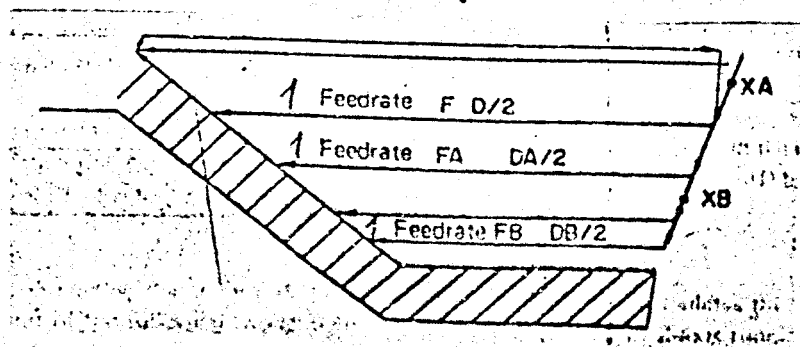


Рис. I05:

I - скорость подачи

к) Если во время операции "е" осуществляется резание по понижающему уклону или если обрабатываемый контур находится ниже точки резания X_r , то первоначальное резание происходит вдоль контура до достижения запрограммированной глубины резания, а затем осуществляется параллельно оси Z вплоть до точки, в которой прямая линия пересекается с окончательным контуром черновой обработки. Резание вдоль параллельной линии осуществляется с подачей, которая определяется словом F . (F_A/F_B).

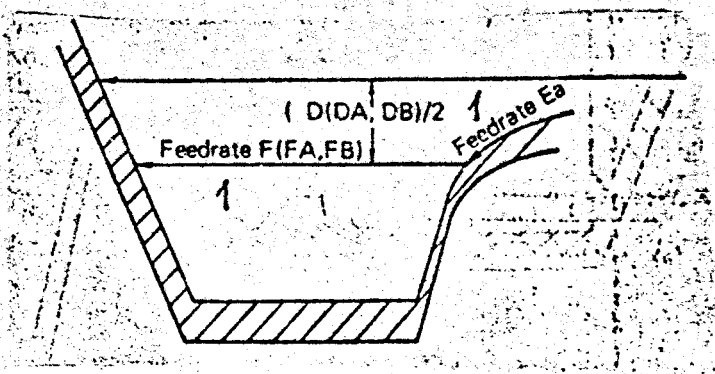


Рис. I06:

I - скорость подачи

л) Операции "е" и "ж" повторяются, возврат по оси Z осуществляется в начальную точку резания, указанную в п. "к". После завершения позиционирования по оси Z производится позиционирование по оси "х" до точки, в которой начинался предыдущий цикл резания.

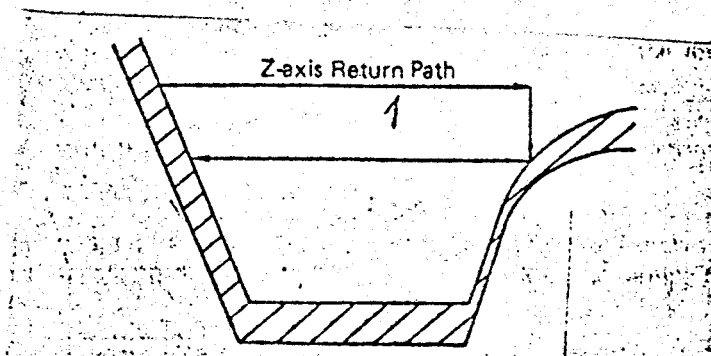


Рис. I07:

I - траектория возврата по оси

м) Этапы "к" и "л" повторяются до обработки наиболее углубленного сечения вдоль оси X . После окончательной обработки подобного сечения происходит отвод по обеим координат-

ным осям, как указывалось выше с последующим позиционированием по оси X до точки с координатными значениями первого уровня обработки по понижающемуся уклону + (D + 0, I) мм. Для возврата по оси Z используется точка с координатами, которые соответствуют точке во время цикла обработки с учетом первой траектории резания углубленного сечения. Осуществляется позиционирование в соответствующую точку по оси X.

После завершения обработки вдоль понижающегося уклона возобновляется обработка до начала подобного резания и повторяются этапы после "п." "г".

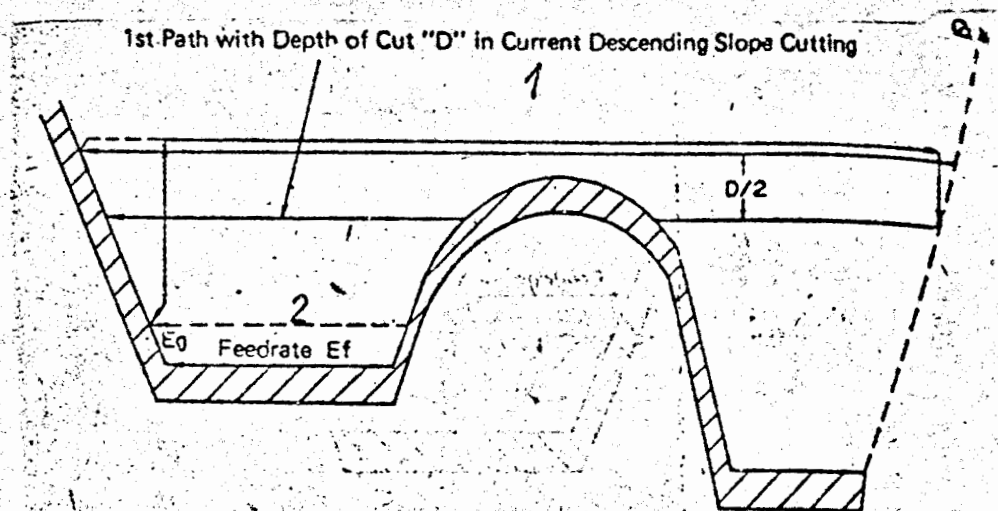


Рис. 108:

1 - первая траектория при указанной глубине резания во время выполнения обработки вдоль понижающегося уклона, 2 - скорость подачи

в) Перечисленные выше операции повторяются до достижения уровня по оси X, в котором резание осуществляется ниже значения (Xa + ζ), если используется обычная подача на врезание вдоль оси X. После достижения этой точки окончательное черновое резание производится вдоль контура, так что остается припуск на чистовую обработку вплоть до точки В.

Скорость подачи во время резания вдоль контура окончательной чистовой обработки определяется соответствующим словом.

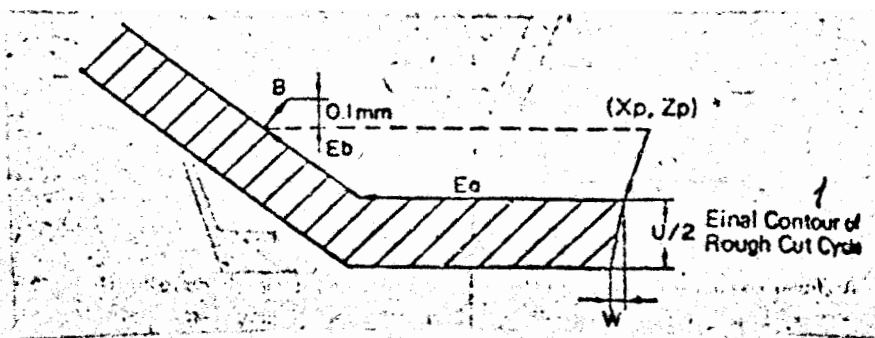


Рис.109:

I - окончательный контур цикла черновой обработки

о) После завершения предыдущего этапа осуществляется возврат по осям в начальную точку автоматического программирования (X_s, Z_s).

Ниже указываются два варианта движения возврата вдоль координатных осей :

Одновременный возврат по двум координатным осям в то время, как производится управление по первому кадру в программе определения контура (кадр после кадра, содержащего коды G81 или G82) с помощью кода G00.

- Позиционирование в первую очередь производится вдоль оси X, а затем осуществляется возврат по оси Z в начальную точку автоматического программирования после того, как установлен режим G01 в соответствующем кадре.

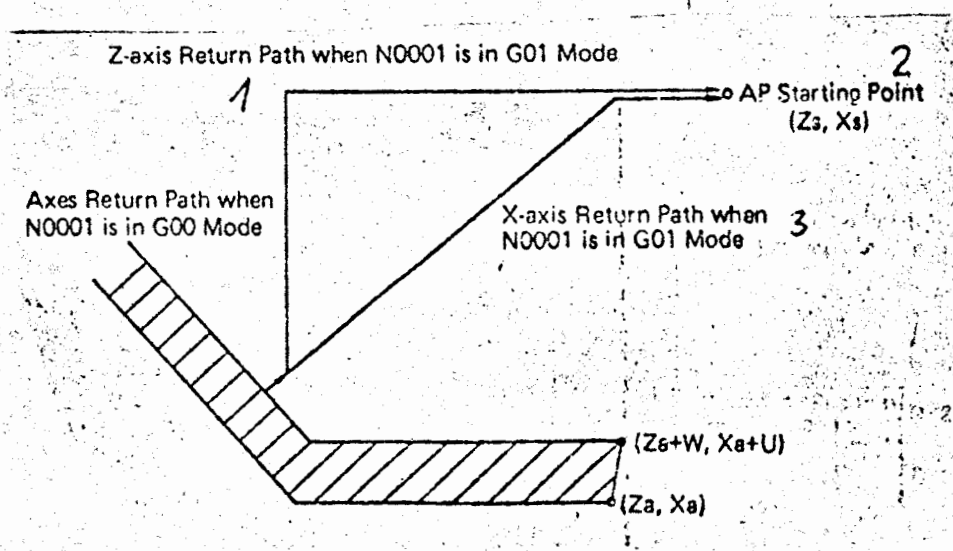


Рис.110:

1 - траектория возврата по указанной оси, если установлен режим G01 в кадре 0001, 2 - начальная точка автоматического программирования, 3 - траектория возврата вдоль оси X, если установ-

ден код 301 в кадре 0001

Если в кадре №0001 не установлены коды 600 или 601, то предусматривается этап возврата с кодом 600.

На этом цикл черновой обработки завершается.

Б. Цикл чистовой токарной обработки прутковых деталей

а) С помощью команд в кадре №0201 осуществляется позиционирование в точке смены инструментов.

б) С помощью команд в кадре №0202 производится выбор команд S, T и M для цикла чистовой обработки.

в) С помощью команды M001 в кадре №0203 система управления осуществляет поиск программы, соответствующей наименованию программы M001. Эта программа обеспечивает выполнение цикла резания в режиме токарной обработки прутков.

г) Цикл чистовой обработки выполняется в соответствии с размерными данными, содержащимися в программе определения контура при заданных режимах резания для цикла чистовой обработки.

д) После завершения цикла чистовой обработки выполняются команды в кадре, который следует за №0203.

4) Замечания, относящиеся к токарной обработке прутков.

а) Если оба слова U и W указаны для резания вдоль снижающегося уклона, возможно подрезание вдоль уклонов. В связи с этим не следует программировать указанные команды одновременно с выбором понижающегося уклона.

В случае продольного резания используется только слово U, а в случае поперечного резания только слово W.

При наличии слов U и W происходит постоянный сдвиг вдоль координатных осей X или Z в зависимости от указанного слова.

б) Максимальное допустимое количество понижающихся уклонов в режиме I автоматического программирования.

Максимальное программируемое количество понижающихся уклонов равно 10.

Для конфигурации, представленной на рис. II, количество понижающихся уклонов равно 5. Если программируется более 11 понижающихся уклонов, включается тревожная сигнализация.

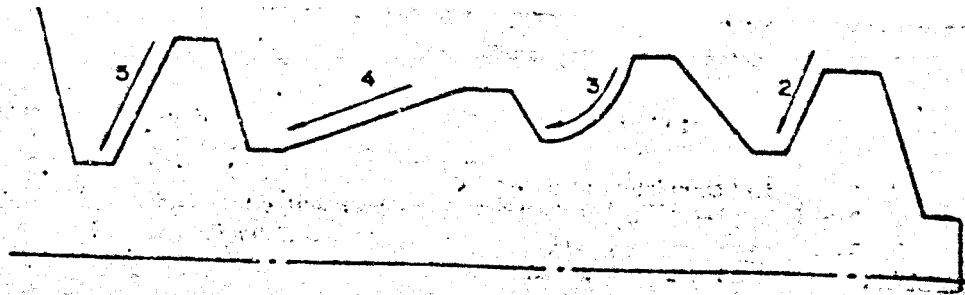
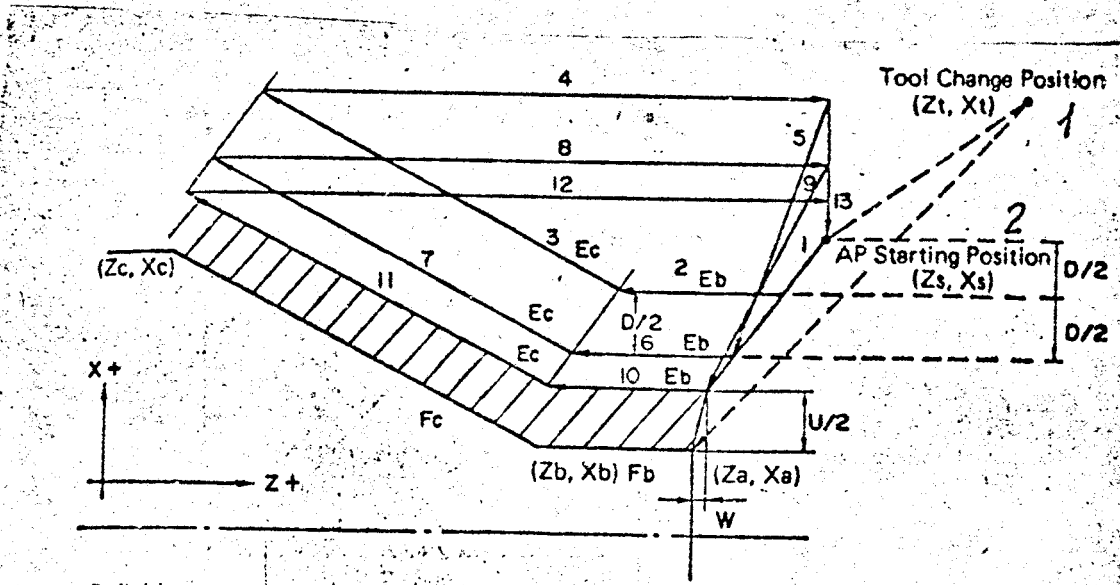


Рис. III

14.5.2. Режим 2 автоматического программирования (токарно-копировальная обработка)

I) Траектория режущего инструмента и программа для товарной обработки.



Contour Definition 3

		Feedrate and spindle speed for finish cut cycle			4
		Feedrate for rough cut cycle, rough cut along contour			5
		Start of contour definition			6
NAP20	G81				
N0201	G00	Xa	Za		
N0202	G01	Xb	Zb	Fb Sb Eb	
N0203		Xc	Zc	Fc Sc Ec	
N0204	G80				
		End of contour definition			7

1 - положение смены инструмента, 2 - начальное положение автоматического программирования, 3 - определение контура, 4 - скорость подачи и скорость шпинделя для цикла чистовой обработки, 5 - скорость подачи для цикла черновой обработки, черновая обработка по контуру, 6 - начало определения контура, 7 - окончание определения контура

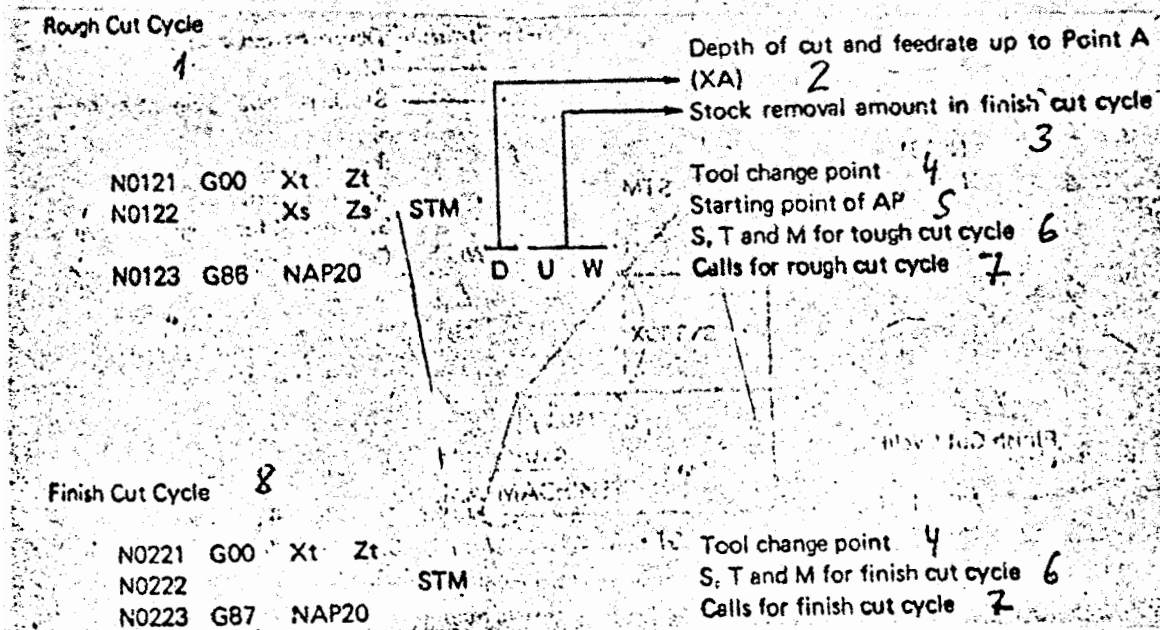
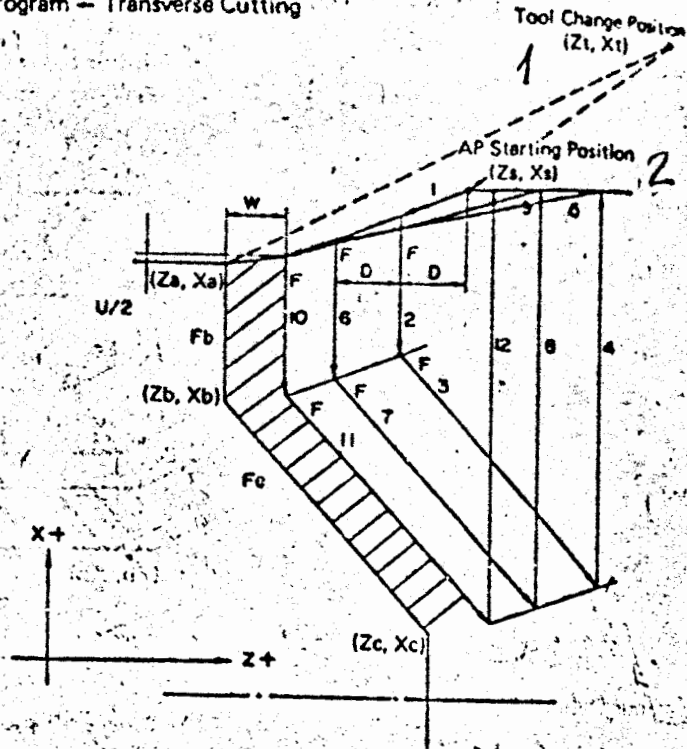


Рис.113:

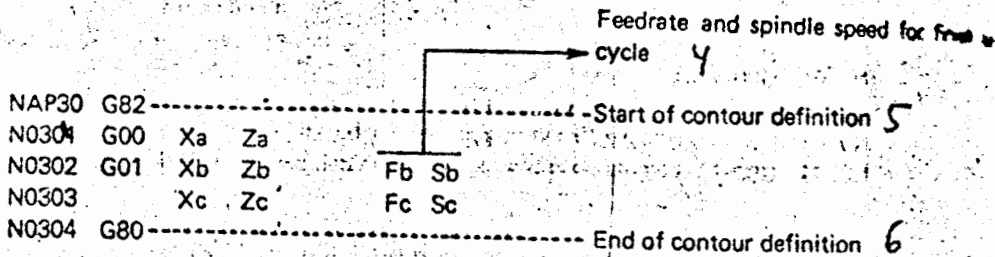
1 - цикл черновой обработки, 2 - глубина резания и скорость подачи вплоть до точки А, 3 - величина снимаемого припуска во время цикла чистовой обработки, 4 - точка смены режущего инструмента, 5 - начальная точка для автоматического программирования, 6 - команды для цикла черновой обработки, 7 - вызов цикла черновой обработки, 8 - цикл чистовой обработки

2) Траектория инструмента и программа для поперечного
2134--E P-134 резания

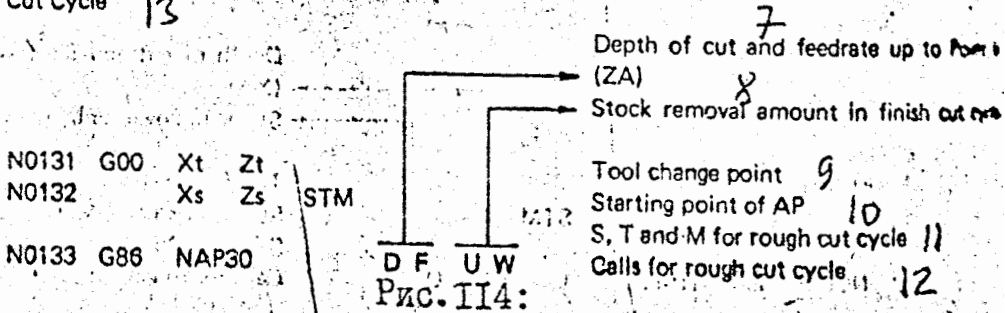
(2) Tool Path and Program -- Transverse Cutting



Contour Definition 3



Rough Cut Cycle 13



Finish Cut Cycle 14



1 - положение смены инструмента, 2 - начальная точка для автоматического программирования, 3 - определение контура, 4 - скорость подачи и скорость шпинделя для чистового цикла обработки, 5 - начало определения контура, 6 - окончание определения контура, 7 - глубина резания и величина подачи до указанной точки, 8 - величина снимаемого припуска для цикла чистовой обработки, 9 - точка смены инструмента, 10 - начальная точка автоматического программирования, 11 - команды для цикла черновой обработки, 12 - вызов цикла черновой обработки, 13 - цикл черновой обработки, 14 - цикл чистовой обработки

3) Описание цикла токарно-копировальной обработки

А. Цикл черновой обработки при продольном резании

а) С помощью команд, содержащихся в кадре № 0121, установить положение смены инструмента.

б) С помощью команд в кадре № 0122 выбрать команды S, T и M для цикла черновой обработки и произвести позиционирование начальной точки для автоматического программирования.

При отсутствии команд S, T или M в соответствующем кадре действуют команды, выбранные для предшествующих кадров.

в) В случае команды MАР20 в кадре № 0123 система управления производит поиск программы, которой присвоено наименование MАР20. В соответствии с этой программой вводится цикл чернового резания в режим токарно-копировальной обработки.

В этом кадре указываются также следующие параметры для цикла чернового резания.

D - глубина резания

U - составляющая X для припуска на чистовую обработку

W - составляющая Z для припуска на чистовую обработку

При необходимости программировать также слово F. При отсутствии слова F в программе определения контура становится эффективной скорость подачи, введенная командой в предшествующий кадр.

г) После считывания команд в кадре № 0201 программы определения контура система управления рассчитывает точку пересечения указанных ниже двух прямых линий. Первая линия параллельна оси Z и проходит через точку $X_s - D/2$, а вторая проходит через две точки (X_s, Z_s) и (X_{s+1}, Z_{s+1}) . Выполнить позиционирование в

расчетной точке

Одновременно с позиционированием в системе управления рассчитывается расстояние (XOFF, ZOFF) между указанными двумя точками (X_p, Z_p) и (X_a+U, Z_a+W) (см. рис. II5)

$$\begin{aligned} X_p &= X_s - D \\ Z_p &= Z_a + W + (Z_s - Z_a - W) (1 - D/(X_s - X_a - U)) \\ XOFF &= X_p - (X_a + U) \\ ZOFF &= Z_p - (Z_a + W) \end{aligned}$$

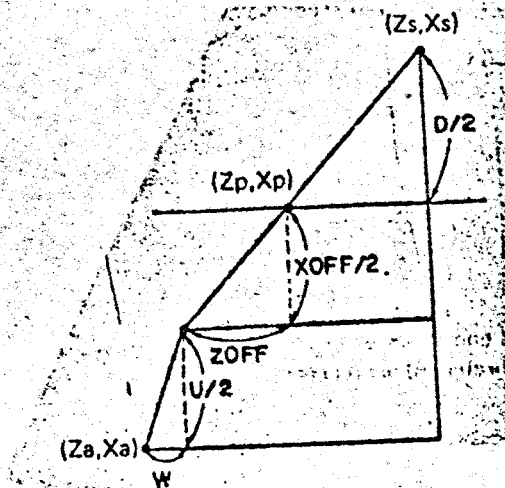


Рис. II5

В случае чрезмерного припуска на чистовую обработку вводятся команды U или W , обеспечивающие реверсирование направления подачи на врезание и включается тревожная сигнализация.

д) Производится расчет целевой точки * I.

Целевая точка * I получается сдвигом точек, определяемых командой в соответствии с программой поиска контура параллельно соответствующим осям $(XOFF+U, ZOFF+W)$

Обработка производится с подачей, которая определяется словом E в каждом кадре программы.

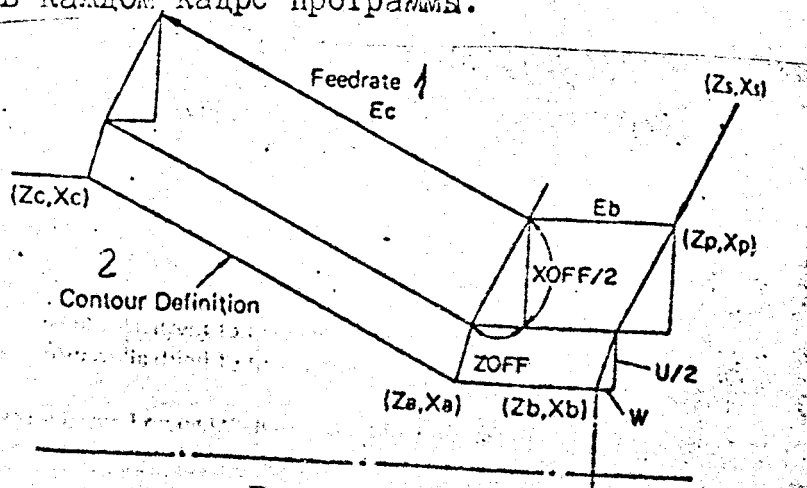


Рис. II6:

1 - скорость подачи, 2 - определение контура

е) Повторить предыдущий этап до завершения определения контура (действующий код G80).

Затем осуществляется возврат по оси Z до координаты Z_s начальной точки автоматического программирования.

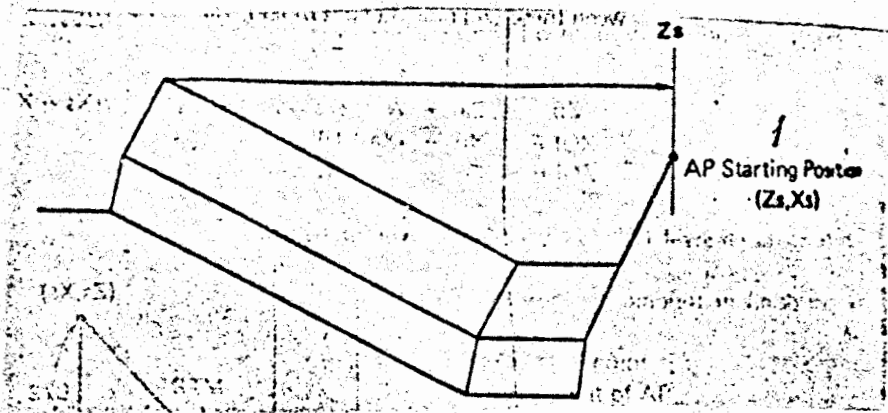


Рис. II7:

I - начальная точка автоматического программирования

ж) На этом завершается один цикл черновой обработки. Производится расчет новых значений X_{off} , Z_{off} и повторяются этапы "г - е".

з) Повторяются перечисленные выше этапы до тех пор, пока точка подачи не превысит значения $X_a + u$. В этой точке в системе управления используются координаты (0,0) вместо (X_{OFF} , Z_{OFF}), а резание выполняется вдоль траектории со сдвигом от заданного контура на величину (u, w).

В конце определяемого контура осуществляется сдвиг по оси Z в такое же координатное положение, как и в начальной точке автоматического программирования, а затем осуществляется сдвиг по оси X в начальную точку автоматического программирования.

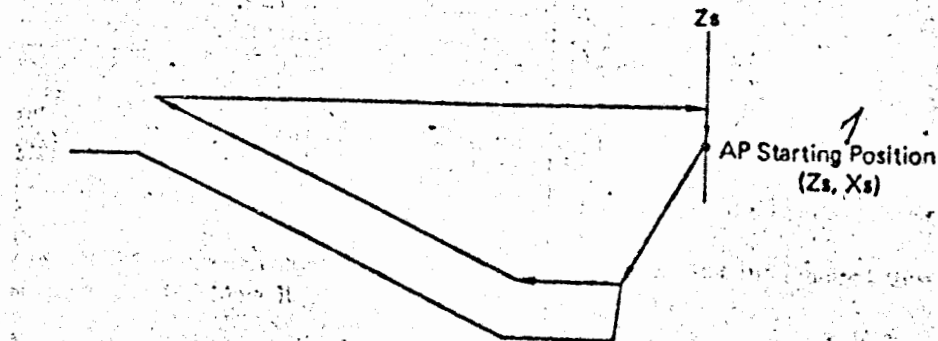


Рис. II8:

I - начальная точка автоматического программирования

и) На этом завершается цикл черновой обработки и выполняются кадры с командами после №0124 .

Б. Цикл чистового резания при токарно-копировальной обработке

а) В соответствии с командами кадра №0221 выполняется позиционирование в точке смены инструментов.

б) С помощью команд в кадре № 0222 осуществляется выбор команд S, T и M для цикла чистовой обработки.

в) С помощью команды NPAR1 в кадре №0223 система управления осуществляет поиск программы, связанной с наименованием программы NAR20. В соответствии с этой программой осуществляется цикл чистового резания в режиме токарно-копировальной обработки.

г) Цикл чистовой обработки выполняется после введения размерных данных, предусмотренных программой выбора контура с учетом заданных режимов резания для цикла чистовой обработки.

д) После завершения цикла чистовой обработки выполняются команды в кадре после №0223.

14.5.3. Режим 3 автоматического программирования (непрерывный цикл нарезания резьбы)

1) Траектория резца и программа для режимов G34 и G35

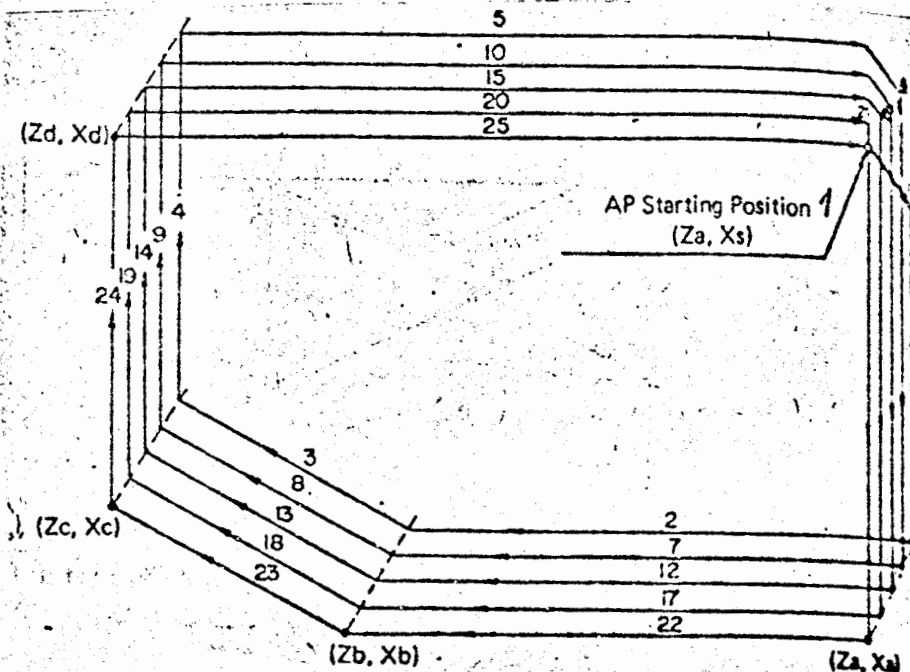


Рис. II9:

I - начальное положение для автоматического программирования

Contour Definition 1

NAP40	G81		
N0401	G00	Xa	Za
N0402	G34	Xb	Zb
N0403		Xc	Zc
N0404	G01	Xd	Zd
N0405	G80		

Programming Calling for Thread Cutting Cycle 2

N0141	G00		S T M
N0142		Xs	Zs
N0143	G88	NAP40	M32(M33) M73(M74, M75) V H D U

Рис.120 (продолжение рис.119)

1 - определение контура, 2 - программа вызова цикла нарезания резьбы

2) Описание цикла непрерывного нарезания резьбы

а) С помощью команд в кадре №0141 производится выбор команд S, T и M для нарезания резьбы

б) Произвести позиционирование в начальную точку автоматического программирования (Xs, Zs) с использованием команд, содержащихся в кадре №0142.

в) Слова V, H, D и U в кадре № 0143 обозначают данные, необходимые для цикла нарезания резьбы.

V - угол вершины вставки резьбонарезного резца

H - высота нарезаемой резьбы

D - глубина резания

U - величина снимаемого припуска при чистовом резании

Для выбора режима нарезания резьбы и схемы подачи резца на врезание используются коды M двух типов.

- Код G88 NAP40 вызывает программу определения контура и обеспечивает выполнение требуемого цикла нарезания резьбы (режим 3 автоматического программирования).

Подробные сведения, относящиеся к циклу нарезания резьбы, приведены ниже.

Примечание 1. Для нарезания резьбы на торцевой поверхности нужно использовать G82 для определения контура резьбы вместо кода G81, который используется в режимах 1 и 2 автоматического программирования. В этом случае программа M27 определяет соответствующую ось для шага резьбы по оси X в кадре G34/G35. Окончательный припуск устанавливается словом W вместо слова U.

3) Подробные сведения, относящиеся к циклу нарезания резьбы.
а) На рис. I2I приводятся данные, необходимые для цикла нарезания резьбы.

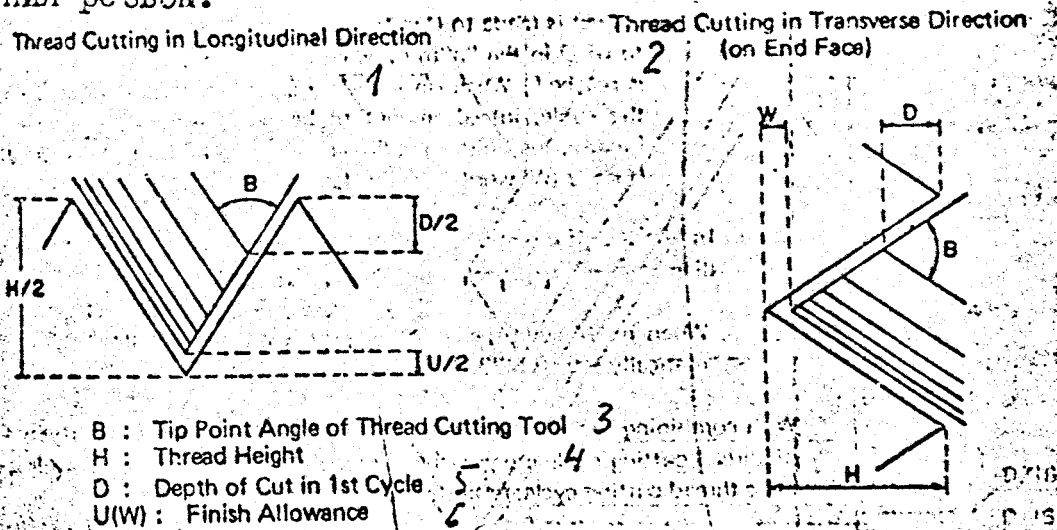


Рис. I2I:

1 - нарезание резьбы в продольном направлении, 2 - нарезание резьбы в поперечном направлении (на торцевой поверхности), 3 - угол вершины резьбонарезного резца, 4 - высота резьбы, 5 - глубина резания для первого цикла, 6 - припуск на чистовую обработку

б) Коды M, используемые для назначения режимов резания и схемы подачи на врезание .

- Режимы резания

M32 - подача на врезание по прямой линии вдоль поверхности резьбы

M33 - зигзагообразная подача на врезание

Если не указан ни один из этих кодов M, то система управления автоматически выбирает код M32.

- Схема подачи на врезание

M73 - схема I подачи на врезание

Подача на врезание осуществляется в соответствии с диаметром D при каждом цикле нарезания резьбы вплоть до точки, удаленной на D мм от положения $H-u(w)$. После достижения этой точки величина подачи на врезание изменяется до $D/2$, $D/4$ и $D/8$, причем припуск на чистовую обработку определяется с помощью $u(w)$. Во время цикла чистовой обработки подача осуществляется на величину, указанную $u(w)$.

Если слово $u(w)$ не указано, то цикл чистовой обработки не выполняется.

M74 - схема 2 подачи на врезание

Подача на врезание осуществляется по диаметру D при каждом цикле нарезания резьбы вплоть до достижения положения $H-u(w)$. После этого цикл чистовой обработки осуществляется с величиной подачи на врезание $u(w)$. Если слово $u(w)$ не указано, то цикл подачи на врезание не выполняется.

M75 - схема 3 подачи на врезание

Подача на врезание осуществляется по D в соответствии с первым циклом, до $\sqrt{2} D$ в соответствии с вторым циклом и до $\sqrt{n} D$ в соответствии с n -тым циклом до достижения положения $H-u(w)$. После того, как $(\sqrt{n} - \sqrt{n-1}) \times D$ становится меньше, чем $u(w)$ во время цикла, величина подачи на врезание становится равной $u(w)$. И, наконец, цикл чистовой обработки выполняется с величиной подачи, равной $u(w)$. Если слово $u(w)$ не указано, то цикл чистовой обработки не выполняется.

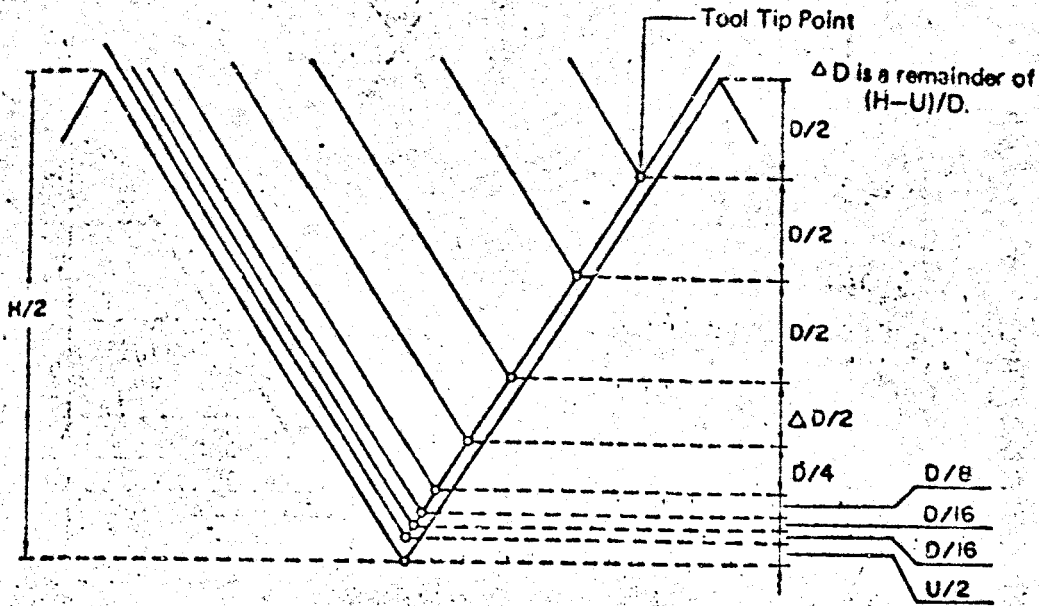
Примечание. В направлении оси X действительная величина подачи равняется $D/2$, так как D определяется диаметром.

Если коды M , определяющие схему подачи на врезание, не указаны, то система управления автоматически выбирает код M73.

За счет комбинирования кодов M , определяющих режим резания и схему подачи на врезание, можно обеспечить шесть вариантов цикла нарезания резьбы при продольном нарезании резьбы и три цикла нарезания поперечной резьбы.

(i) Longitudinal Thread Cutting Cycle

a) M32 + M73



b) M33 + M73

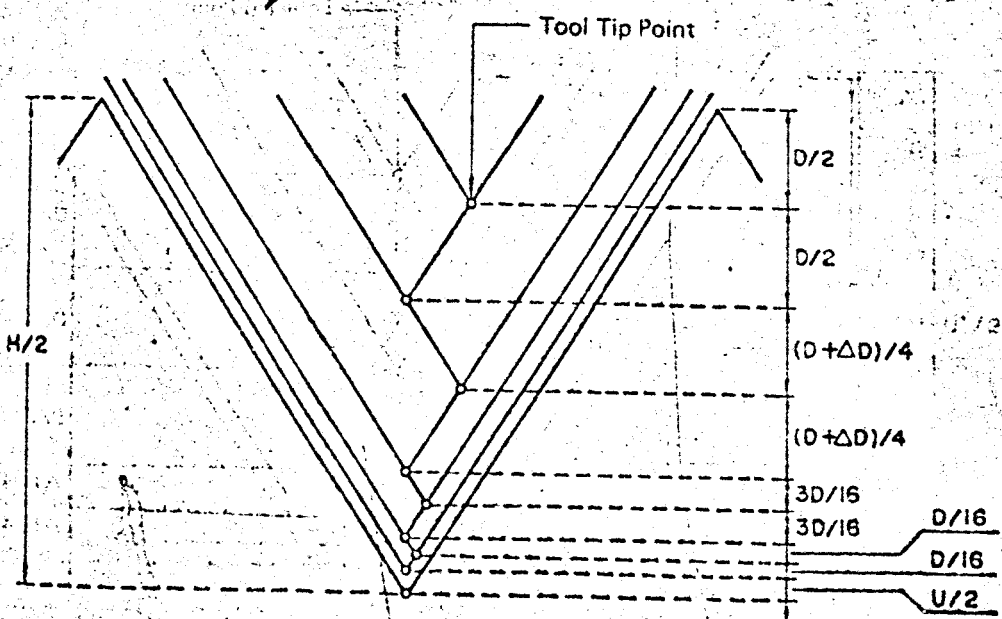
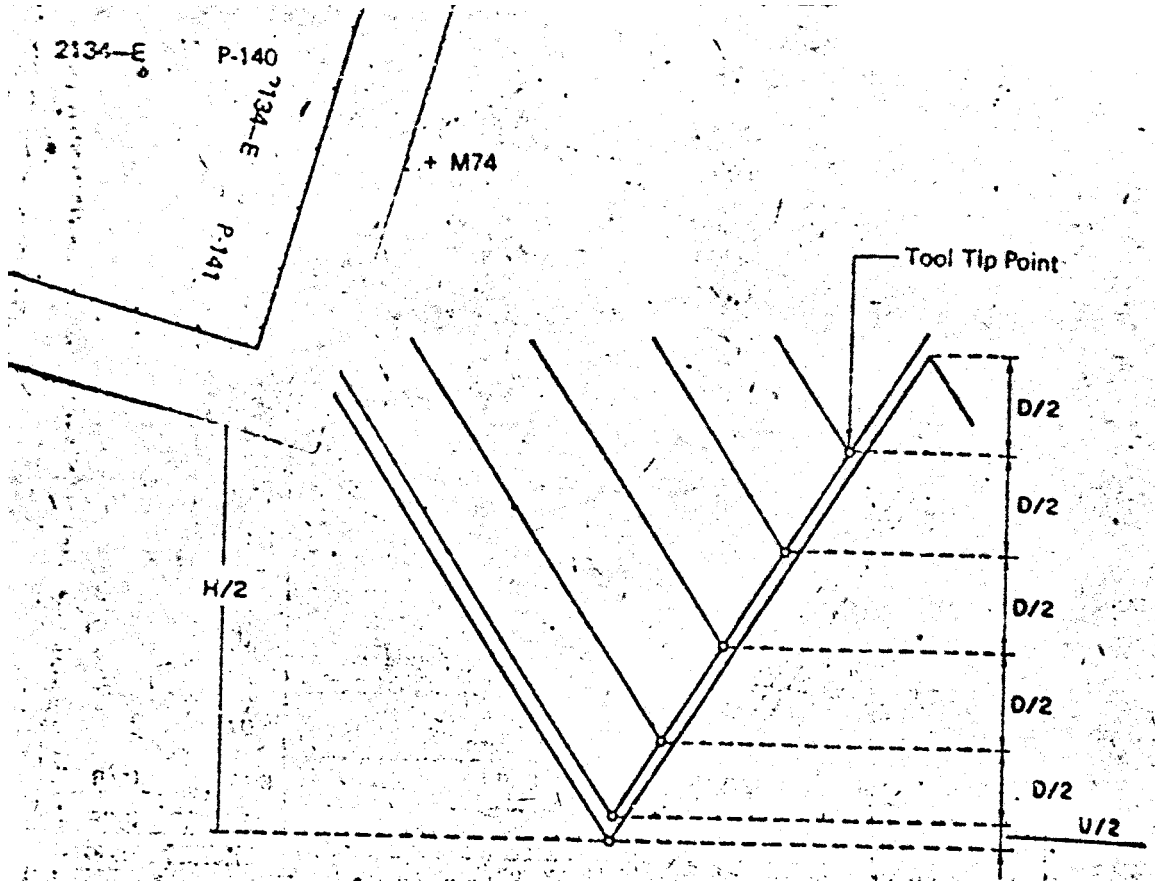


Рис.122. Цикл продольного нарезания резьбы:
 1 - вершина резца, 2 - остаточный размер



d) M33 + M74

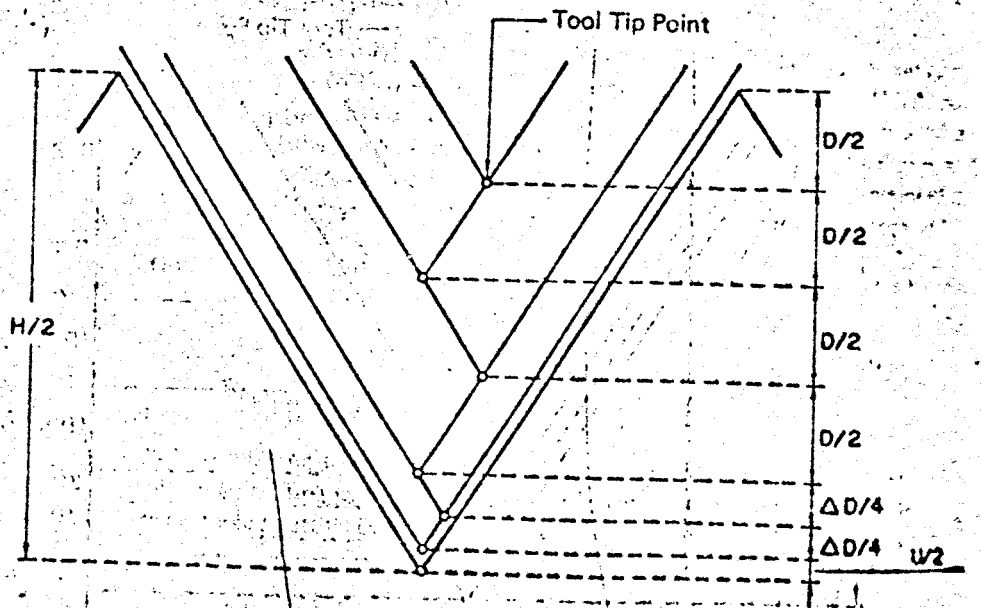
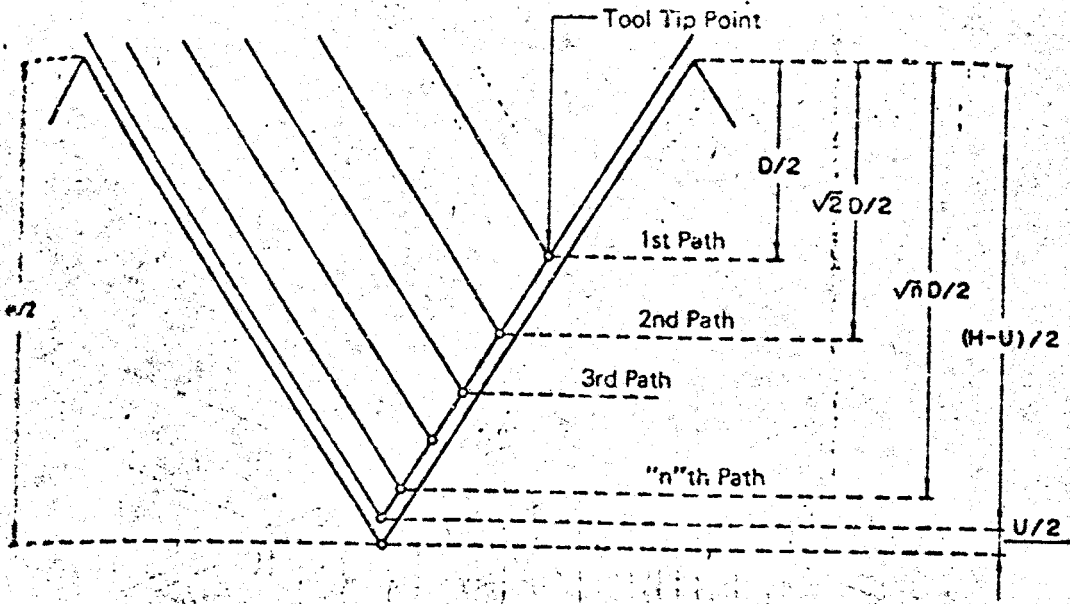


Рис. 123:

I - вершина резца

e) M32 + M75



f) M33 + M75

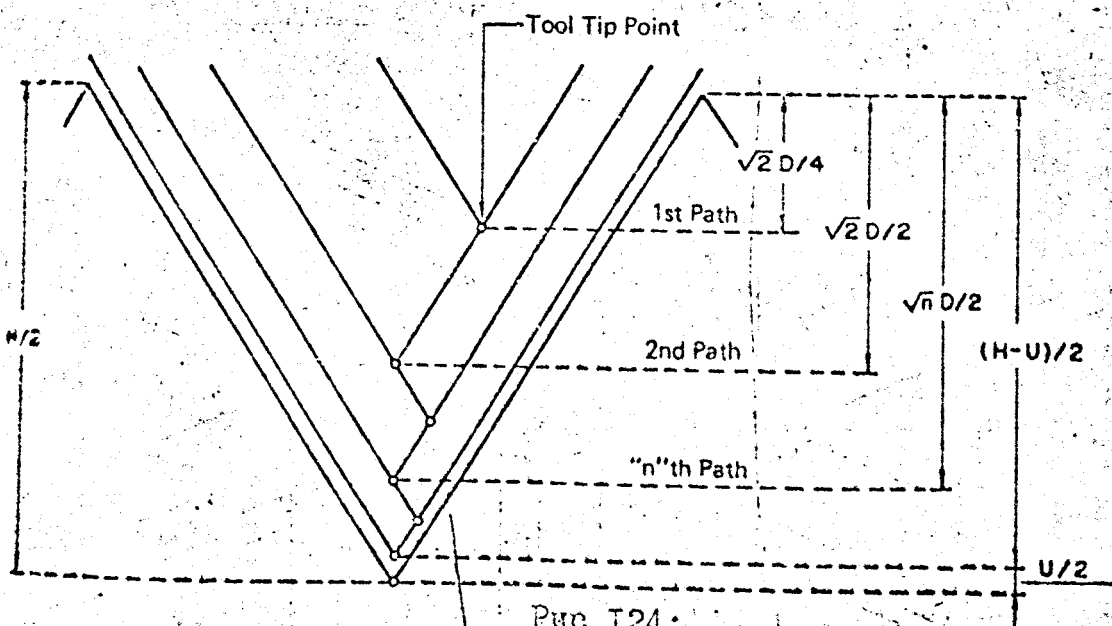
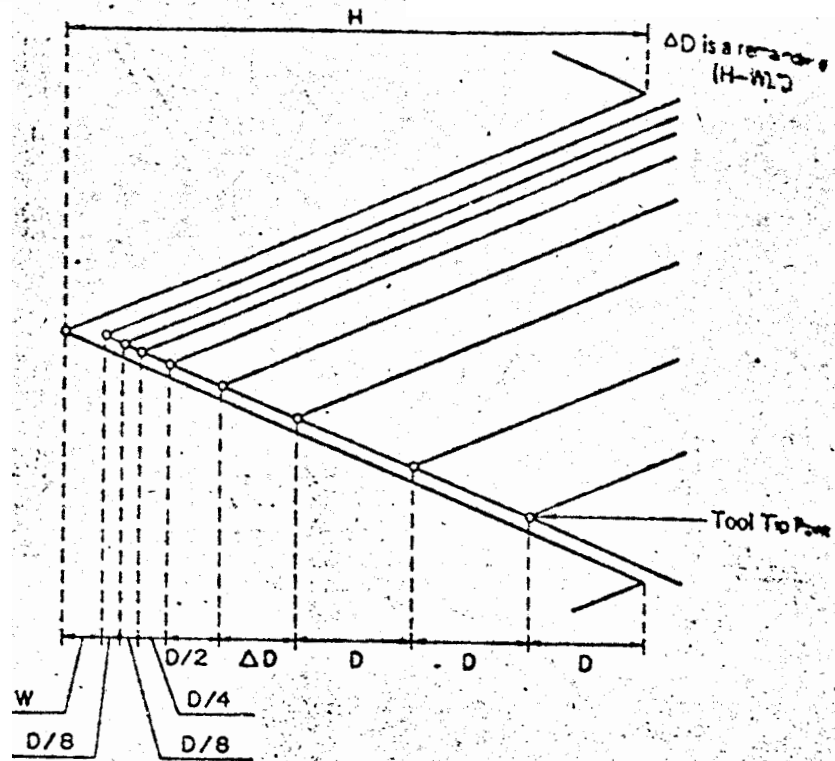


Рис. 124:

1 - верхняя резца, 2 - номер прохода

ii) Transverse Thread Cutting Cycle

a) M32 + M73



M33 + M73

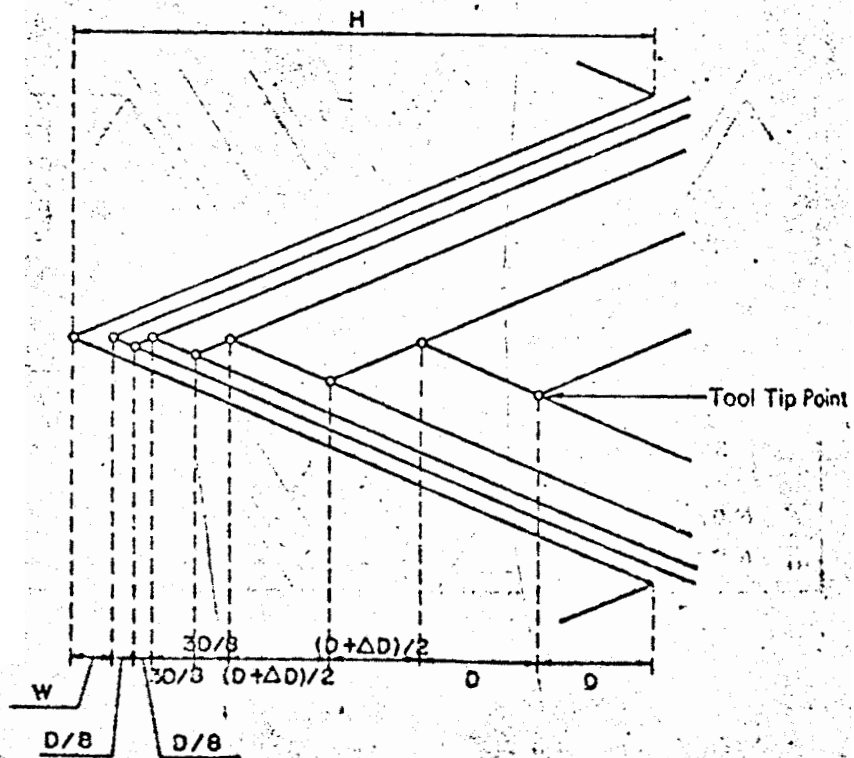
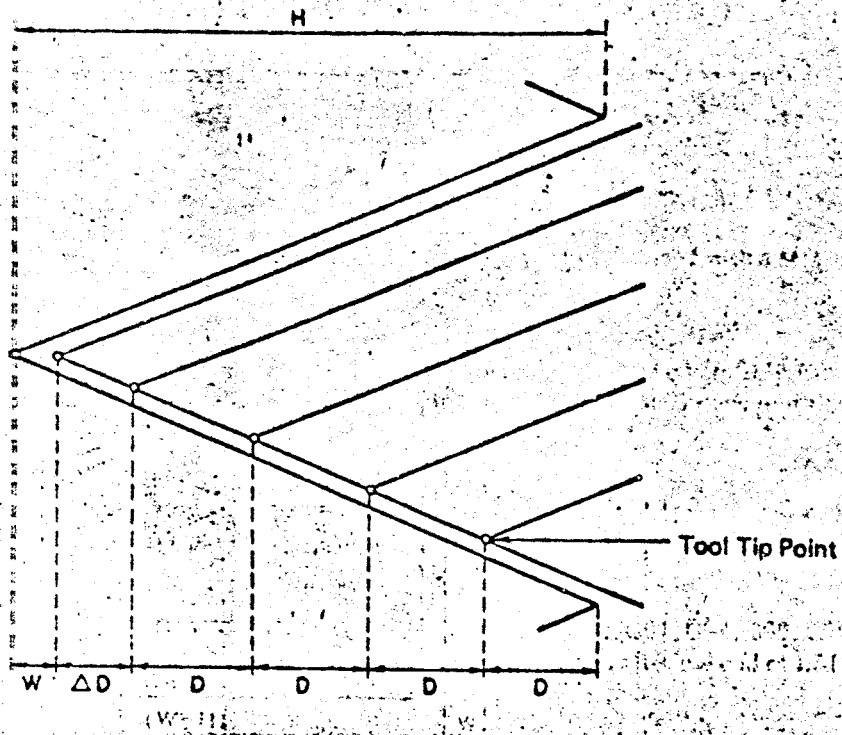


Рис. I25:

1 - остаточный размер, 2 - вершина резца,

c) M32 + M74



d) M33 + M74

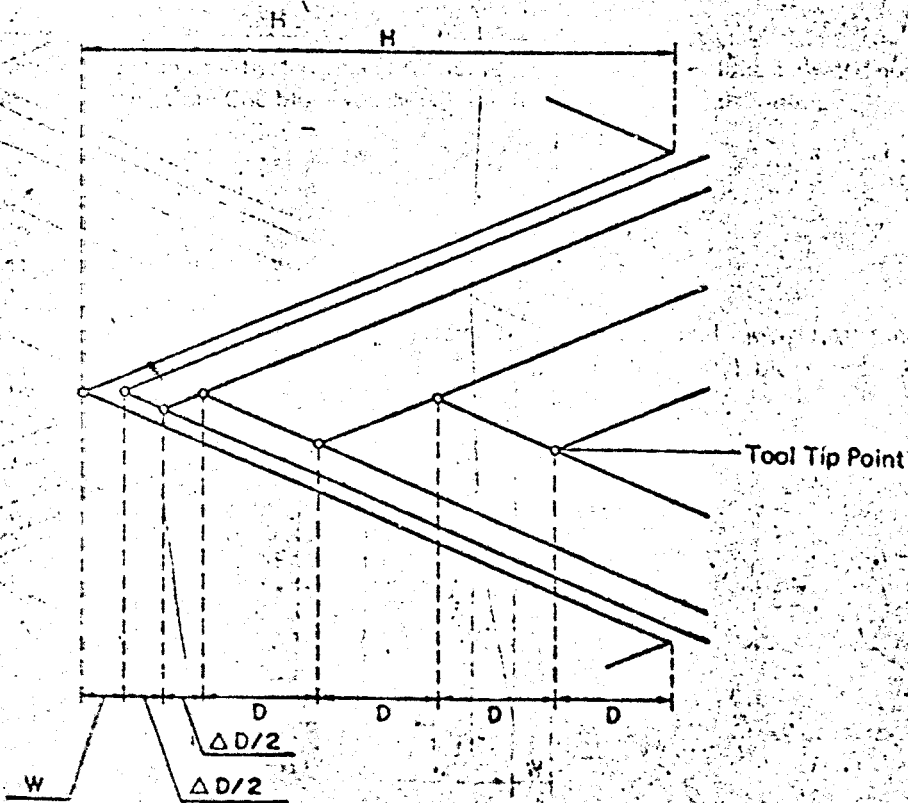
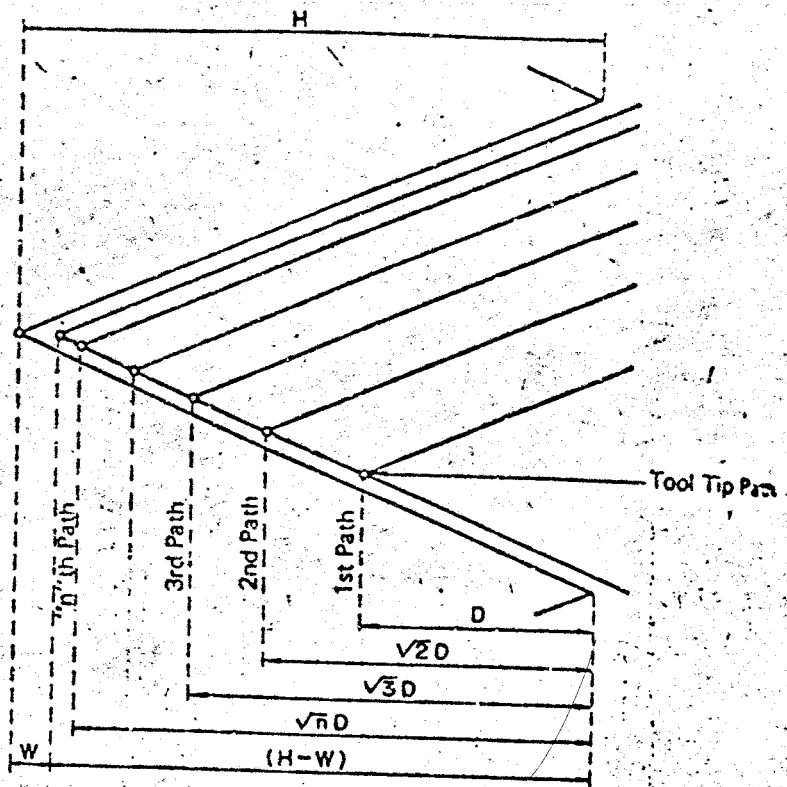


Рис.126:

e) M32 + M75



f) M33 + M75

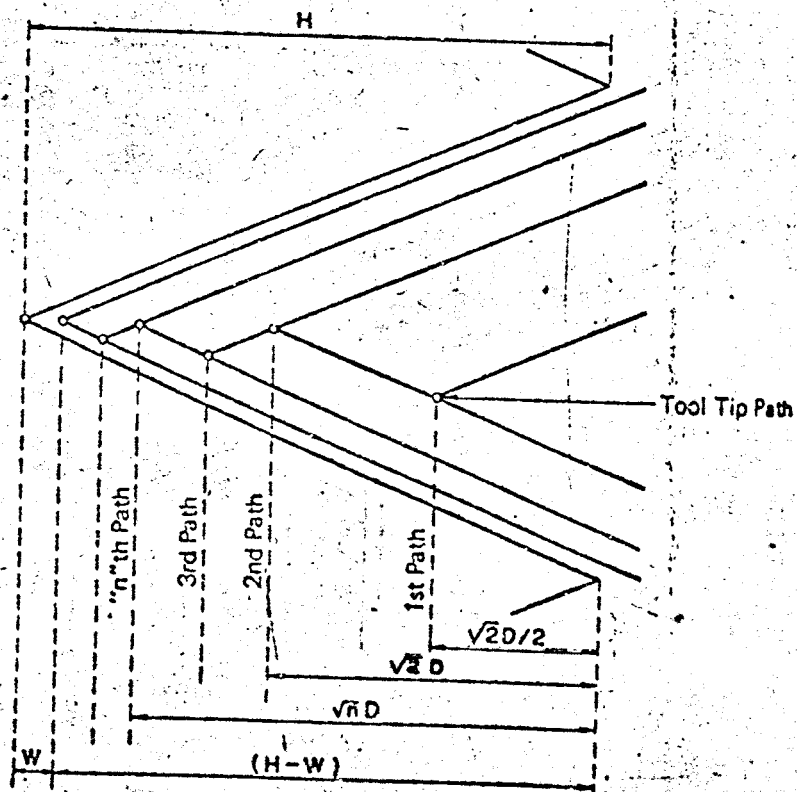


Рис. 127:

I - вершина резца, 2 - номер прохода

14.6. Меры предосторожности

Ниже перечислены меры предосторожности при автоматическом программировании на токарном станке:

1) Убедиться в том, что при программировании контура номер последовательности устанавливается сразу после кода G для вызова исполнения программы IAP.

G85, G86, G87 и G88

2) Коды G81 и G82, используемые для индикации начала определения контура, должны иметь соответствующее наименование последовательности.

3) Для абсолютного или пошагового программирования коды G90 и G91 включают коды G85, G86, G87 или G88. Однако, если код G выбирает другую размерную систему, то вводятся изменения в указанную программу определения контура.

4) Для кодов G64, G65, G94, G95, G96 и G98 выбираются команды G85, G86, G87 или G88. После введения невозможны изменения в пределах программы определения контура.

5) Для кодов M00, G01, G02, G03, G31, G32, G33, G34, G35, G64, G65, G90, G91, G94, G95, G96 и G97 вводятся команды G85, G86 и G87 после окончания автоматического программирования.

6) Невозможен переход от одной автоматической программы к другой.

7) Если в код G вызова автоматического программирования IAP (G85, G86, G87 и G88) вводится режим компенсации радиуса при вершине резца, то включается тревожная сигнализация.

8) Компенсация радиуса закругления резца вводится во время автоматического программирования, однако нужно отменить этот активный режим до прохождения кадра G80 в конце определения контура.

NLAP1	G81		
N0001	G00	Xa	Za
N0002	G41(G42)G00	Xb	Zb Fb
:	:	:	:
:	:	:	:
N0010	G40	X	Z
N0011	G80		
	G40		
	G80		

Be sure to activate and cancel IAP function between G81 (G82) block and G80 block.

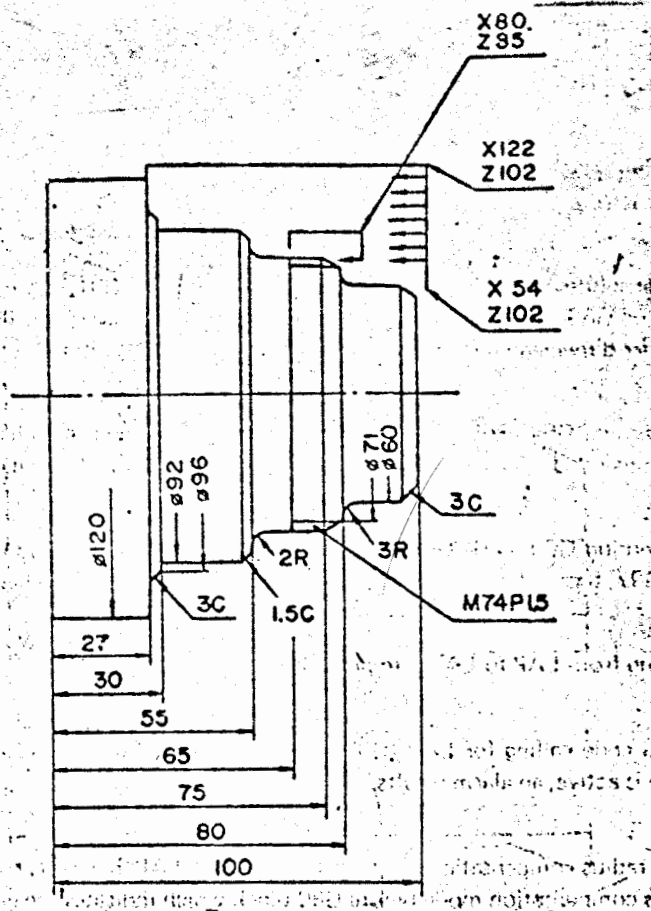
Рис. 128:

I - обязательно включить и отключить функцию IAP между кадрами

G81 (G82 и G80).

14.7. Применение функций IAR

В данном разделе приводится пример использования режима I автоматического программирования.



Example of Program

00001					
N001	G81				
N002	G00	X54	Z102		
N003	G01		Z100		
N004		X60	Z97		
N005			Z83		
N006	G02	X66	Z80	I3	
N007	G01	X71			
N008		X74	Z75		E0.4
N009			Z57		
N010	G02	X78	Z55	I2	E0.45
N011	G01	X89			
N012		X92	Z53.5		E0.4
N013			Z30		
N014		X96			E0.45

Рис. 129. Пример программы

- I75 -

N1014	X102	Z27								(*8)
N1015	X122									
N1016	G80									
N100	G00	X800	Z102							(*1)
N101				S900	T0101	M43	M03			(*2)
N102	X122									(*3)
N103	G85	NAP1		D8	U0.2	F0.45				(*4)
N104	G00	X800	Z102					M05		
N105				S1000	T0303			M03		(*5)
N106	G87	NAP1								(*6)
N107	G00	X800	Z102					M05		
N108				S950	T0505			M03		
N109	X80	Z85								
N110	G33	X72.9	Z65			F1.6				
N111		X72.3								
N112		X71.9								
N113		X71.73								
N114	G00	X800	Z102					M05		
N115								M02		

- *a Contour defining program 1
- *1 Tool change position 2
- *2 S, T and M for rough cut cycle 3
- *3 Rough cut starting point 4
- *4 Calling for bar turning rough cut cycle 5
- *5 S, T and M for finish cut cycle 6
- *6 Calling for finish cut cycle 7

Note: Contour defining program beginning from G81 and ending with G80 may be entered any part within the above indicated program.

Рис. I80 (продолжение рис. I29):

I - программа определения контура, 2 - программа смены инструмента, 3 - команды для цикла черновой обработки, 4 - начальная точка черновой обработки, 5 - вызов чернового цикла токарной обработки прутков, 6 - для цикла чистовой обработки, 7 - вызов цикла чистовой обработки, 8 - примечание. Возможен ввод любой части программы определения контура, начиная с G81 и кончая G80 в пределах указанной выше программы

Раздел I5. Функция компенсации радиуса при вершине резца

I5. I. Применение

I5. I. I. Общее описание

Функции обработки данных в системе ЧПУ типа OSP 5000 облегчаются за счет применения функции компенсации радиуса при вершине резца.

Радиус закругления вершины резца на большинстве резцов, используемых для токарных операций, фактически находится между программируемыми траекториями резца и действительным окончательным

контуром детали, поэтому функция компенсации радиуса закругления реза учитывает геометрическую ошибку и осуществляет автоматическую корректировку с использованием упрощенного программирования.

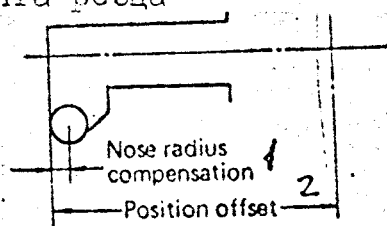
При операциях контурной обработки матриц или при фрезерных операциях резание осуществляется фрезами большого диаметра. В системах управления предусматривается функция сдвига фрезы, так что фактически образуемая траектория режущего инструмента имеет сдвиг относительно запрограммированной траектории на величину, эквивалентную радиусу применяемой фрезы. Подобная функция является стандартной для системы ОУР 5000. Однако в сочетании с функцией сдвига фрезы применяемая функция компенсации радиуса вершины реза позволяет облегчить работу на токарном станке.

15.1.2. Компенсация радиуса закругления вершины реза для токарной обработки

В отличие от фрезерования для операций токарной обработки требуются дополнительные функции.

15.1.2.1. Сдвиг реза и компенсация радиуса закругления при вершине.

При токарных операциях для чистовой обработки заготовки используются различные типы и конфигурации резцов, например, резцы для обработки по внутреннему диаметру, резцы для обработки по наружному диаметру, резцы для черновой и чистовой обработки, сверла и т.д.. Как правило, функция компенсации радиуса закругления вершины реза должна выполняться одновременно с функцией сдвига реза



The tool nose radius function is activated by a six-digit T command:

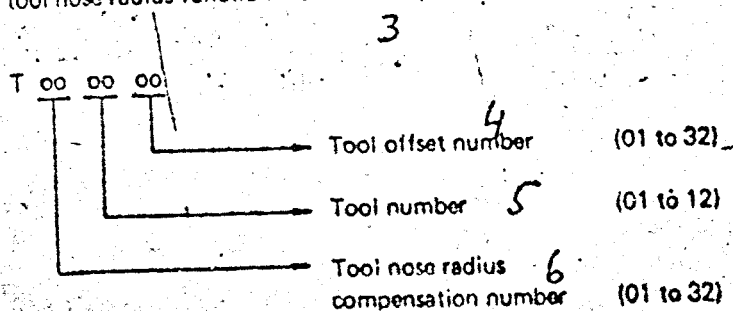


Рис. 131:

I - компенсация радиуса при вершине резца, 2 - положение сдвига, 3 - включение функции радиуса закругления вершины резца осуществляется шестизначной командой T, 4 - номер сдвига резца, 5 - номер резца, 6 - номер компенсации радиуса закругления вершины резца

15.1.2.2. Компенсация радиуса при вершине резца в точке разрыва

Одной из характерных особенностей операции токарной обработки является наличие многочисленных точек разрыва на траектории режущего инструмента. Более того, многие из этих точек характеризуются углами менее, чем 180° .

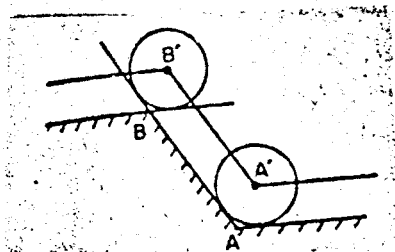


Рис.132. Траектория резца в системе 5000I

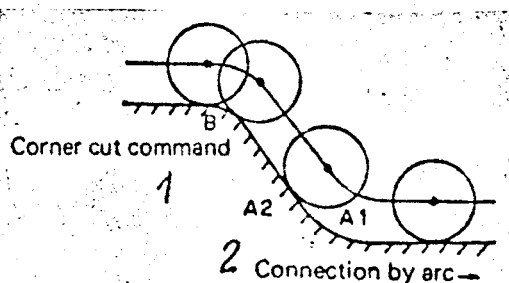


Рис.133: Обычная система программирования:

1 - команда на подрезание угла, 2 - соединение дугой

Точка A на рис.132 представляет точку разрыва с углом менее 180° . В этом случае программа, заложенная в систему OGP 5000 вводит функцию компенсации закругления радиуса при вершине резца и позволяет построить траекторию движения резца за счет простого ввода координат точки A и точки B. Система управления ,

в которой используются обычные методы программирования, предусматривает плавное соединение подобных точек за счет введения помимо точки А дополнительных точек А1 - А2 (см. рис.133). Система ОУР 5000I предусматривает осуществление подобной функции, причем команды на подрезание угла необходимы в точке разрыва с углом более 180° , а точка В больше не нужна. Размерные слова для точек А и В позволяют построить траекторию режущего инструмента А' В' по отношению к центру вершины резца.

15.1.3. Операция, связанная с компенсацией

15.1.3.1. Геометрическая ошибка резания, связанная с радиусом закругления вершины резца.

Если резание осуществляется по траекториям А, В-С-Д-Е (см. рис.135) без осуществления функции компенсации радиуса закругления вершины резца, то остаются неподрезанные участки, показанные штриховкой, которые создают геометрические ошибки. Это получается из-за того, что наладка режущего инструмента осуществляется в расчете на резание воображаемой точки Р (см. рис.134) с учетом базовой точки и программирования траектории в соответствии с командами в системе ЧПУ. Однако фактическая точка режущей кромки не находится точно в базовой точке из-за радиуса закругления вершины резца, так что получаются геометрические ошибки.

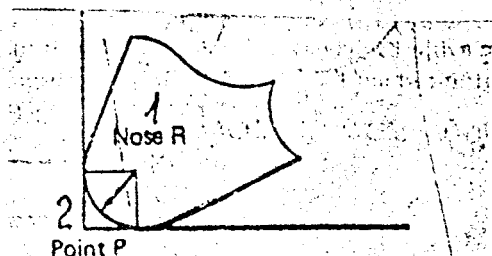


Рис.134. Наладочная точка инструмента:
1 - радиус закругления вершины, 2 - точка

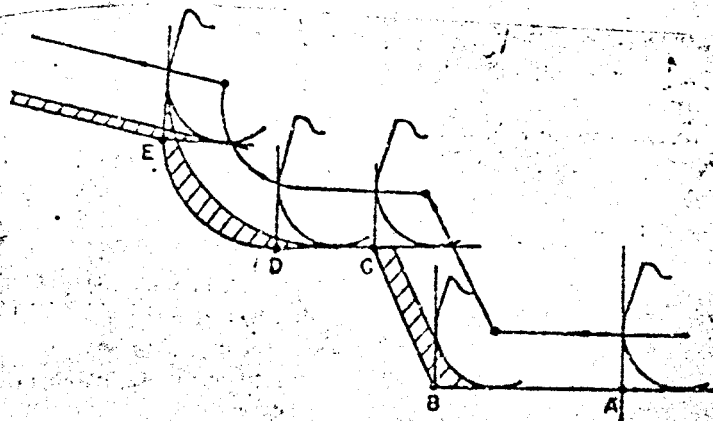


Рис.135. Траектория режущего инструмента и возникающая

ошибка без компенсации радиуса закругления вершины резца

Функция компенсации радиуса закругления вершины резца обеспечивает автоматическую компенсацию ошибок, связанных с радиусом закругления и с разностью между программируемыми и фактическими траекториями.

15.1.3.2. Движение компенсации

После включения функции компенсации радиуса закругления вершины резца производится изменение траектории резца (см. рис.136) для исключения неподрезанных участков, представленных на рис.135. При этом осуществляется точная окончателная обработка, предусмотренная программой.

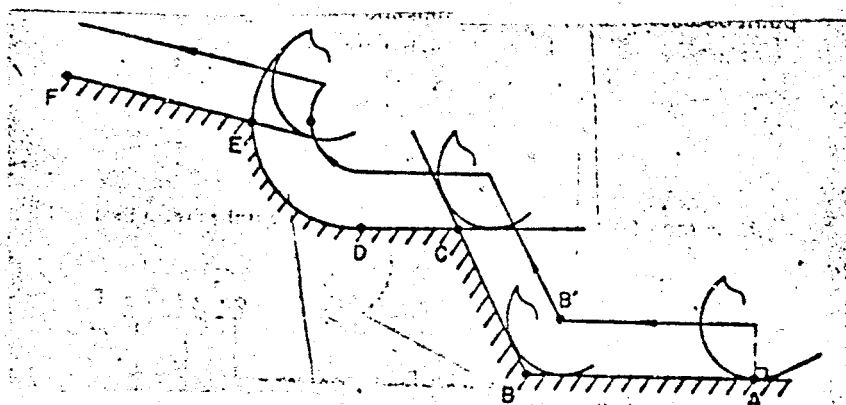


Рис.136. Траектория режущего инструмента с использованием компенсации радиуса закругления вершины резца.

15.1.4. Программирование

Командные коды программирования G, M и T используются для введения функции компенсации радиуса закругления вершины резца и более подробно рассматриваются ниже.

15.1.4.1. Коды G

- G40 - (используется для отмены режима компенсации радиуса закругления вершины резца).
- G41 - компенсация радиуса закругления вершины резца (слева). Используется, если резец перемещается влево по заготовке.
- G42 - правая компенсация радиуса закругления резца. Используется, если резец движется вправо по заготовке.

Примечание 1. Правая или левая сторона заготовки определяются в зависимости от направления движения резца.

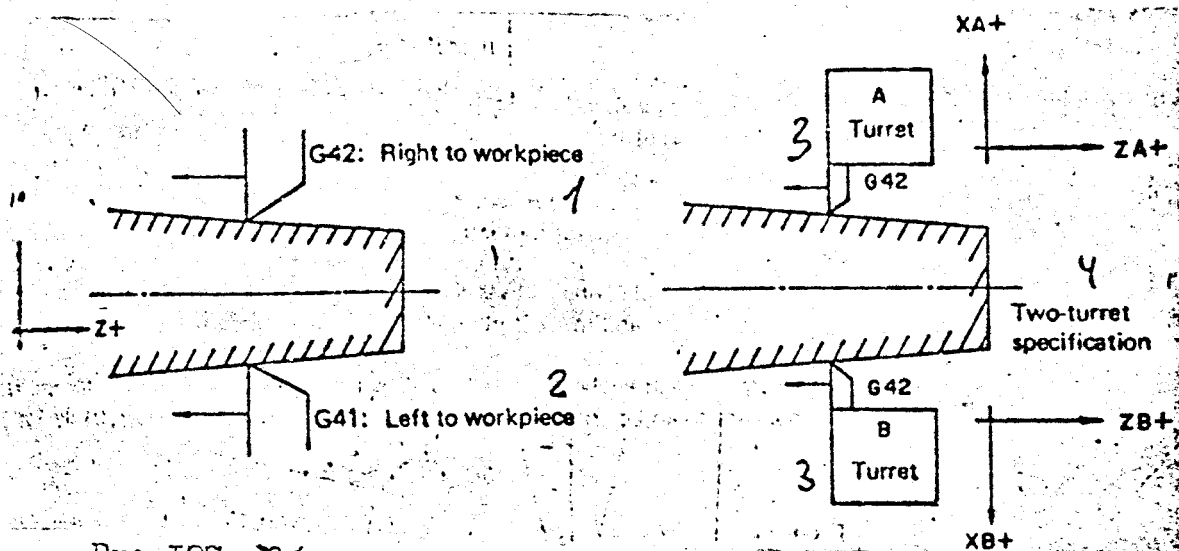


Рис.137. Обозначения для G41 и G42:

1 - вправо по заготовке, 2 - влево по заготовке, 3 - револьверная головка, 4 - обработка с помощью двух револьверных головок

Коды G41 и G42 выбираются с учетом координатной системы, принятой на станке, поэтому выбор должен производиться, как показано на рис.138, для токарных станков, оснащенных координатной системой, в которой положительное направление вдоль оси X направлено в сторону оператора. Это относится к токарным станкам серии 1B или 1S.

Обозначение кодов G для токарных станков, оснащенных задней револьверной головкой (с зеркальным отражением функций), может осуществляться по той же схеме, что и для станков с двумя револьверными головками. Однако применяется обратное обозначение для револьверных головок A и B.

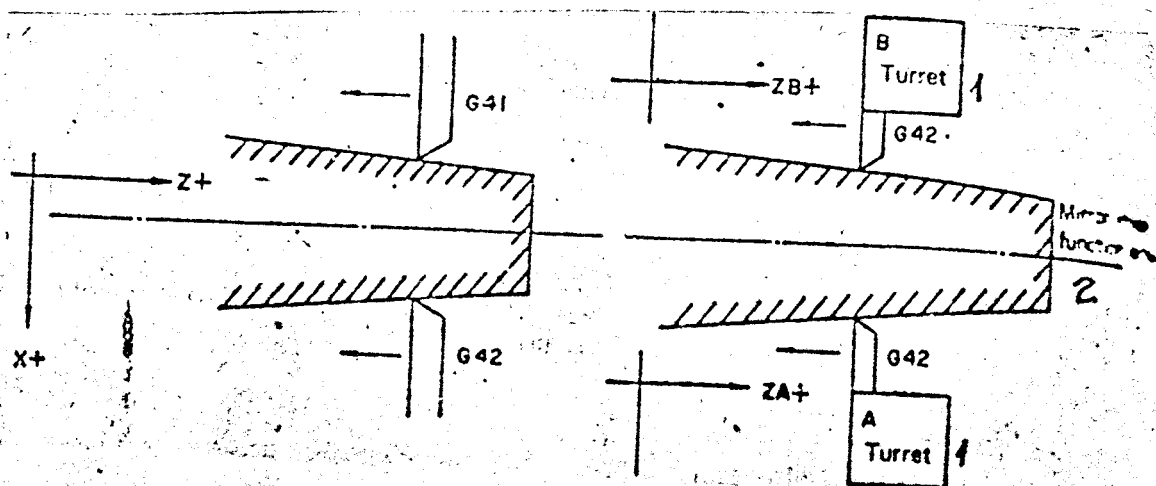


Рис.138:

1 - револьверная головка, 2 - зеркальное отражение

15.1.4.2. Коды T

Шесть цифровых знаков после знака адресования T указывают номер компенсации радиуса закругления вершины резца, номер инструмента и номер сдвига инструмента.

Для ввода данных сдвига с помощью переключателей клавиатуры предусмотрено 32 пары данных компенсации по радиусу и по сдвигу резца. (01-32). Для номера резца предусмотрены коды 01-16.

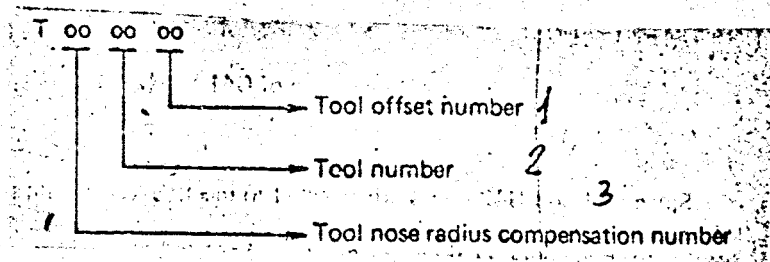


Рис. 139:

1 - номер сдвига резца, 2 - номер резца, 3 - номер компенсации радиуса закругления при вершине резца

15.1.4.3. Пример программы

Ниже приводится программа обработки контура, представленного на рис. 136.

Размерные команды, команды на различные функции и прочие команды, используемые для обработки указанного контура без компенсации радиуса закругления вершины резца, программируются по известной методике, однако добавляются коды G для вызова режима компенсации радиуса закругления вершины резца и команды T с указанием номера компенсации радиуса закругления резца в новой программе.

```

N000 ... ..
N000 G42 G00 Xa Za T010203
N000      G01      Zb F
N000      Xc Zc
N000      Zd
N000      G02 Xe Ze I K
N000 G40 G00 Xf Zf
N000 ... ..
    
```

Внимание. Если точка пересечения не найдена при использовании команд с размерными словами, то выводится тревожная сигнализация.

15.1.5. Данные компенсации радиуса закругления вершины резца.

Как показано в предыдущем разделе, программа введения компенсации радиуса закругления вершины резца значительно упростилась. Однако для точной окончательной обработки деталей с помощью этой функции необходимо точно измерить радиус закругления вершины резца, а измеренное значение правильно ввести в запоминающее устройство ЧПУ.

15.1.5.1. Измерение радиуса закругления вершины резца

Измерить центр окружности закругления вершины резца по отношению к контрольной точке вершины резца, находящейся в воображаемой точке вершины резца, используемой для предварительной настройки резца (см. рис.140).

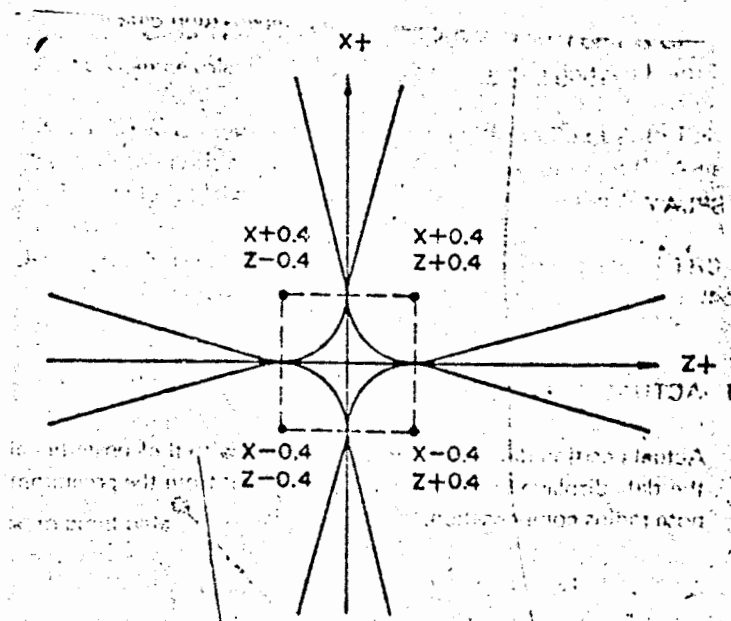


Рис.140. Величина компенсации для радиуса закругления вершины, равного 0,4 мм

Если в системе управления отсутствует функция компенсации закругления вершины резца, то контролируется перемещение вдоль координатных осей с тем, чтобы контрольная точка вершины резца перемещалась бы по запрограммированной траектории.

15.1.5.2. Настройка данных компенсации

Перед тем, как включить цикл обработки, необходимо ввести данные измерения для компенсации в память ЧПУ.

Это осуществляется по аналогии с введением данных сдвига резца с помощью переключателей клавиатуры на панели управления.

Пример

Для введения данных компенсации радиуса закругления вершины резца $X + 0,35$ и $Z + 0,35$ для инструмента №2 выполнить перечисленные ниже операции.

1) Повернуть клавишу "Выбор режима" в положение "Включено". (Данные для инструмента).

2) На станках с двумя револьверными головками или с задней револьверной головкой нажать клавишу F6 "Револьверная головка", так что на дисплее появится обозначение револьверной головки А.

3) Нажать клавишу управления курсором и подвести курсор к координате X компенсации радиуса закругления вершины резца.

4) После нажатия клавиши F1 (настройка) ввести 0,35 с помощью десяти клавишей и клавиши "Запись-исполнение". При этом в память ЧПУ вводятся данные компенсации радиуса закругления вершины резца.

Подвести курсор к знаку Z для O1. Ввести 0,35 с помощью десяти клавиш после нажатия клавиши F1 "настройка", а затем нажать клавишу "запись-исполнение". При этом в память ЧПУ вводятся данные компенсации радиуса закругления резца по оси Z.

Цифровые данные, введенные с помощью клавиатуры, появляются на экране дисплея при каждом нажатии цифровой клавиши, что облегчает проверку вводимых данных.

Максимальная величина данных компенсации радиуса закругления резца составляет $\pm 999,999$ мм.

15.1.6. Индикация

На дисплее отображаются различные данные (см. инструкцию по эксплуатации системы OSP 5000C). Ниже рассматриваются данные, используемые в режиме компенсации радиуса закругления вершины резца.

15.1.6.1. Фактическое положение

Данные фактического положения отображаются на экране по аналогии с обычной системой управления. Однако эти данные могут отличаться от запрограммированных в связи с используемой компенсацией радиуса закругления вершины резца.

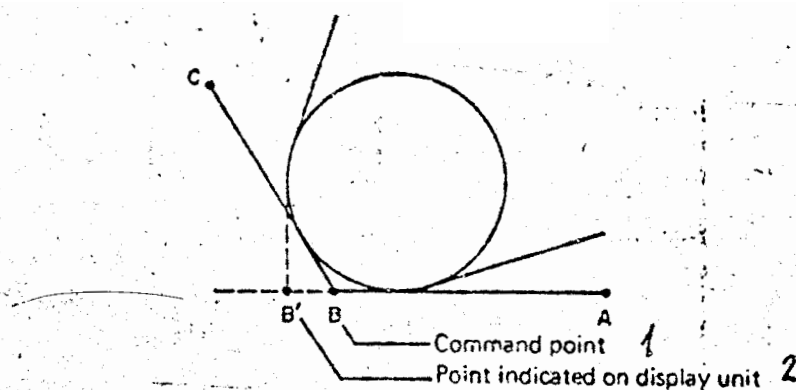


Рис. I41:

1 - точка выдачи команды, 2 - точка, отображаемая на дисплее

15.1.6.2. Дисплей с тревожными сообщениями

При выдаче тревожного сообщения в связи с компенсацией радиуса закругления вершины резца загорается лампа тревожной сигнализации в секции "Состояние дисплея", а тревожные сообщения на экране дисплея соответствуют содержанию тревожной сигнализации.

15.1.7. Прочие замечания

15.1.7.1. Условия включения функции компенсации радиуса закругления вершины резца

1) Считывание буфера, состоящего из двух кадров

Для перемещения режущего инструмента из точки А в точку В в системе управления должно быть известно положение точки В. Расчет координат точки В производится на основании расчетных данных для точки В с учетом точки С, которая следует за точкой В.

Если начинается перемещение вдоль оси из точки А в точку В, то в системе управления в первую очередь считываются данные, относящиеся к точке В, а затем к точке С. Затем осуществляется перемещение в точку В. Таким образом, в системе управления происходит считывание данных положения двух последовательных точек для позиционирования вдоль оси, т.е. считывание буфера, состоящего из двух кадров.

Для индикации данных, относящихся к буферу из двух кадров на экране отображаются данные положения для точки, в которой осуществляется следующее позиционирование (см. рис. I42).

Примечание. Система управления OSP 5000I фактически работает в режиме с тремя буферами.

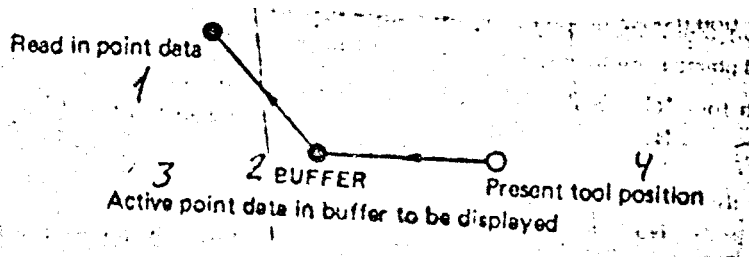


Рис. I42

1 - считывание данных, относящихся к точке, 2 - буфер, 3 - активные данные, относящиеся к точке в буфере для индикации, 4 - внешнее положение режущего инструмента

2) Использование компенсации радиуса закругления вершины резца на моделях станков с задней револьверной головкой.

Для станков, оснащенных задней револьверной головкой, производится ввод функции компенсации радиуса закругления вершины резца. Однако, невозможен выбор кода G для определения головки, например, G13 или G14 во время действия указанной функции. Если требуется выбор револьверной головки, в то время как действует функция компенсации радиуса закругления вершины резца, то необходимо отключить режим за счет введения кода G40 до выдачи команды G13 или G14.

3) Возможность использования функции закругления вершины резца в режиме автоматического программирования IAPe

Во время работы в режиме IAP имеется возможность введения функции компенсации радиуса закругления вершины резца.

4) Возможность компенсации радиуса закругления вершины резца при одновременной обработке по четырем координатам.

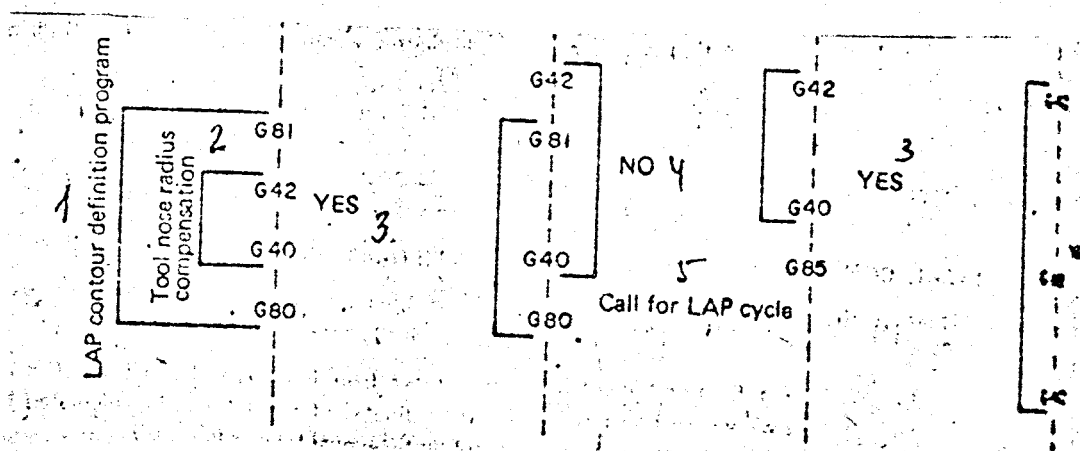


Рис. I43:

1 - программа определения контура в режиме IAP, 2 - компенса-

для радиуса закругления вершины резца, 3 - да, 4 - нет, 5 - вызов цикла IAP

На станках с двумя револьверными головками в указанном варианте возможен ввод функции компенсации радиуса закругления резца, причем в программах для отдельных револьверных головок должна быть предусмотрена независимая функция компенсации радиуса закругления вершины резца.

15.1.7.2. Режим компенсации радиуса закругления вершины резца для траектории центра радиуса R закругления вершины резца

Для осуществления перемещения, показанного на рис. I44, нужно, чтобы в программе для режима компенсации радиуса закругления вершины резца была предусмотрена траектория окружности с радиусом R закругления вершины резца.

N1	G42	X1	Z1
N2		X2	Z2
N3		X3	Z3
N4	G41	X4	Z4
N5		X5	Z5

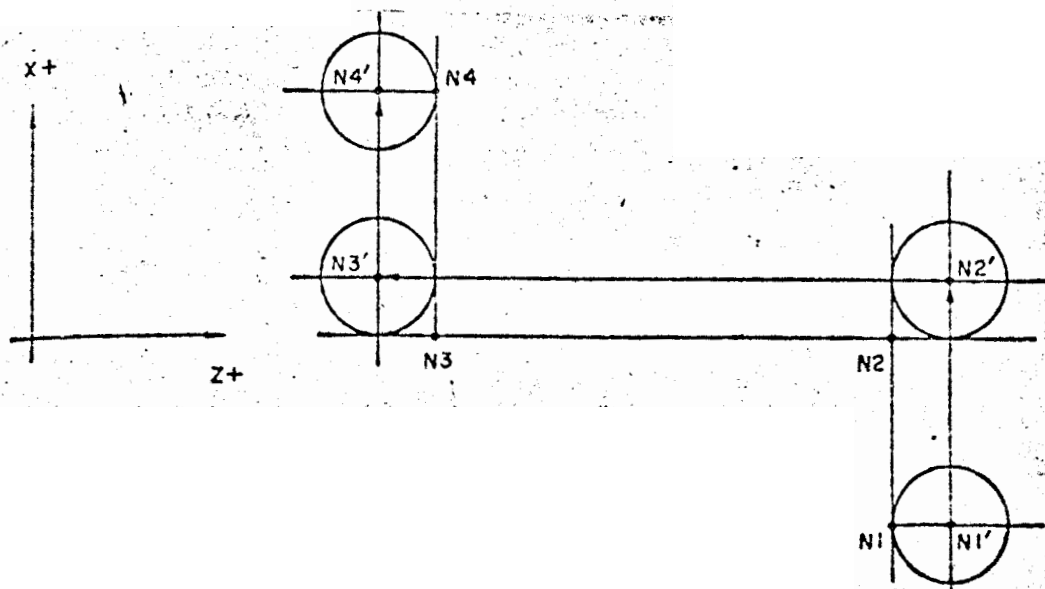


Рис. I44

1) Для получения точки №2', когда центр радиуса R закругления вершины резца находится в точке №1' выполнить следующее:

а) Провести прямую линию, параллельную направлению перемещения резца №1-№2 со сдвигом в заданном направлении (направо, если указан код G42) на величину, равную компенсации радиуса закругления вершины резца. При этом получается прямая линия, проходящая через точки №1' и №2'.

б) Провести прямую линию, параллельную направлению движения резца №2-№3, со сдвигом в указанном направлении (вправо или выше линии №2-№3, так как код G42 определяет режим компенсации) на величину компенсации радиуса закругления вершины резца. При этом получается прямая линия, проходящая через точки №2' и №3'.

в) Центр радиуса закругления вершины R для командной точки №2 соответствует точке пересечения указанных двух прямых линий.

Центр радиуса R закругления вершины резца перемещается из точки №1' в точку №2'.

2) Для нахождения точки №3' выполнить следующее:

а) Провести прямую линию, параллельную направлению движения резца №2-№3, со сдвигом в заданном направлении (вправо или выше линии №2-№3, так как код G42 определяет режим компенсации) на величину компенсации радиуса закругления вершины резца. При этом получается прямая линия, проходящая через точки №2' и №3'.

б) Провести прямую линию, параллельную направлению движения резца №3-№4 со сдвигом в заданном направлении (влево, если указан код G41) на величину компенсации радиуса закругления вершины резца. При этом получается прямая линия, проходящая через точки №3' и №4'.

в) Центр радиуса R закругления вершины резца для командной точки №3 соответствует точке пересечения двух указанных прямых линий.

Центр радиуса R закругления вершины резца перемещается из точки №2' в точку №3'.

3) Для нахождения точки №4' выполнить следующее:

Повторить операции, предназначенные для нахождения точек №3, №4 и №5.

15.2. Использование программирования

15.2.1. Коды G41 и G42

Коды G41 и G42 используются для введения режима компенсации радиуса закругления вершины резца. Часто эти коды путаются при программировании для обработки детали, поэтому в данном разделе рассматриваются различия между этими кодами.

Код G41

Этот код компенсации радиуса закругления вершины резца используется в том случае, когда режущий инструмент перемещается к левой стороне заготовки с учетом направления перемещения.

Внимание. Термины "влево" и "вправо", используемые для обозначения режима компенсации радиуса закругления вершины резца, относятся к принятой системе координат. Таким образом, на станках моделей 1Н и 1S код G41 используется в том случае, когда режущий инструмент перемещается к левой стороне заготовки, если смотреть на револьверную головку с низкой стороны.

Код G42

Этот код компенсации радиуса закругления вершины резца используется в том случае, когда режущий инструмент перемещается к правой стороне заготовки с учетом направления перемещения.

Внимание. Термины "влево" и "вправо", используемые для обозначения режима компенсации радиуса закругления вершины резца, относятся к принятой системе координат. Таким образом, на станках моделей 1Н, 1S код G42 используется в том случае, когда режущий инструмент перемещается к правой стороне заготовки, если смотреть на револьверную головку с нижней стороны.

15.2.2. Характеристики ввода режима компенсации радиуса закругления вершины резца

15.2.2.1. Общее описание

```
N0 G00 X0 Z0  
N1 G42 X1 Z1 T000000  
N2 G01 X2 Z2
```

В приведенном ниже примере используются рассмотренные выше программы для осуществления обработки по наружному диаметру. При этой обработке применяется положительная величина компенсации радиуса закругления вершины резца по каждой оси (R_x^+ , R_z^+).

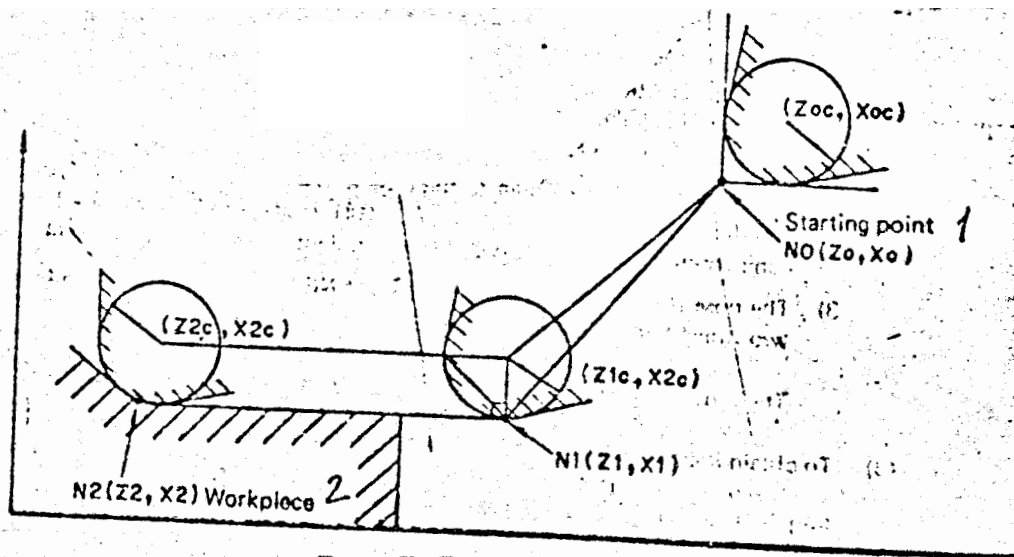


Рис. I45:

I - начальная точка, 2 - заготовки

Без функции компенсации радиуса закругления вершины реза позиционирование выполняется таким образом, чтобы контрольная точка вершины реза находилась бы точно в программируемых координатах. На начальном этапе введения режима компенсации радиуса закругления вершины реза с помощью кодов G41 или G42 позиционирование выполняется таким образом, чтобы окружность при вершине реза контактировала бы с сегментом, проходящим через координаты, соответствующие программированию в кадре, который содержит коды G41 и G42, и в следующем кадре. Такое перемещение по осям называется начальным.

На начальном режиме компенсации радиуса закругления вершины реза возможен сдвиг обеих координатных осей X и Z даже если в кадре содержится только одно размерное слово X или Z (см. пример программы I).

I5.2.2.2. Примеры программ

I) Пример I

```

N1 G00 X100 Z100 S1000 T010101 M03
N2 G42 X80
N3 G01 Z50 F0.2
    
```

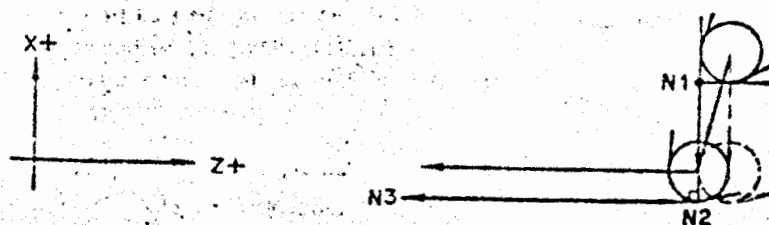


Рис. I46:

Программист может ожидать, что перемещение оси соответствует пунктирным линиям, так как в кадре №2 содержится только слово X, однако фактическая траектория режущего инструмента, сформированная на начальном этапе режима компенсации радиуса закругления вершины резца, соответствует сплошным линиям на рис. I46.

2) Пример 2

Ниже приводится желательная программа ввода режима компенсации.

```
N1 G00 X100 Z100 S1000 T010101 M03  
N2 X80  
N3 G42 Z90  
N4 G01 Z50 F0.2
```

При этой программе в кадре G42 содержится только слово, а точки №2, №3 и №4 располагаются на одной и той же прямой линии.

Примечание. Режим работы при введении функции компенсации радиуса закругления вершины резца должен определяться кодом G00 или G01. В противном случае вводится тревожное сообщение.

3) Пример 3.

Если на начальном этапе режима компенсации радиуса закругления вершины резца отсутствуют оба слова X и Z или если в кадре начального режима указана такая же точка, в которой в настоящий момент установлены координатные оси, то позиционирование выполняется таким образом, чтобы окружность вершины резца контактировала бы с сегментом, проходящим через координаты, установленные программой, и координаты в очередной последовательности. Сдвиг для компенсации радиуса закругления вершины резца осуществляется в последовательности, представленной на рис. I47.

На основании указанной на рис. I47 программы радиус окружности вершины резца устанавливается таким образом, что он контактирует с сегментом №2 - №3 и №3 - №4. Таким образом, в кадрах с командами после №3 формируется последовательность, необходимая для режима компенсации радиуса закругления вершины резца.

Примечание. В том случае, когда в последующем кадре ука-

зывается та же точка, что и в начальном кадре, то тревожное сообщение выдается, если в последовательных двух кадрах не содержатся размерные слова X и Z.

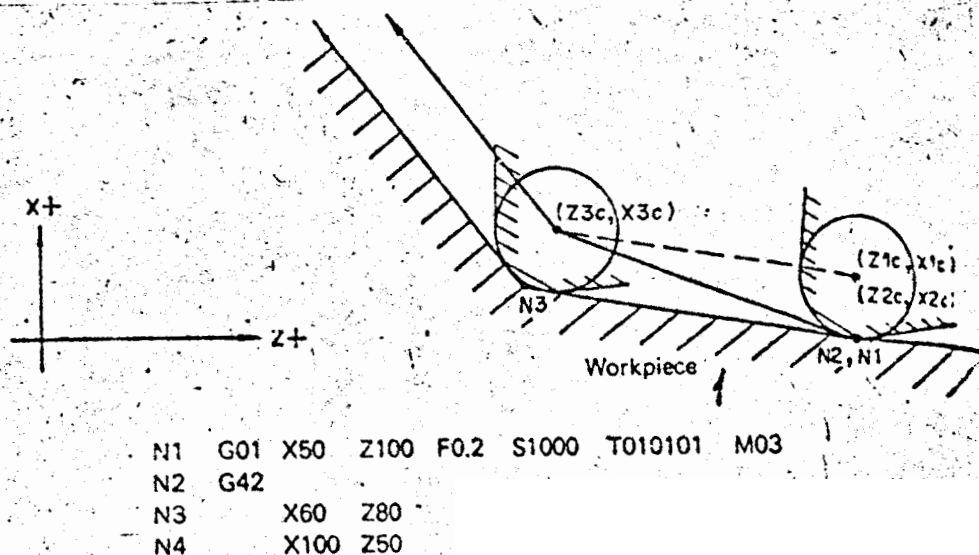


Рис. I47:

I - заготовка,

15.2.3. Характеристики режима компенсации радиуса закругления вершины резца

Функция компенсации радиуса закругления вершины резца позволяет автоматически компенсировать изменение радиуса вершины резца при непрерывном резании.

Подобная компенсация осуществляется автоматически, поэтому при упрощенной подготовке программ нужно учитывать некоторые ограничения, если эта функция компенсации радиуса закругления вершины резца не используется.

15.2.3.1. Резание с переходом от одной прямой линии к другой прямой линии

I. Средняя точка на прямой линии

Если указывается средняя точка на прямой линии, то необходимо обеспечить ее полными командами. В остальном программирование осуществляется по обычной схеме.

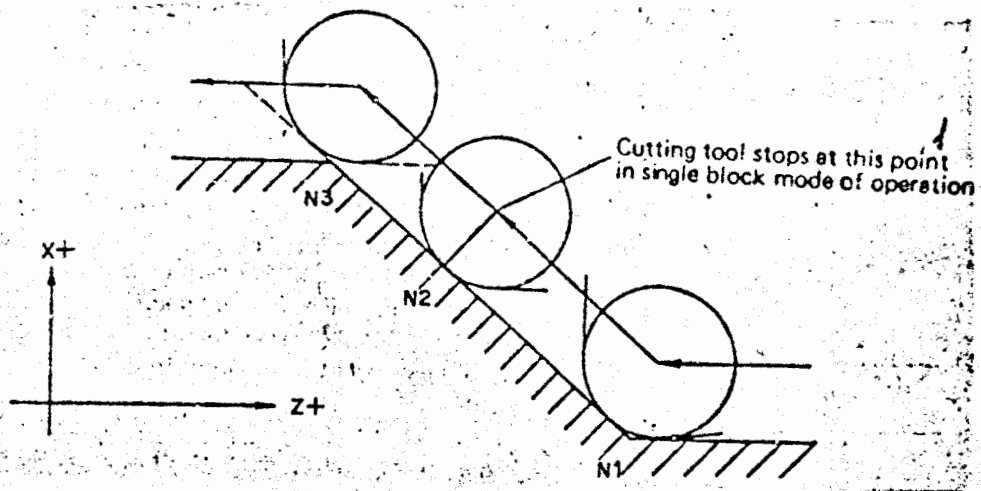


Рис. I48:

I - режущий инструмент останавливается в указанной точке при режиме обработки с единичными кадрами

Если точка №2 находится на линии №1 - №3, то режущий инструмент устанавливается таким образом, что окружность вершины режущего инструмента контактирует с точкой №2.

2) Возврат вдоль прямой линии

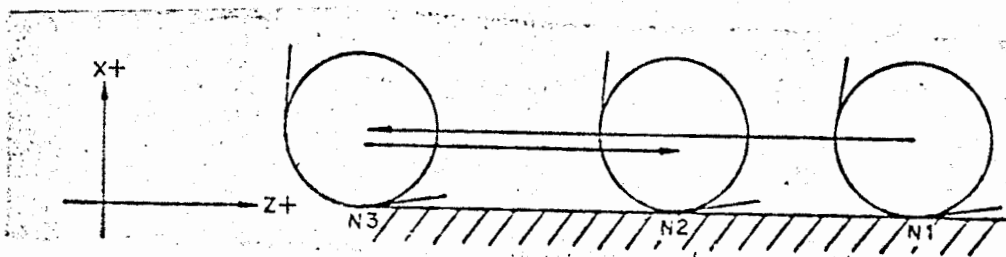


Рис. I49

При таком перемещении вдоль оси не возникает никаких затруднений, если в программе не используется функция компенсации радиуса закругления вершины резца. Однако при использовании этой функции необходимо принимать меры предосторожности.

Пример

N1	G42	G01	X1	Z1
N2			X2	Z2
N3	G41		X3	Z3

В этом примере выдаются команды, относящиеся к точкам

№ 2 и № 3 в то время, как режущий инструмент находится в точке №1.

Если режущий инструмент приближается из точки №1 к точке №2, то программируется код G42, так как движение режущего инструмента осуществляется на правой стороне заготовки с учетом направления приближения режущего инструмента. Однако в случае движения возврата резца от точки №2 к точке №3 режущий инструмент находится на левой стороне заготовки с учетом направления движения режущего инструмента, так что вместо кода G42 используется код G41.

После выполнения указанной выше программы можно получить фактическую траекторию движения окружности при вершине режущего инструмента, используя для этого перечисленные ниже операции.

а) Центр окружности при вершине резца (№2') в точке №2 можно найти следующим образом:

- Найти линию, параллельную прямой линии №1 - №2 с величиной сдвига (G42) на расстояние радиуса закругления вершины резца, действительное для точки №1.

- Найти линию параллельную прямой линии №2 - №3 при сдвиге (код G41) на величину радиуса закругления вершины резца, действительную в точке №2.

- Центр окружности при вершине резца находят в точке пересечения двух прямых линий, построенных в соответствии с приведенными выше указаниями. Если эти две линии параллельны друг другу, то точку пересечения в таком случае найти не удастся. Для подобного случая в системе управления предусмотрена специальная характеристика, позволяющая выполнять позиционирование таким образом, чтобы окружность при вершине резца контактировала бы с точкой №2. При этом траектория окружности при вершине резца смещается при сдвиге режущего инструмента от точки №1 к точке №2 и определяется через линию №1' - №2'.

б) Центр окружности при вершине резца (№3') в точке №3 получается в соответствии с приведенными выше операциями.

Подобным способом рассмотренную выше программу можно использовать для возврата режущего инструмента вдоль прямой линии при условии введения функции компенсации радиуса закругления вершины резца.

Примечание. Если любая из указанных трех точек не имеет точного расположения на одной и той же прямой линии, то может

произойти существенное отклонение от предполагаемой траектории.

З) Две линии образуют острый угол

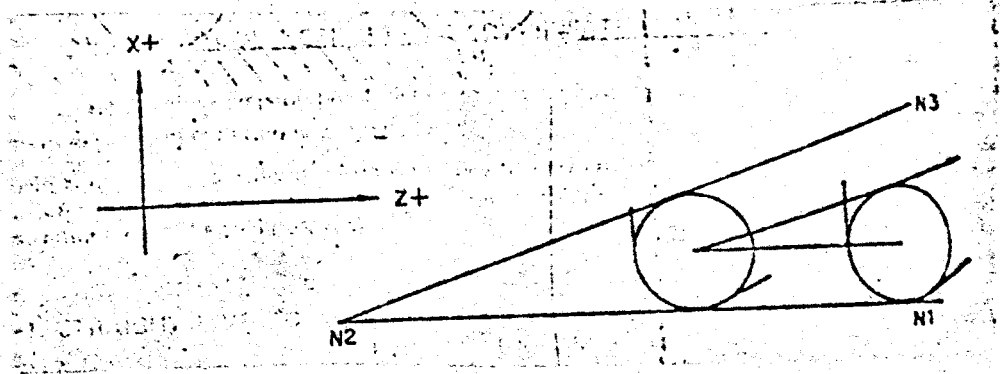


Рис. 150

На основании графика на рис. 50 можно показать, что режущий инструмент не может попасть в точку №2, если предполагается сдвиг из точки №1 в точку №2. Это объясняется тем, что режущий инструмент может перемещаться только в точку, в которой окружность при вершине резца касается линии №2 - №3.

Предшествующий пример относится к случаю усложненного программирования. Ниже приводится еще один пример.

```
N1 G42 X100 Z100 F0.2 S1000 T010101 M03
N2 Z50
N3 G00 X300 Z300 M05
```

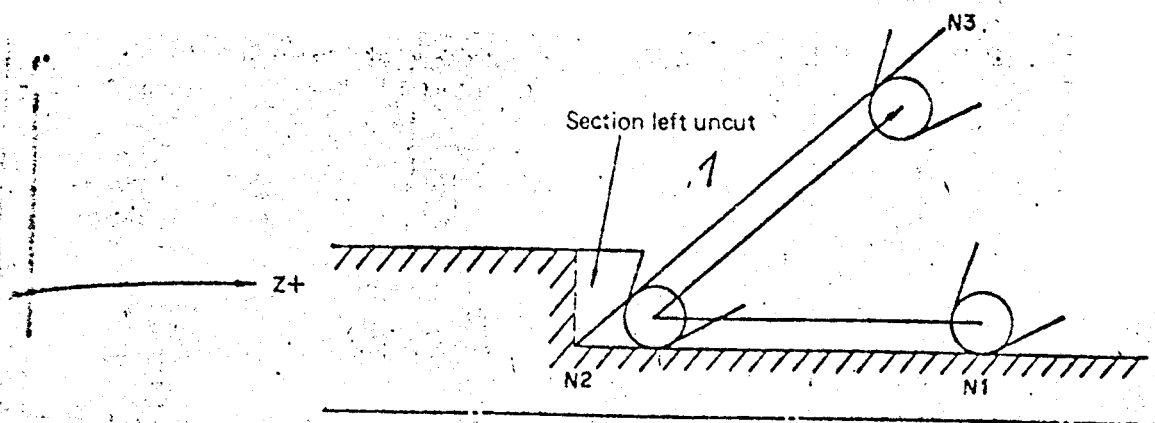


Рис. 151:

I - Неподрезанный участок

Программист намечает подрезание до точки №2 (например 50) с учетом небольшого необработанного участка в зоне острого угла в связи с радиусом закругления вершины резца. Однако в отличие от этого намерения режущий инструмент оставляет значительный необработанный участок, так как он останавливается до достижения заданной точки.

Для усовершенствования подобной программы нужно ввести в программу еще одну точку, представленную на рис.152.

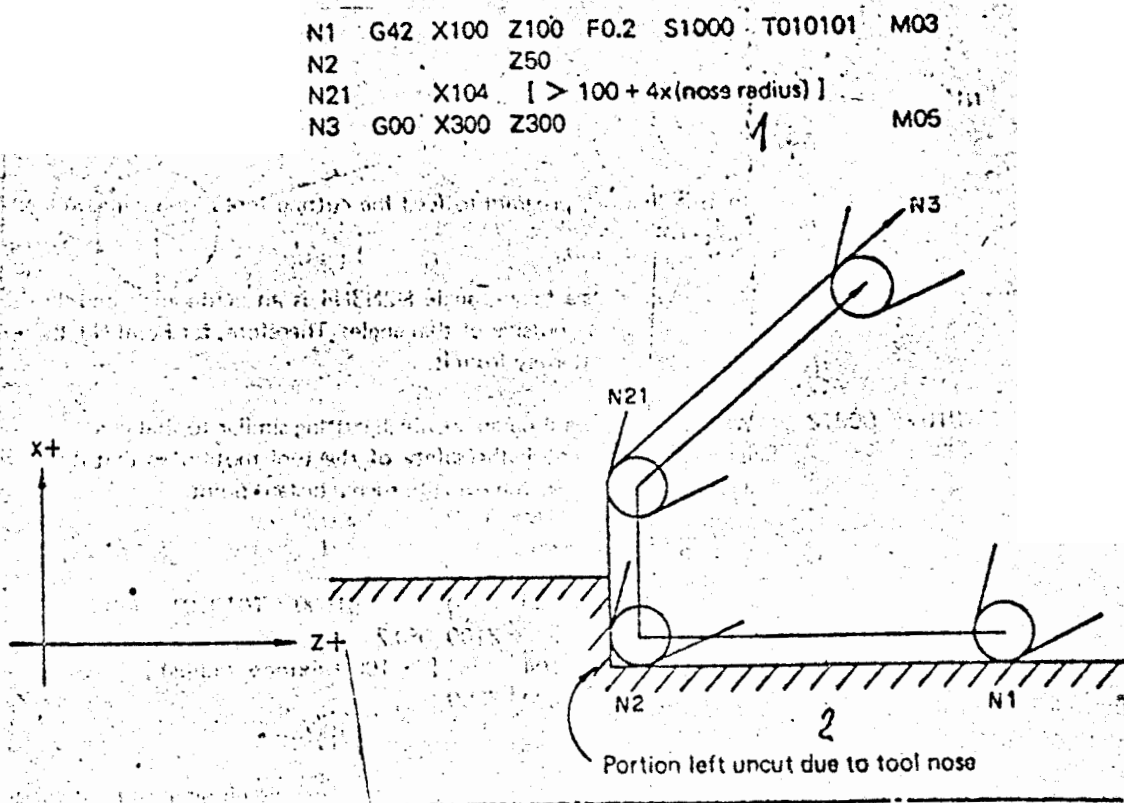


Рис.152:

1 - радиус закругления вершины резца, 2 - необработанный участок в связи с закруглением вершины резца.

В соответствии с усовершенствованной программой получается траектория инструмента, представленная на рис.152, причем обработка соответствует ожидаемой схеме за исключением небольшого неподрезанного участка, оставшегося в связи с закруглением вершины резца.

Для перемещения резца вдоль оси X в положительном направлении в соответствии с кадром №2I, слово X, как правило, имеет значение, которое в 4 раза превышает радиус закругления вершины. Это объясняется необходимостью удвоения радиуса закругления вершины резца для построения окружности вершины резца. Кроме того, когда слово X выражается через диаметр, то его значение нужно увеличить в два раза. Таким образом, цифровое значение подобного слова X должно в 4 раза превышать радиус закругления вершины резца.

Если используется значение меньше, чем требуемое, то режущий инструмент может перемещаться в противоположном направлении к точке № 2I, что вызовет подрезание поверхности.

4) Две линии образуют тупой угол

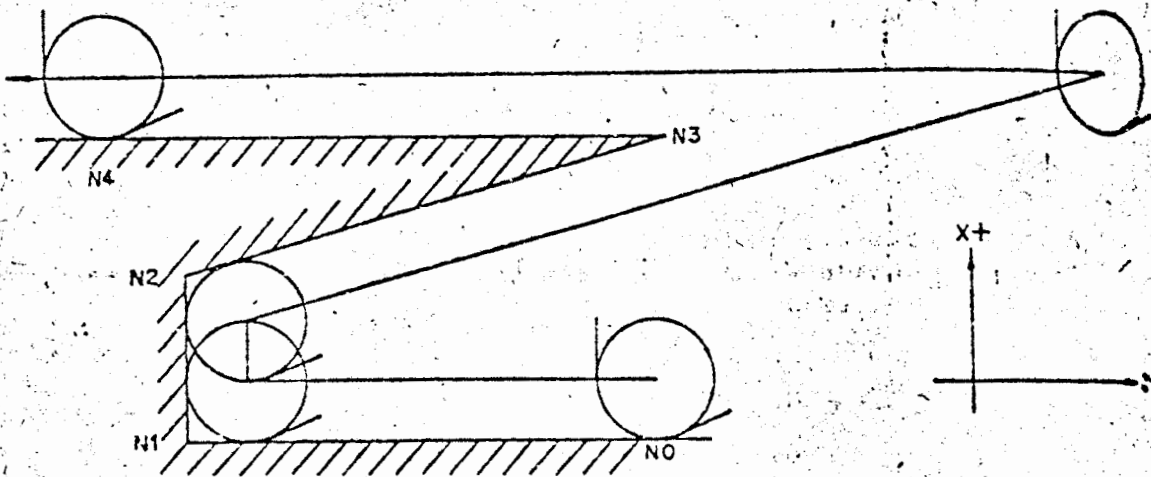


Рис. I53

В данном разделе используется программа для перемещения режущего инструмента по траектории №0 - №1 - №2, -№3, -№4

Как показано на рис. I53, угол №2- №3 - №4 является острым и режущий инструмент движется по внешней периферии этого угла. Таким образом, для точки №3 режущий инструмент значительно удаляется от соответствующей точки.

При подготовке программы, требующей обработки вдоль аналогичного контура, необходимо проверить безопасность перемещения режущего инструмента с тем, чтобы он не задевал за препятствия при сдвиге в столь удаленную точку.

Пример

```
N0 G00 X100 Z300 S1500 T010101 M03
N1 G01 Z100 F0.2
N2 X104 [> 100 + 4x(nose radius)] 1
N3 G00 X200 Z300
N4 G01 Z50 S1000
```

I - радиус закругления вершины резца

Весьма выгодно усовершенствовать программу таким образом, чтобы исключить последовательное позиционирование в удаленной точке с помощью команд в кадре №3.

Если угол №2 - №3 - №4 не является острым, то подобные затруднения не возникают. Для исключения острых углов из обрабатываемого контура единственным возможным решением является введение укороченного прямого участка №3 №4.

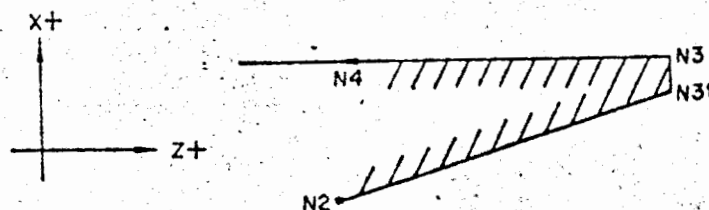
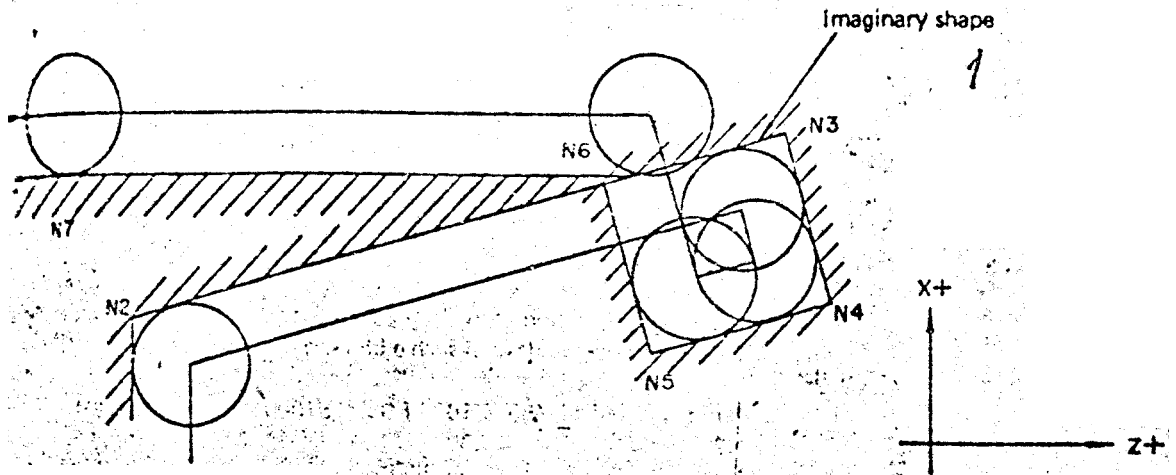


Рис. I54

В некоторых случаях подобное изменение невозможно, поэтому для обработки острого угла без сдвига режущего инструмента в удаленную точку выполнить операции, перечисленные ниже.

На рис. I55 приводится пример усовершенствованной программы.

В соответствии с этой программой режущий инструмент перемещается вдоль воображаемого прямоугольника №3 - №4 - №5 - №6. При этом оператор имеет возможность оценивать отклонение режущего инструмента от запрограммированного контура. Следует отметить, что одна сторона воображаемого прямоугольника должна более, чем в два раза превышать радиус закругления вершины резца.

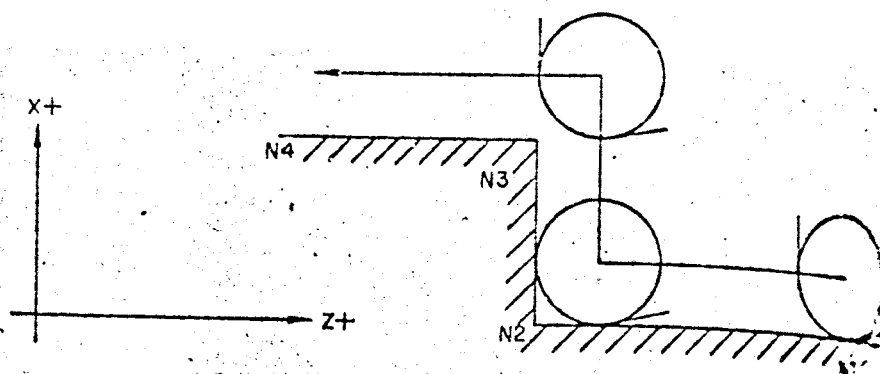


N0	G42	G00	X100	Z300		S1500	T010101	M03
N1		G01		Z100	F0.2			
N2			X104					
N3		G00	X200.48	Z301				
N4			X198.48	Z301.24	F1			
N5			X198	Z300.24				
N6		G01	X200	Z300				
N7				Z50	F0.2	S1000		

Рис. I55:

I - воображаемый контур

5) Две линии образуют прямой угол



N1	G42	G01	X100	Z100	F0.2	S1000	T010101	M03
N2				Z60				
N3			X150					
N4				Z20				

Рис. I56

Особых затруднений в подобном случае не возникает.

б) Команды , связанные с одинаковыми точками

а) Если предусмотрен кадр , не содержащий команд на перемещение вдоль оси при действующем режиме компенсации радиуса закругления вершины резца, то траектория окружности вершины резца будет такой же, как в случае, когда подобный кадр не предусмотрен.

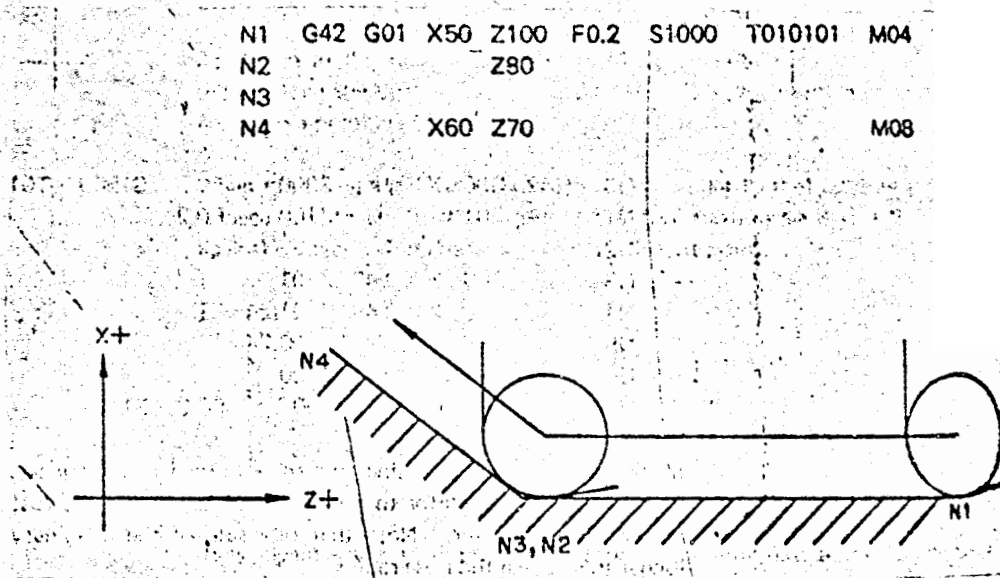


Рис. I57

б) Если предусмотрены два или несколько кадров, не содержащих команд на перемещение вдоль оси, или если команда относится к той же точке, что и в предшествующей последовательности, то осуществляется повторяемость режима компенсации радиуса закругления вершины резца.

В этом случае осуществляется движение вдоль оси, при котором окружность вершины резца контактирует с запрограммированным контуром в координатной точке, соответствующей программе. При считывании кадра с командами, содержащими размерные слова X или Z режущий инструмент возвращается в положение, соответствующее правильной компенсации.

Программа I

N1	G41	G01	X50	Z100	F0.2	S100	T010101	M04
N2				Z80				
N3				Z80				
N4			X60	Z70				M08

При этой программе может произойти подрез, указанный на

рис. I58.

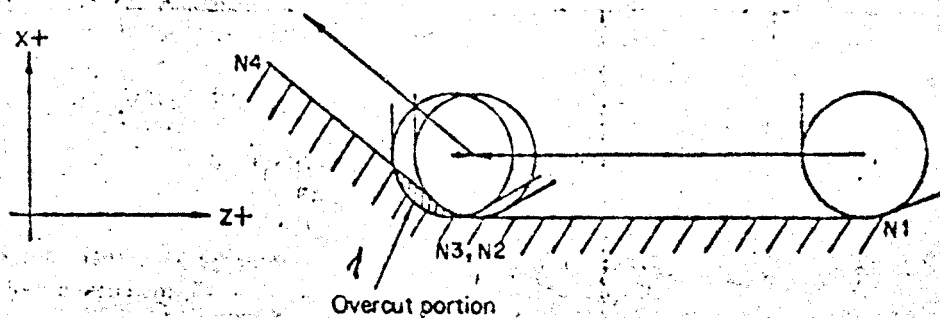


Рис. I58:

I - участок с подрезом

В зависимости от обрабатываемого контура не возникает подобное неожиданное перемещение, связанное с подрезанием, когда используется программа 2.

Программа 2

```
N1 G42G01 X50 Z100 F0.2 S100 T010101 M04  
N2 Z80  
N3 Z80 M08  
N4 X40 Z70
```

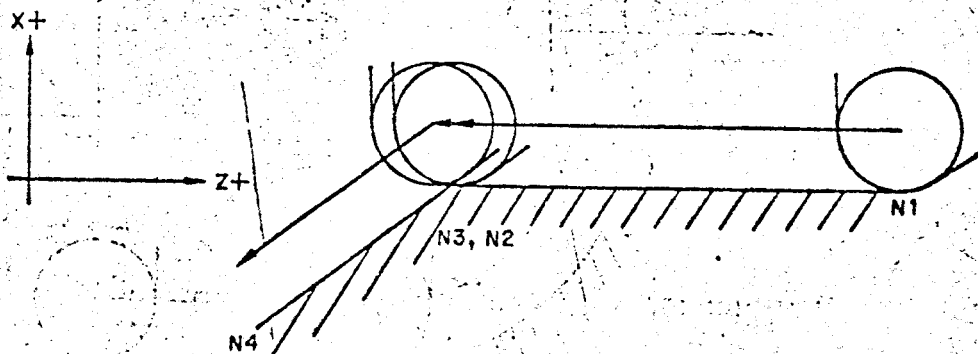


Рис. I59

15.2.3.2. Резание с переходом от прямой линии к дуге (резание с переходом от дуги к прямой линии)

1) Дуга в пределах одного квадранта

В программе, предусматривающей непрерывное перемещение режущего инструмента от прямой линии к дуге, это перемещение осуществляется по аналогии с движением от одной прямой линии к другой прямой линии (см. рис. I60).

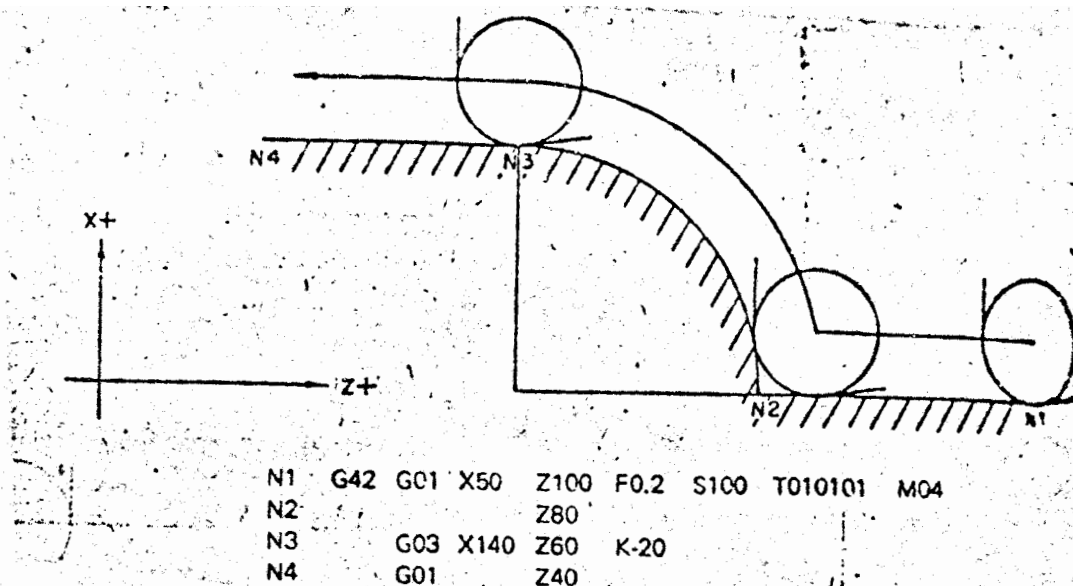


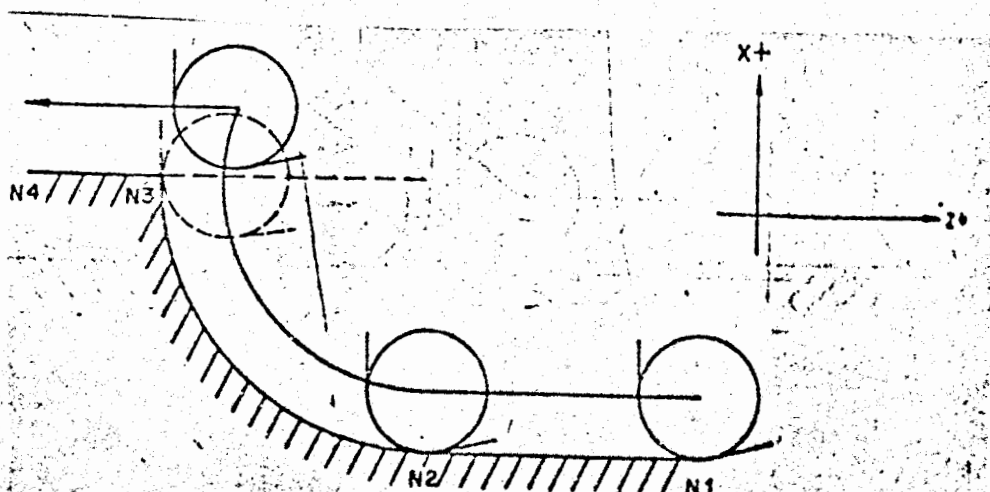
Рис.160

Положение режущего инструмента в точке №2 определяется таким образом, чтобы окружность вершины резца соответствовала бы целевому положению для контакта с линией №1 - №2 и с дугой №2 - №3. В точке №3 позиционирование режущего инструмента осуществляется аналогичным образом.

Если режущий инструмент перемещается от точки №3 к точке №4, то режим резания меняется от кругового интерполирования к линейному. В случае разрывов в зоне точки №3 производят расчет траектории резца с учетом режима компенсации радиуса закругления вершины режущего инструмента, причем этот расчет вводится в систему управления. Если разрыв сохраняется, то выдается тревожное сообщение, а работа станка прекращается.

2) Дуга в двух квадрантах

- а) Радиус дуги превышает удвоенный радиус закругления вершины резца.



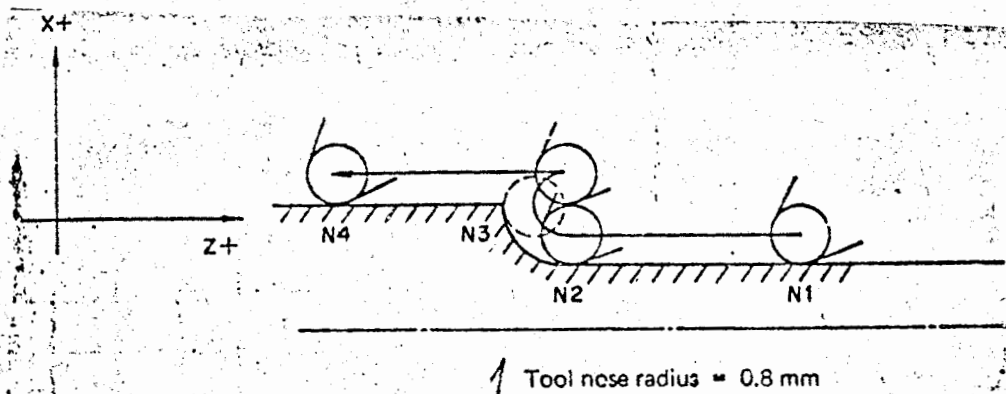
```

N1- G42 G01 X50 Z100 F0.2 S100 T010101 M04
N2      Z80
N3      X140 Z60 I20
N4      G01- Z40
    
```

Положение резца, определяемое командами в кадре №2, соответствует точке, в которой окружность вершины резца контактирует с точкой №2.

В последовательности №3 позиционирование режущего инструмента осуществляется таким образом, что он контактирует с продолжением прямой линии №2 - №3 и с продолжением дуги №3 - №4.

б) Радиус дуги равен удвоенному радиусу закругления вершины резца.



```

N1 G42 G01 X100 Z100 F0.2 S1000 T010101 M04
N2      Z80
N3      G02 X103.2 Z78.4 I1.6
N4      G01 Z40
    
```

Рис. I62:

I - радиус закругления вершины резца

Если радиус программируемой дуги равен удвоенному радиусу окружности при вершине резца (радиус закругления вершины резца), то режущий инструмент располагается в точке, в которой окружность вершины резца контактирует с продолжением дуги №2 - № 3 и с продолжением прямой линии №3 - № 4 после выполнения команд в кадре №3. При этом режущий инструмент размещается непосредственно над точкой №2.

в) Радиус дуги меньше удвоенного радиуса при вершине резца.

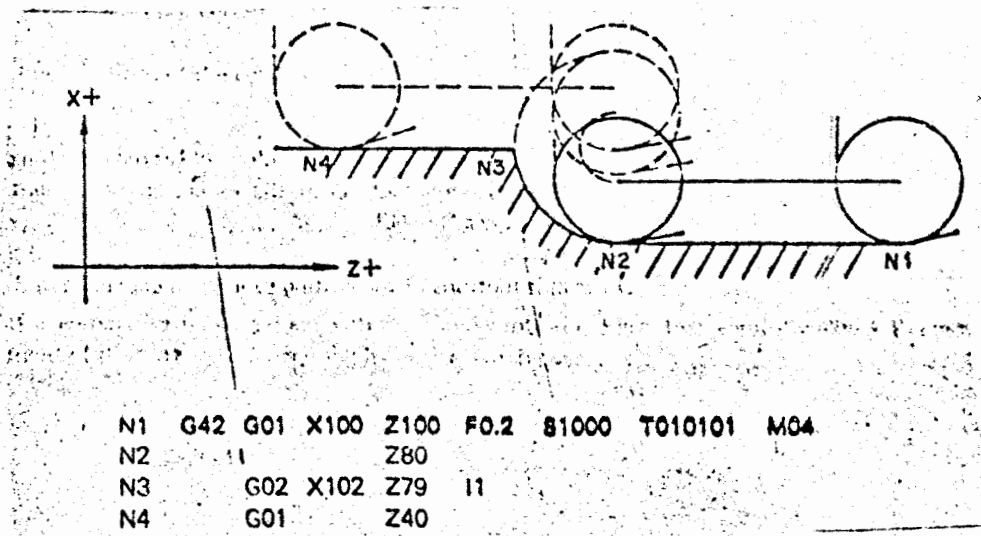


Рис. I63

В соответствии с командами в кадре №3 позиционирование режущего инструмента соответствует точке, в которой окружность вершины резца контактирует с продолжением дуги №2 - №3 и с продолжением прямой линии №3 - №4. Однако такую точку получить невозможно, так что система управления выполняет команды, содержащиеся в кадре №2, после чего выдается тревожное сообщение, а станок останавливается.

Внимание

При обработке внутренней стороны дуги радиус дуги должен быть равен или больше удвоенного радиуса закругления вершины резца.

$$R \geq 2 \times RN$$

где R - радиус дуги

RN - радиус окружности при вершине резца (радиус закругления вершины резца)

г) Дуга в трех квадрантах

Позиционирование с помощью команд, содержащихся в кадре №3, осуществляется до точки, в которой радиус закругления вершины резца касается продолжения прямой линии №1 - №2 и продолжение дуги №2 - №3 (см. рис. I64).

Разъяснения, относящиеся к другим перемещениям режущего инструмента вдоль координатных осей, приведены в п.3.2.2.

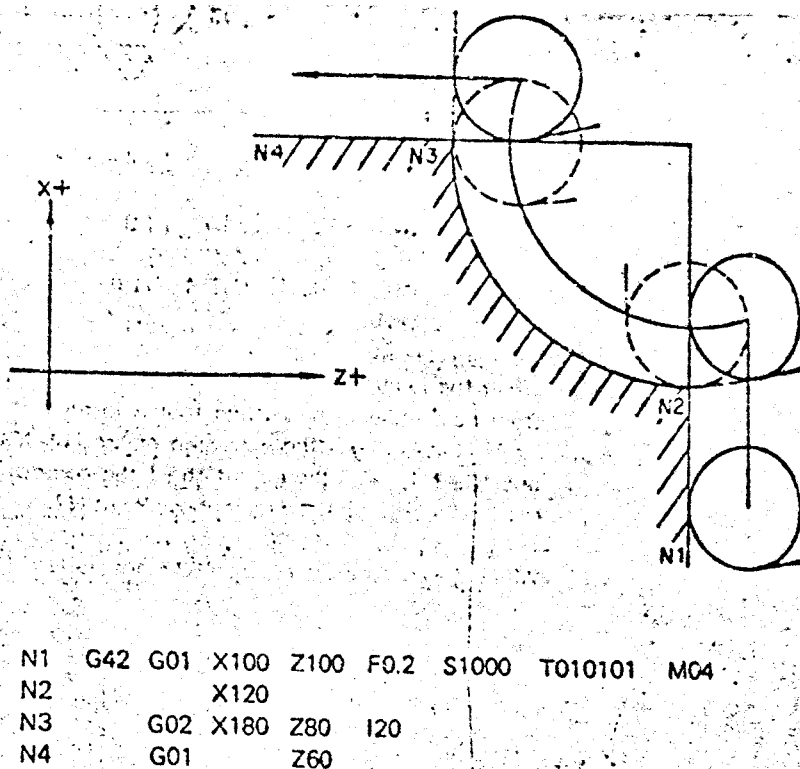


Рис. I64

15.2.3.3. Резание с переходом от одной дуги к следующей.

Программирование резания при переходе от одной дуги к следующей осуществляется по аналогии с переходом от прямой линии к дуге.

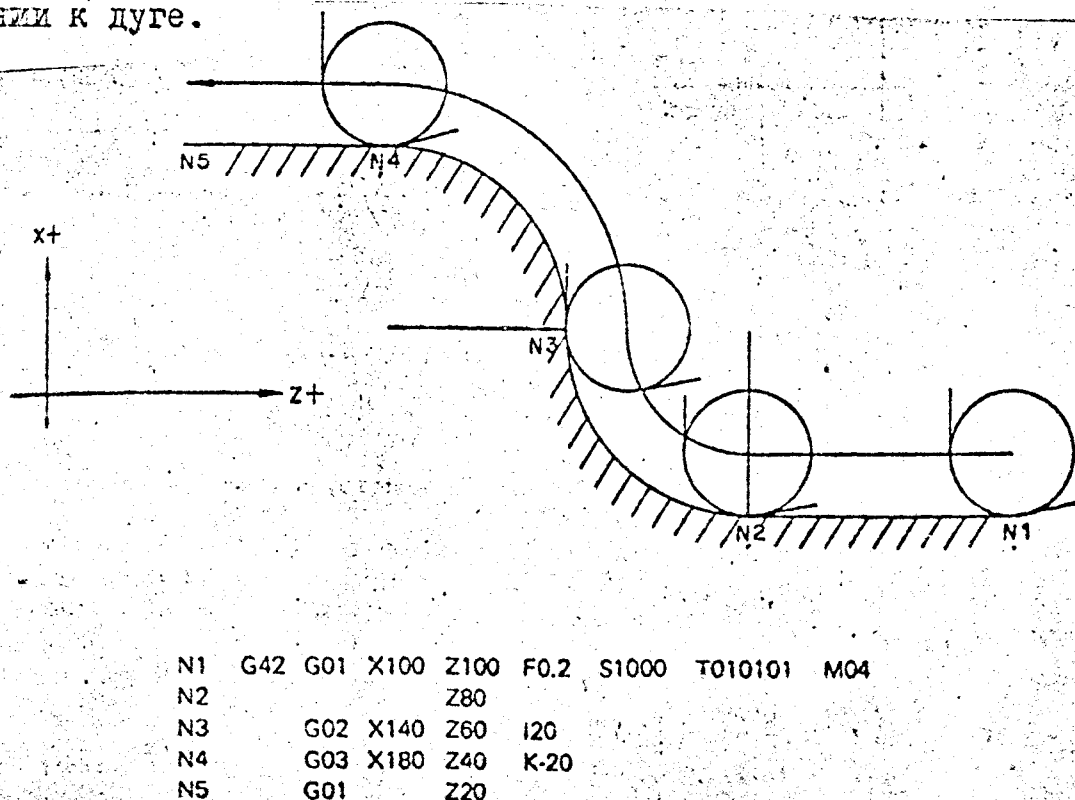


Рис. I65

Построение траектории режущего инструмента осуществляется таким образом, чтобы окружность вершины режущего инструмента контактировала бы с каждой дугой или ее продолжением.

Если траектория режущего инструмента имеет разрывы в процессе расчета траекторий вследствие ошибки, то в системе управления этот расчет повторяется.

Если после этого в траектории режущего инструмента сохраняются разрывы, то станок останавливается, а на экране дисплея появляется тревожное сообщение.

Разъяснение прочих перемещений режущего инструмента приводится в п. 3.2.

15.2.3.4. Переход от G41 к G42 и от G42 к G41

Для переключения режима компенсации радиуса закругления резца с G41 к G42 или с G42 на G41 рекомендуется отменить режим компенсации, который введен кодом G40.

Если переключение производится при действующем режиме компенсации, нужно тщательно проверить движение режущего инструмента, связанное с таким переключением.

I) Переключение во время резания с переходом от прямой линии к прямой линии (см. рис. I66).

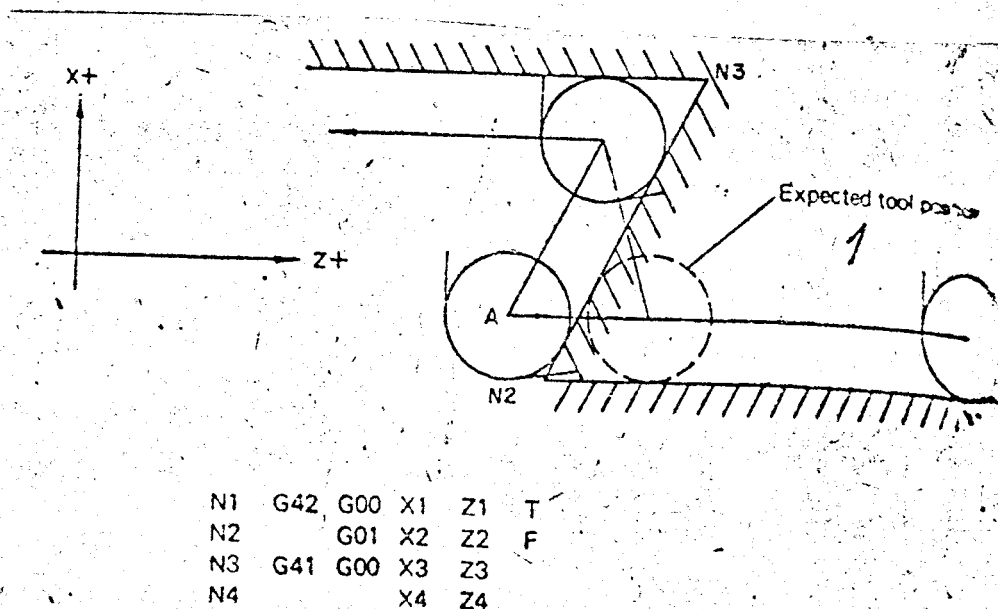


Рис. I66:

I - ожидаемое положение режущего инструмента

Ниже рассматривается перемещение режущего инструмента ,

создаваемое приведенной выше программой.

Кадры с командами №1 и №2 определяются кодом G42, а кадры с командами №3 и №4 определяются кодом G41. Для позиционирования режущего инструмента в точке №2 окружность вершины резца должна контактировать с правой стороной прямой линии №1 - №2, так как в кадре №2 установлен режим G42. В соответствии с кадром №3 центр окружности при вершине резца располагается на левой стороне прямой линии №2 - №3, так как в кадре №3 действует код G41. В результате режущий инструмент позиционируется в точке А в соответствии с приведенными выше указаниями.

Примечание. Позиционирование в соответствии с кадром №2 осуществляется на левой стороне прямой линии № 2 - №3.

2) Переключение во время резания с переходом от прямой линии к дуге.

Движение режущего инструмента в этом варианте можно объяснить довольно легко. Сохраняются те же принципы, что и в п.3.4.]

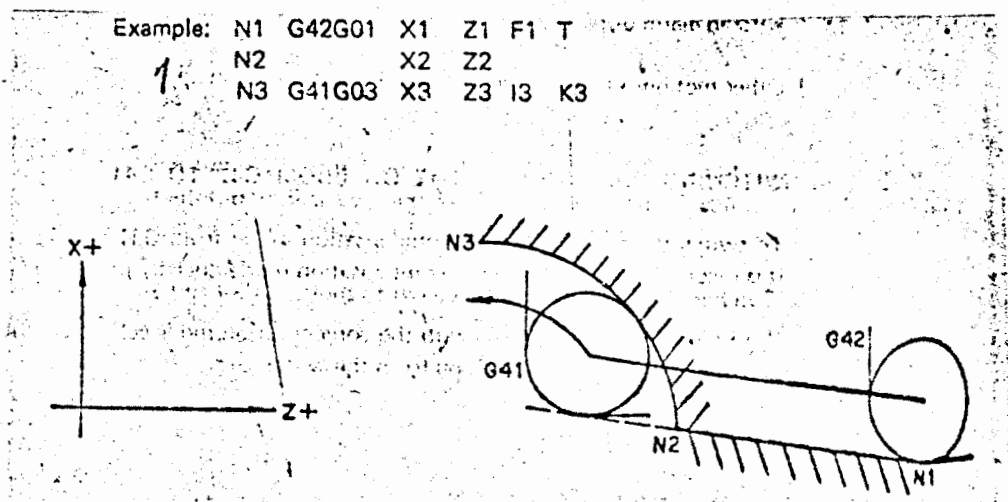


Рис. I67:

I - пример.

3) Переключение во время резания с переходом от дуги к прямой линии (см. рис. I68).

Это перемещение режущего инструмента разъяснить довольно легко. Используются те же принципы, что и в п. 3.4.1.

Example: N1 G42G01 X1 Z1 F1 T
1 N2 G03 X2 Z2 I2 K2
N3 G41G01 X3 Z3

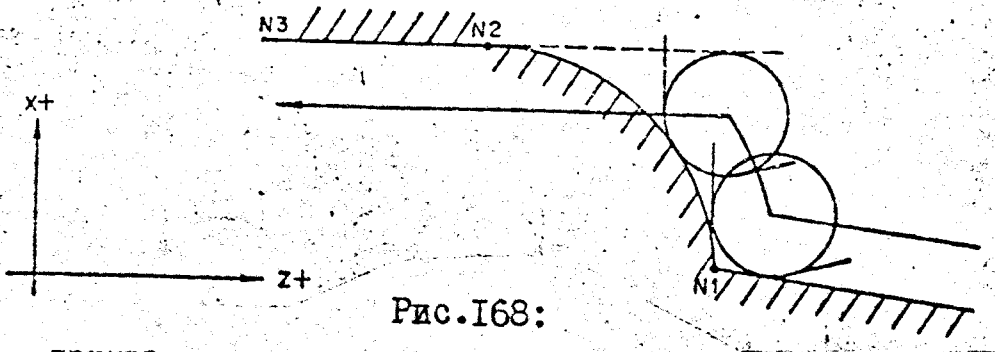


Рис. I68:

I - пример

4) Переключение во время резания с переходом от одной дуги к следующей (см. рис. I69).

Перемещение режущего инструмента определяется теми же принципами, что и в п. 3.4.1.

Example: N1 G42G01 X1 Z1 F1 T
1 N2 G02 X2 Z2 I2 K2
N3 G41G03 X3 Z3 I3 K3

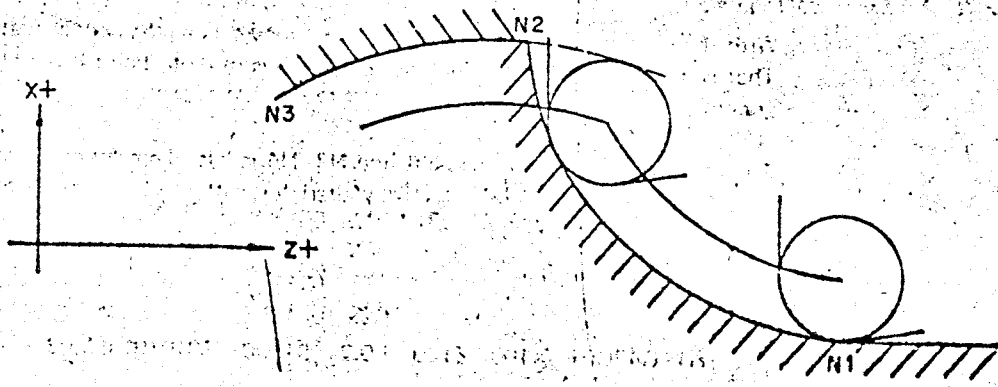


Рис. I69:

I - пример

15.2.4. Характеристики при отключении режима компенсации радиуса закругления вершины резца

15.2.4.1. Код G41 в сочетании с командами на перемещение вдоль координатных осей X или

Для отмены режима компенсации радиуса закругления вершины резца вводится код G40. Очень важно проверить перемещение режущего инструмента при отмене режима компенсации с целью исключения неожиданных затруднений.

В режиме компенсации радиуса закругления вершины резца формирование траектории режущего инструмента производится таким образом, что окружность при вершине резца всегда контактирует с программируемым контуром. Одновременно контролируется положение вдоль оси так, чтобы контрольная точка вершины резца следовала бы вдоль запрограммированного контура, когда введен режим компенсации радиуса закругления вершины резца. Таким образом часто возникает подрез или неполная обработка в случае ввода или отмены режима компенсации радиуса закругления вершины резца.

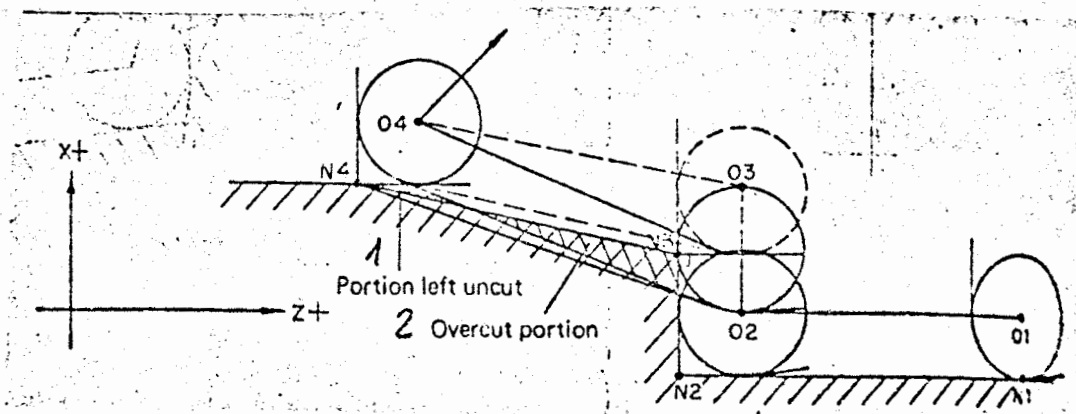


Рис. 170:

1 - оставшийся необработанный участок, 2 - участок подреза

При обработке контура, состоящего из прямоугольных участков, программирование осуществляется для варианта, когда не включена функция компенсации радиуса закругления вершины резца (см. приведенную ниже программу).

```
N1 G01 X100 Z100 F0.2 S1000 T010101 M03
N2          Z80
N3          X120
N4          X130 Z20
N5 G00 X300 Z300
```

При указанных командах режущий инструмент перемещается вдоль траектории, обозначенной пунктирными линиями. Позиционирование для программирования в точке №3 осуществляется для точки №3 и для точки №4 в случае запрограммированной точки №4.

Остается неподрезанный участок, параллельный прямой линии №3 - №4. Таким образом, функция компенсации радиуса закругления вершины реза эффективно используется для точной обработки подобного контура (см. приведенные ниже программы).

Программа 1

```
N1 G42G01 X100 Z100 F0.2 S1000 T010101 M03
N2          Z80
N3          X120
N4 G40     X130 Z20
N5 G00     X300 Z300
```

Траектория режущего инструмента, определяемая указанной программой, соответствует изображению сплошными линиями.

Позиционирование в запрограммированной точке №3 осуществляется до точки, в которой окружность вершины режущего инструмента контактирует с точкой №3, а позиционирование в запрограммированной точке №4 выполняется в соответствии с точкой 04. Аналогичная точка будет достигнута в соответствии с программой, в которую не включена функция компенсации радиуса закругления режущего инструмента.

Таким образом неподрезанный участок остается вблизи точки №4, а в зоне вблизи точки №3 имеется участок с избыточным подрезанием.

Программа 2

```
N1 G42G01 X100 Z100 F0.2 S1000 T010101 M03
N2          Z80
N3          X120
N4          X130 Z20
N5 G40G00 X300 Z300
```

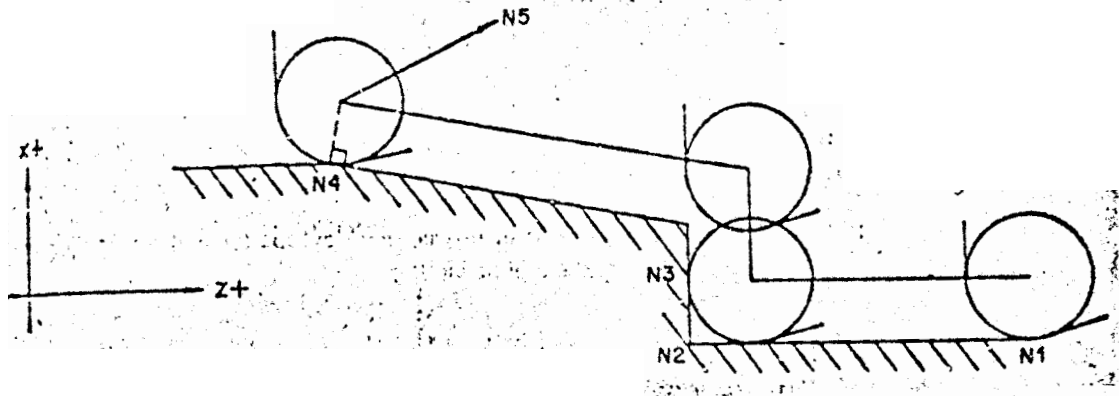


Рис. 171

Для обработки точного контура вплоть до точки №4 в кадре №5 используется код G40 для отмены режима компенсации радиуса закругления вершины резца.

С помощью этой программы получается ожидаемый контур, однако окружность при вершине резца выходит за программируемую точку №4 вдоль оси Z так как окружность вершины резца контактирует с точкой №4. Затруднения отсутствуют, если при этом не вводится избыточное перемещение или не наблюдается подрезание.

Для исключения подобного подрезания вдоль оси Z использовать программу 3.

Программа 3

```
N1 G42G01 X100 Z100 F0.2 S1000 T010101 M03
N2 Z60
N3 X120
N4 X130 Z20
N41 X134
N5 G40G00 X300 Z300
```

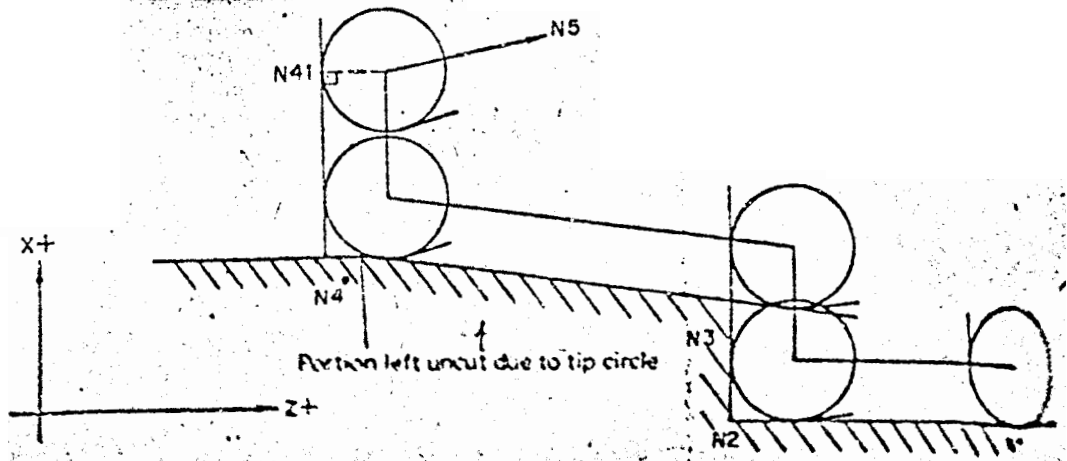


Рис. 172.

I - необработанный участок в связи с окружностью при вершине резца

В соответствии с этой программой в кадры вводится команда на отвод режущего инструмента вдоль оси X. Кроме того, в кадре №5 отменяется режим компенсации радиуса закругления вершины резца.

Эта программа обеспечивает необходимую обработку, хотя и остается небольшой необработанный участок вблизи запрограммированной точки M в связи с окружностью при вершине резца.

Программа 4

```
N1 G42G01 X100 Z100 F0.2 S1000 T010101 M03
N2 Z60
N3 X120
N41 X130.5 Z18
N42 X134
N5 G40G00 X300 Z300
```

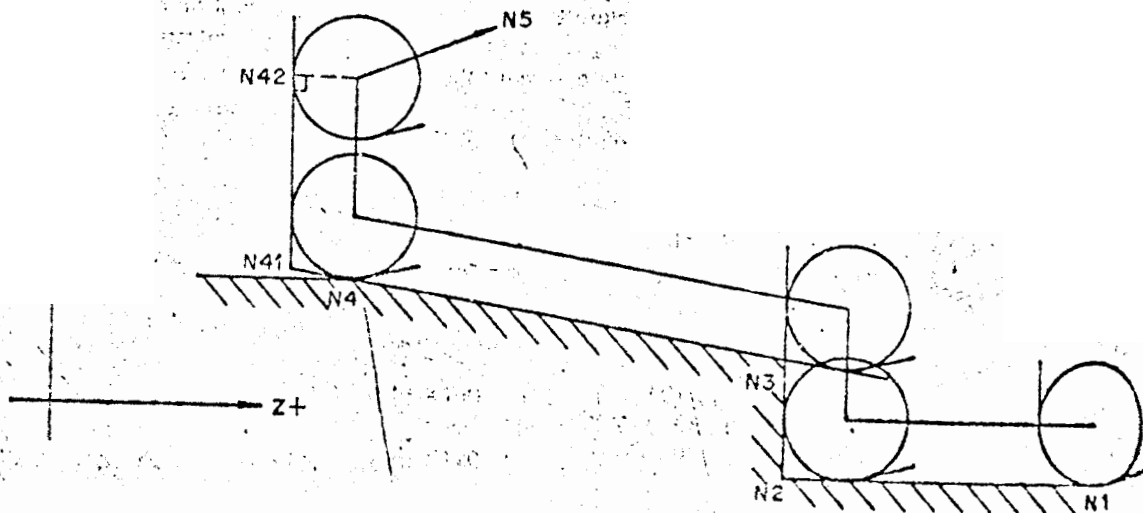


Рис. I73

В соответствии с этой программой прямая линия M3 - M4 продолжается до точки M4I, а режущий инструмент отводится в направлении +X в кадре №42. В кадре №5 производится отмена режима компенсации радиуса закругления вершины резца.

Эта программа обеспечивает обработку точного контура вплоть до точки M4, так что необработанный участок не остается.

15.2.4.2. Команды I и K в соответствии с кодом G40

В блоке с кодом G40 вводимые слова I и K обозначают воображаемую точку для компенсации радиуса закругления вершины резца вместе с словами X и Z, которые обозначают точку, в которой отменяется компенсация радиуса закругления вершины резца. При этом отменяются ненужные перемещения вдоль осей, которые требуются для обычной программы.

```

N1 G42G01 X100 Z100 F0.2 S1000 T010101 M03
N2           Z60
N3           X120
N4           X130 Z20
N5 G00G00 X300 Z300 I10 K0
    
```

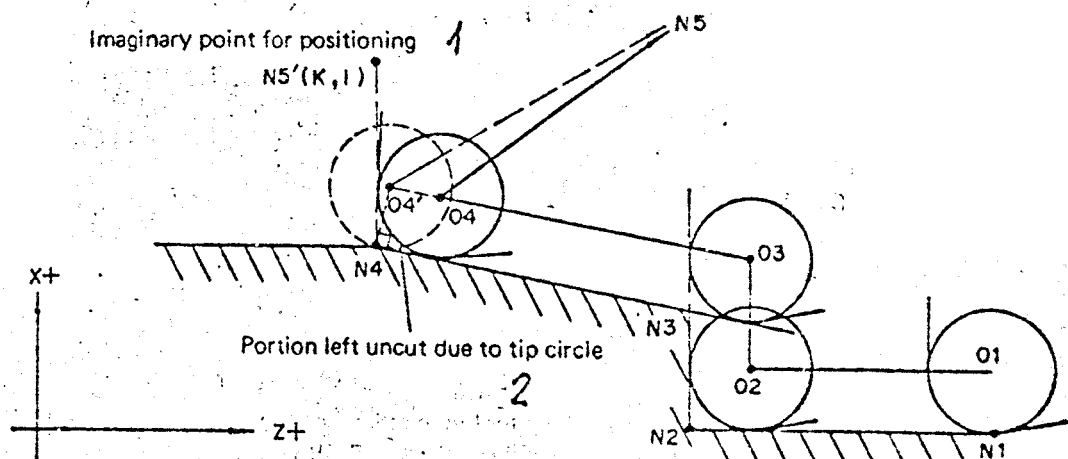


Рис. 174:

1 – воображаемая точка для программирования, 2 – неподрезанный участок, связанный с радиусом закругления вершины резца

Если в кадре №5 с кодом G40 отсутствуют слова I и K, установить режущий инструмент с помощью команд, содержащихся в кадре №4, так чтобы окружность при вершине резца контактировала с запрограммированной точкой №4, а затем перемещалась бы вдоль траектории, обозначенной пунктирными линиями вплоть до точки №5.

При введении слов I и K в кадр №5 обеспечивается позиционирование режущего инструмента в точке, в которой окружность вершины резца контактирует с прямой линией №3 - №4 и воображаемой прямой линией №4 - №5' при осуществлении команд в кадре №4.

Выполнение команд в кадре №5 обеспечивает подведение режущего инструмента к запрограммированной точке №5, в которой отменена функция компенсации радиуса закругления при вершине резца.

Примечание. Ввод команд I и K осуществляется ступенчато. В этом случае размеры определяются с учетом точки №4.

Если указывается лишь одно из слов I или K, система управления устанавливает значение соответствующего слова, равное 0. При этом КО в указанной выше программе исключается.

15.2.4.3. Независимый код G40

Если код G40 программируется без других команд в одном кадре, то позиционирование производится в точке, в которой окружность при вершине резца контактирует с точкой, указанной в предыдущем кадре с командой G40, но без слов X и Z для включения перемещения вдоль оси.

```
N1 G42G01 X100 Z100 F0.2 S1000 T010101 M03  
N2           Z60  
N3           X120  
N4           X130 Z20  
N5 G40  
N6           G00 X300 Z300
```

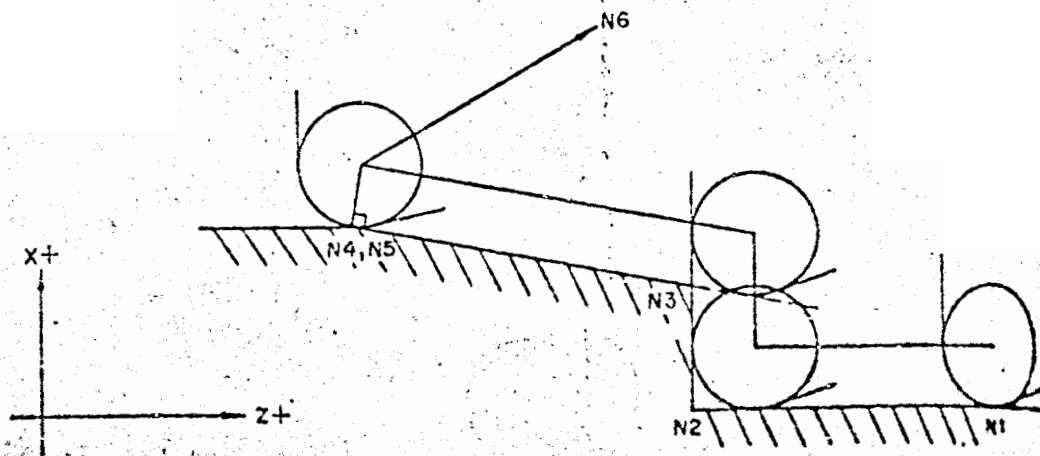


Рис. 175

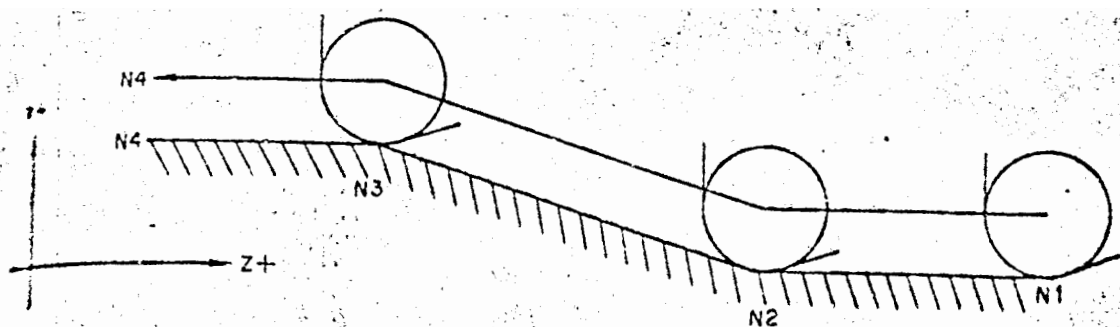
Внимание. При отмене режима компенсации радиуса закругления вершины резца (G40) режим должен определяться G00 или G01. В противном случае выдается тревожное сообщение.

15.2.5. Отвод резца для изменения режима S или M во время резания

Функция компенсации радиуса закругления вершины резца предназначена для автоматической компенсации радиуса закругления в программе непрерывного резания. При использовании размеров заготовки в принятой программе компенсация вводится автоматически для чистовой обработки в соответствии с запрограммированными размерами. Однако подобная функция требует тщательного программирования для прекращения непрерывного резания и изменения команд S или M.

В данном разделе рассматриваются некоторые примеры программирования, когда программист встречается с неожиданными результатами при отводе режущего инструмента во время резания по непрерывной траектории.

Ниже рассматривается исходный контур и соответствующая ему программа (рис.176).



N1	G42G01	X100	Z100	F0.2	S1500	T010101	M03
N2			Z80				
N3		X120	Z40		S1000		
N4			Z20				

Рис.176

Исходный контур состоит из прямой линии, наклонного участка, еще одной прямой линии и не содержит более сложных участков

На основании этой программы составляется программа отвода режущего инструмента в точке №3 в направлении +X с целью изменения скорости шпинделя (см. приведенные ниже программы).

15.2.5.1. Программа I

```
N1 G42G01 X100 Z100 F0.2 S1500 T010101 M03
N2          Z80
N3          X120
N31 G00 X124
N32 G01 X120          S100
N4          Z20
```

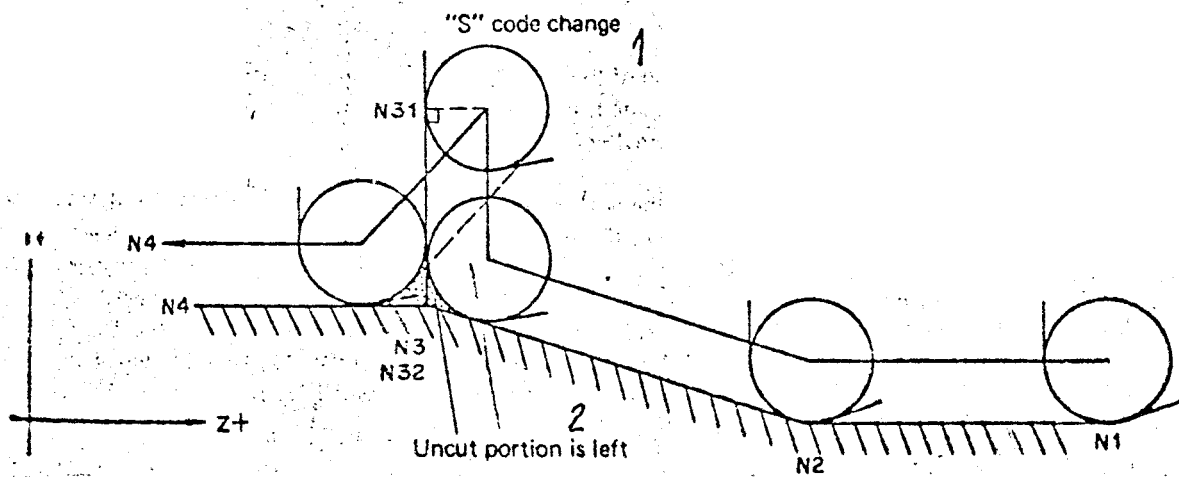


Рис.177:

1 - изменение кода, 2 - оставшийся неподрезанный участок

Отвод режущего инструмента производится в точке №3 в направлении +X, затем меняют скорость шпинделя и включают непрерывное резание.

При этой программе режущий инструмент устанавливается в точке, в которой окружность при вершине резца контактирует с точкой №31 при осуществлении команд в кадре №31 для трех программируемых точек №3, №31 и №32, которые лежат на одной прямой линии. Перемещение от точки №1 к точке №31 на стороне позиционирования по отношению к соответствующей линии осуществляется вправо. И наоборот, команды в блоке обеспечивают позиционирование режущего инструмента в точке, в которой окружность вершины резца контактирует с прямой линией №31 - №32 и №3 - №4

на правой стороне в направлении движения резца. Благодаря этому режущий инструмент перемещается не только в направлении оси X, но и в направлении оси Z, хотя в кадре №32 содержится только слово X.

При таком движении режущего инструмента остается неподрезанный участок, показанный выше.

15.2.5.2. Программа 2

```
N1 G42G01 X100 Z100 F0.2 S1500 T010101 M03
N2           Z80
N3           X120 Z40
N31 G00 X124
N32           X120 Z42 S1000
N4 G01           Z20
```

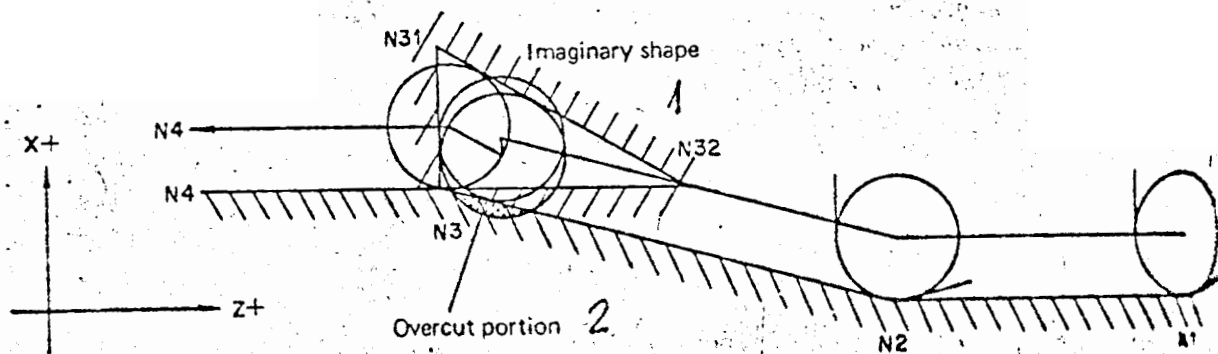


Рис.178:

1 - воображаемый контур, 2 - подрезанный участок

В соответствии с этой программой предусматривается исключение неподрезанного участка, полученного для программы 1. Несмотря на исключение неподрезанного участка, отмечается затруднение, связанное с избыточным подрезанием, которое рассматривается ниже.

После того, как система управления направляет режущий инструмент из точки №2 в точку №3, осуществляется считывание данных позиционирования для точки №31, а также для точки №3. При этом окружность при вершине режущего инструмента, позицио-

нируется с учетом двух прямых линий №2 - №3 и №3 - №31. После этого позиционирование выполняется в точке, в которой окружность при вершине резца контактирует с двумя прямыми линиями №3 - №31 и №31 - №32, когда позиционирование производится в соответствии с командами, содержащимися в кадре №31. При этом режущий инструмент перемещается в направлении -X, хотя команды в этом кадре определяют перемещение резца в направлении +X. Это можно ожидать в результате позиционирования в кадре №3, когда окружность вершины резца находится на стороне линии №31 - №32.

Аналогичное позиционирование режущего инструмента в кадре №32 осуществляется для точки, в которой окружность вершины резца контактирует с двумя прямыми линиями №31 - №32 и №32 - №4. При этом режущий инструмент перемещается в направлении, противоположном запрограммированному.

15.2.5.3. Программа 3

```

N1 G42G01 X100 Z100 F0.2 S1500 T010101 M03
N2           Z80
N3           X120 Z40
N31 G00 X126
N32 X120 Z43 S1000
N4 G01 Z20
    
```

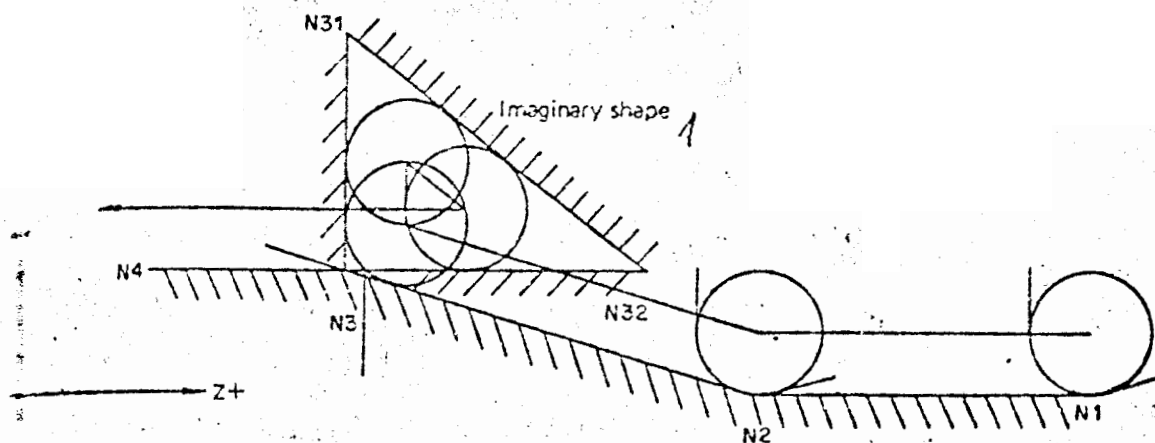


Рис. 179:

I - воображаемый контур

При этой программе режущий инструмент перемещается по контуру по аналогии с программой 2, хотя числовые значения

изменены для исключения подрезания.

Указанная программа позволяет получить почти ожидаемый чистовой контур, однако сохраняются следующие затруднения:

- возникает подрезание в зависимости от размера окружности при вершине резца.

- нельзя легко определить длину стороны №31 - №32.

Указанные проблемы решаются с помощью замкнутой траектории резца вдоль квадрата в соответствии с приведенными ниже разъяснениями.

15.2.5.4. Программа 4

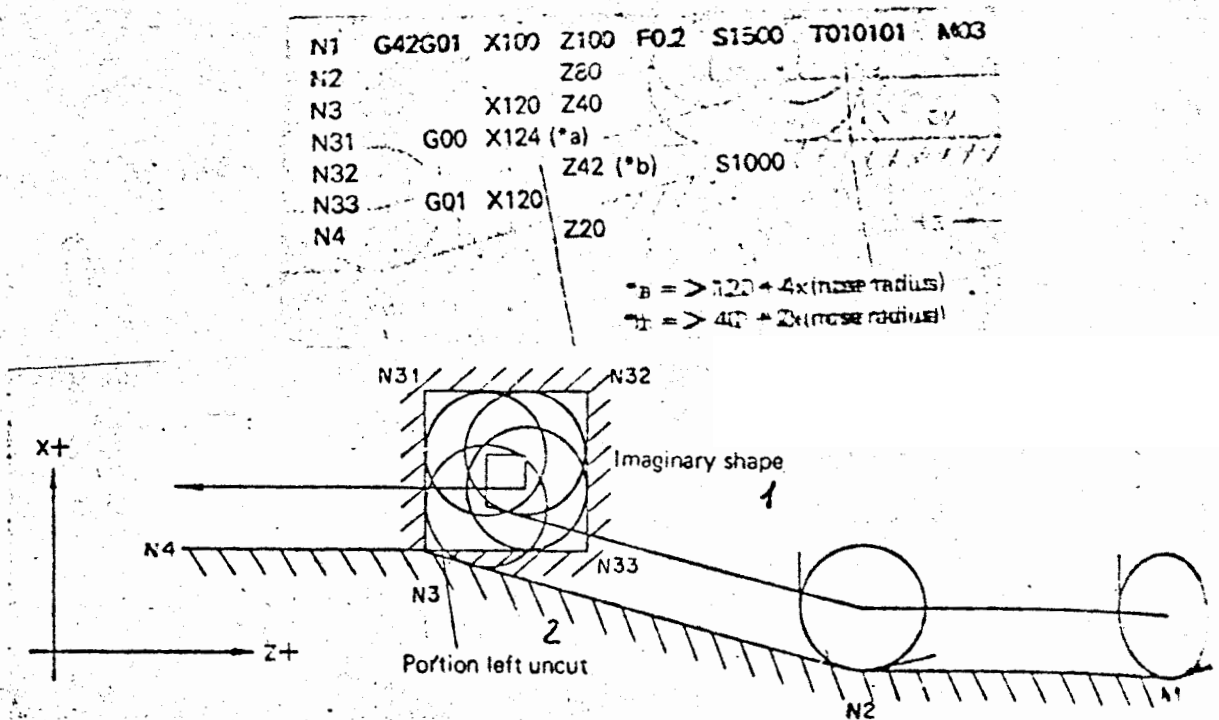


Рис.180:

1 - воображаемый контур, 2 - неподрезанный участок

При замкнутой траектории окружность при вершине резца перемещается вдоль запрограммированного прямоугольника №3 - №31 - №32 - №33. В связи с этим подобную характеристику для оси можно ожидать только, когда соответствующие стороны более, чем в

два раза превышает радиус закругления вершины резца (в 4 раза по оси X).

При этой программе остается неподрезанный участок, причем дальнейшее усовершенствование возможно при использовании программы 5.

15.2.5.5. Программа 5

```
N1 G42G01 X100 Z100 F0.2 S1500 T010101 M03
N2 Z80
N3 X120.5 Z39
N31 G00 X124 (*a)
N32 Z42 (*b) S1000
N33 G01 X120
N4 Z20
```

*a = > 120.5 + 4x(nose radius) 1
*b = > 39 + 2x(nose radius) 1

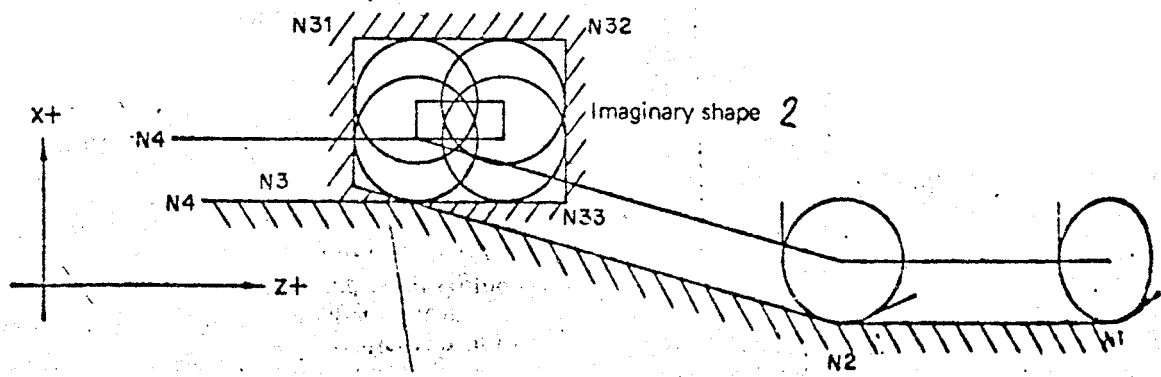


Рис. I8I:

I - радиус закругления вершины, 2 - воображаемый контур

Сдвиг точки N3 происходит в направлении Z на величину радиуса закругления вершины резца для того, чтобы исключить неподрезанный участок, получающийся в соответствии с программой 4.

Программы I-5 позволяют использовать некоторые приемы для осуществления намеченной программы обработки.

Восбращаемый контур замкнутой траектории режущего инструмента предусматривает выбор треугольника или многоугольника, но не треугольника. При использовании треугольников происходят неожиданные перемещения режущих инструментов.

15.2.6. Меры предосторожности

1) Если значение по оси X или Z превышает соответствующее предельное значение, определяемое программой, то выдается тревожное сообщение о нарушении предела.

2) В режиме компенсации радиуса закругления режущего инструмента невозможно запрограммировать команды, которые не создают перемещение вдоль оси, несмотря на наличие размерных слов (например, нулевой сдвиг, определяемый кодом C, или постоянный цикл нарезания резьбы).

3) Для включения режима компенсации радиуса закругления при вершине резца в режиме работы с автоматическим программированием IAP нужно ввести коды G41 или G42 в кадр, который предшествует кадру с кодами G81 или G82 с запрограммированными размерами обработки в режиме IAP. При этом режиме функция компенсации радиуса закругления вершины резца вводится в циклы как черновой, так и чистовой обработки.

Обязательно ввести код G40, который отменяет режим компенсации радиуса закругления вершины резца до указания окончания кода G80 обозначения контура при автоматическом программировании IAP.

4) В режиме компенсации радиуса закругления вершины резца не следует выдавать в последовательных командах одну и ту же точку. Возможна выдача одного кадра, в котором отсутствуют команды на перемещение вдоль оси, так как система управления рассчитана на введение подобного кадра.

5) На начальном этапе режима компенсации радиуса закругления вершины резца система управления начинает ввод команд после считывания в последовательных двух кадрах. Таким образом, при нажатии кнопки "включение цикла" в режиме ручного ввода данных после ввода команд одного кадра включение работы станка невозможно.

6) Возможны команды с приращением в режиме компенсации радиуса закругления вершины резца.

Раздел 16. Задача №1 для пользователя

16.1. Описание функции задачи для пользователя

Фирма "Окума" после долголетней работы в области обработки данных имеет возможность оказать пользователю эффективную помощь при переработке огромного объема данных с помощью системы ОУР ОУР 5000L. Для этого создана функция задачи пользователя.

Наибольшее преимущество функции задачи пользователя заключается в возможности изменения различных рабочих функций. Кроме того, использование управляющих сообщений обеспечивает гибкость функции задачи пользователя.

Предусмотрены два варианта функции задачи пользователя:

- 1) Задача №1 для пользователя (стандартная)
- 2) Задача №2 для пользователя (дополнительная)

В данной инструкции подробно рассматривается задача №1 для пользователя, которая предусматривает выполнение арифметических действий, использование переменных параметров и сообщений. Задача №1 для пользователя предусмотрена главным образом для выполнения непрерывной автоматической обработки с управлением периферийным оборудованием, например, питателями прутковых заготовок, загрузчиками или роботами. Кроме того, предусмотрены функции управления запланированными операциями на станках, что позволяет производить обработку различных заготовок в соответствии с запрограммированным графиком.

Задача №2 для пользователя представляет адекватную функцию задачи с арифметическими действиями, переменными параметрами и сообщениями. В задаче №2 для пользователя предусмотрены все функции, характерные для задачи №1 для пользователя.

16.2. Основные функции для задачи пользователя

Ниже перечислены три основные функции задачи для пользователя:

- 1) Функции сообщений для управления

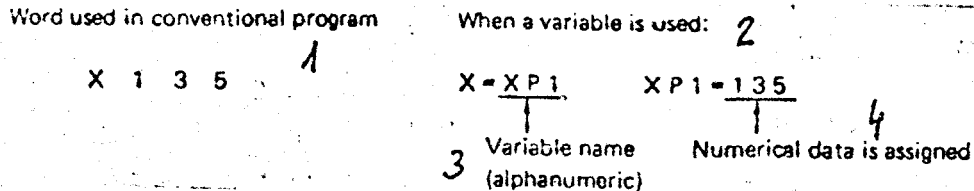
Возможно использование сообщений для управления исполнением запрограммированных последовательностей, например "IF(если) GOTO (переход).

- 2) Переменные параметры

При обычном программировании цифровые данные обычно указываются после адресующих знаков, например X, Y и Z. Вместо цифровых данных с знаками адресования могут быть связаны пере-

менные параметры, выраженные в алфавитно-цифровой форме. При этом фактические цифровые данные связаны с переменными параметрами в соответствующих программах. Эта особенность обеспечивает гибкость и универсальность программы.

Пример:

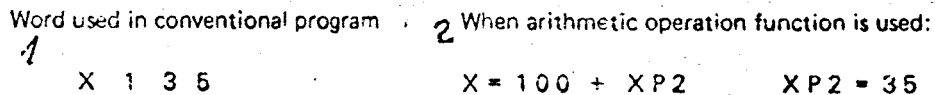


1 - слово, используемое в обычной программе, 2 - использование переменного параметра, 3 - переменное наименование (алфавитно-числовое), 4 - назначение числовых данных

3) Функция арифметических действий

Эта функция обеспечивает прямое программирование арифметических выражений с помощью данных слова.

Пример:



1 - слово, используемое при обычной программе, 2 - использование функций арифметических действий

Ниже приводится подробное описание основных функций задачи пользователя.

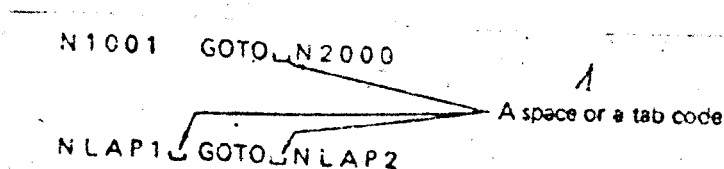
16.2.1. Сообщение для управления

Задача Ю1 для пользователя предусматривает использование двух следующих сообщений для управления: IF (если), GOTO (переход).

Программирование этих сообщений для управления осуществляется либо в начале кадра, или же сразу после программируемой последовательности в начале кадра. Обязательно предусмотреть промежуток или разделительный код после номера последовательности или для разделения использовать сообщение для управления. В противном случае выдается тревожное сообщение.

Если сообщение IF не требуется для создания промежутка, то после этого сообщения указывается знак. - ["

Пример



I - промежуток или код пропуска

После элемента, содержащего более, чем два знака адресования (от A до Z), например, после номера последовательности и кода управления, всегда должен быть указан промежуток или код пропуска.

Примечание I. Наименование последовательности

Наименование последовательности представляет собой код для идентификации соответствующих кадров в программе, причем он состоит из четырех алфавитно-цифровых знаков после знака адресования.

Используются два варианта наименования последовательности.

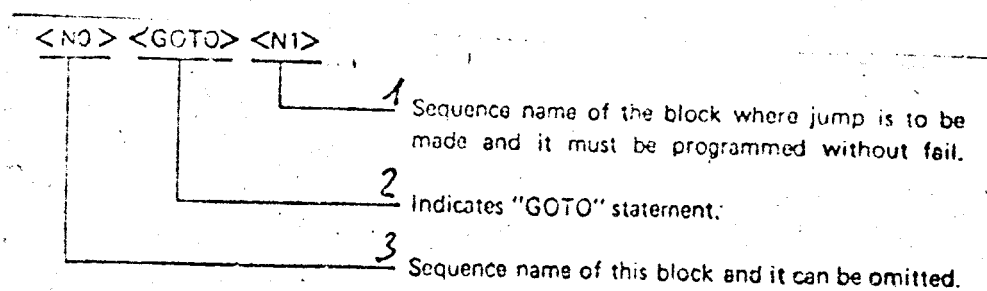
<N> <четыре алфавитно-цифровых знака>

<N><буква> <три алфавитно-цифровых знака>

Рассматриваемые в данной инструкции наименования последовательности относятся к обоим типам наименований.

16.2.1.1. Сообщение GOTO, переход без дополнительных условий

I) Формат программы



1 - наименование последовательности в кадре, в котором предусмотрен переход с обязательным программированием, 2 - индикация сообщения GOTO, 3 - наименование последовательности в кадре и возможность исключения

Примечание. Наименование последовательности <N1> для перехода должно соответствовать программе, содержащей подобное сообщение для управления.

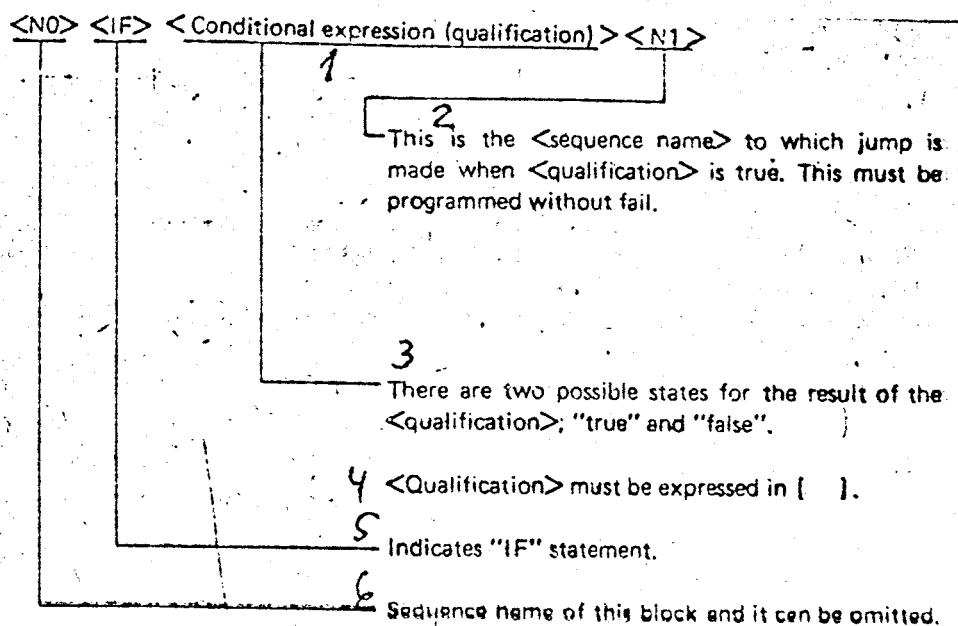
2) Функция

При выдаче сообщения для управления обязательно происходит переход к запрограммированному кадру NI.

Примечание. На станках с двумя револьверными головками невозможен переход от программы для револьверной головки А к программе для револьверной головки В. Кроме того, невозможен переход последовательности от программы для револьверной головки В к программе для револьверной головки А.

16.2.1.2. Сообщение IF, обусловленный переход

1) Формат программы



1 - обусловленное выражение (квалификация), 2 - это соответствует наименованию последовательности, в которой переход осуществляется при условии правильной квалификации. Обязательно программировать этот параметр, 3 - существуют два варианта состояния в результате правильной или ошибочной квалификации, 4 - квалификацию нельзя указывать в квадратных скобках, 5 - индикация сообщения IF, 6 - наименование последовательности в указанном кадре, но с возможностью исключения

2) Функция

а) При правильной квалификации выполнение последовательности перебрасывается в точку NI.

б) При ошибочной квалификации выполняется приведенная ниже последовательность.

Пример

N1000 IF (V1 EO 10) N2000
Means "equal (=)".

I - соответствует равенству

Производится переход к №2000, когда переменный параметр U1 равняется 10. При этом выполняется следующий кадр

16.2.2. Общий переменный параметр

В задаче №1 для пользователя предусмотрены два типа переменных параметров:

- Общий переменный параметр
- Местный переменный параметр

Использование и характеристики этих двух типов переменных параметров различаются.

16.2.2.1. Общий переменный параметр

Термин "Общий" в отношении общих переменных параметров можно рассматривать как общий для главной программы и подпрограммы. Если один и тот же переменный параметр используется в двух или нескольких программах, то переменное число, применяемое в этих программах, должно быть одинаковым. Таким образом, общий переменный параметр, полученный в результате расчета для одной программы, может применяться и для других программ

1) Формат программы

<V> <одна или две цифры> = числовые данные или выражения

Обозначение общего переменного параметра производится с помощью двух знаков с последующей буквой V (например общий переменный параметр U1 - U32).

В программе общий переменный параметр используется в следующем виде:

N101 V5 = 10 или N101 V5 = V5 + 1

Примечание 1. Общие переменные параметры используются для общих функций, относящихся к револьверной головке А и револьверной головке В.

Примечание 2. Общие переменные параметры действуют как в главной программе, так и в подпрограмме.

Примечание 3. На общие переменные параметры не влияет перестройка системы управления или отключение питания. Таким образом данные сохраняются до тех пор, пока не произведена перестройка или не загружена лента с управляющей программой.

Примечание 4. Настройка или изменение общих переменных параметров возможна при введении параметра помимо изменяемого в соответствии с программой. Более подробные сведения относящиеся к этому параметру приведены в п.4.3. инструкции по эксплуатации.

16.2.2.2. Местные переменные параметры

Как следует из термина "местный" местные переменные параметры предусматривают возможность их ввода пользователем с четким наименованием для осуществления необходимых отличий. Так, например, для револьверных головок А и В можно использовать до 127 местных переменных параметров.

1) Формат программы

< Letter > < Letter > < two alphanumerics > = Numerical data or expression
 ↑ 1 1 2 3

"0", "N" and "V" cannot be used. 4

Example 5 'DIA1' 'ITH5'

1 - буква, 2 - два алфавитно-цифровых знака, 3 - цифровые данные или выражения, 4 - невозможность использования, 5 - пример

Невозможно использовать те же наименования, которые присвоены промежуточным операторам, рассмотренным в п. 2.3.2. с расширением знака адресования * I.

I. Расширенные знаки адресования используются для автоматического программирования IAP, для обработки шаблонов, для постоянных циклов пользователя и других расширенных функций, которые не охватываются обычными знаками адресования от А до Z. Ниже перечислены расширенные знаки адресования

<AA>, <AB>, <DA>, <DB>, <FA>, <FB>, <IA>, <IB>.
<KA>, <KB>, <LA>, <LB>, <RA>, <RB>, <SA>, <SB>.
<TA>, <TB>, <UA>, <UB>, <WA>, <WB>, <XA>, <XB>.
<ZA>, <ZB>, <BC>,
.

2) Характеристики местных переменных параметров

а) Местные переменные параметры можно использовать независимо для револьверной головки А и револьверной головки В. Стирание осуществляется при перестройке системы управления.

б) Если в главную программу вводится новая группа местных переменных параметров, например данные, связанные с наименованием нового переменного параметра, в запоминающем устройстве производится регистрация наименования местного переменного параметра и соответствующих данных.

При использовании наименования местного переменного параметра без соответствующих данных включается тревожная сигнализация.

в) Если новые данные относятся к местному переменному параметру, для которого ранее зарегистрированы другие данные, то производится обновление старых данных.

1 Mainprogram
:
:
N0010 DIA1 = 160
:
N0049
N0050 DIA1 = 200

2 In N0010, numerical data "160" is assigned to local variable name "DIA1", which remains effective up to sequence N0049. In N0050, new numerical data "200" is assigned to the same local variable name "DIA1". This clears the old data "160" and it is substituted with the new data "200".

1 - главная программа, 2 - для N0010 введены цифровые данные 160, соответствующие наименованию DIA1 местного переменного параметра, которое сохраняет эффективность вплоть до последовательности N0049. Для N0050 вводится новое цифровое значение 200 при неизменном наименовании DIA1 местного переменного параметра. При этом старое значение 160 стирается и заменяется новым значением 200

г) Для револьверных головок А и В соответственно может использоваться до 127 переменных параметров.

16.2.3. Функция арифметических действий

Возможно выполнение арифметических действий с использованием переменных параметров. Программирование может осуществляться по аналогии с общими арифметическими выражениями.

< Знак адресования > < знак расширенного адреса > < переменный параметр > < выражение >

Правое выражение, предусматривающее арифметическое действие составлено из переменных параметров, сравнительных выражений и операторов.

Ниже рассматриваются некоторые варианты и их содержание

16.2.3.1. Арифметическое выражение

Таблица -22

Operator 1	Meaning 2	Example 3	4 Rule and Remarks
+	Positive sign 5	+ 1 2 3 4	
-	Negative sign 6	- 1 2 3 4	
+	Sum (addition) 7	X = 12.3 + V1	
-	Difference (subtraction) 8	X = 12.3 - V1	
*	Product (multiplication) 9	X = V10 * 10	
/	Quotient (division) 10	X = V11 / 10	

1 - оператор, 2 - значение, 3 - пример, 4 - правило и замечания, 5 - положительный знак, 6 - отрицательный знак, 7 - сумма (сложение), 8 - разность (вычитание), 9 - произведение (умножение), 10 - частное (деление)

16.2.3.2. Сравнение выражений

Таблица 23

Operator 1	Meaning 2	Example 3	Contents 4	5
LT	(Less Than, <) 6	IF [V1 LT 5] N100	Jump to N100 when V1 is less than 5. 12	13 Provide a space on either side of the operator
LE	(Less Than or Equal to, ≤) 7	IF [V1 LE 5] N100	Jump to N100 when V1 is less than or equal to 5. 13	
EQ	(Equal to, =) 8	IF [V1 EQ 5] N100	Jump to N100 when V1 is equal to 5. 14	
NE	(Not Equal to, ≠) 9	IF [V1 NE 5] N100	Jump to N100 when V1 is not equal to 5. 15	
GT	(Greater Than, >) 10	IF [V1 GT 5] N100	Jump to N100 when V1 is greater than 5. 16	
GE	(Greater than or Equal, ≥) 11	IF [V1 GE 5] N100	Jump to N100 when V1 is greater than or 17	

1 - оператор, 2 - значение, 3 - пример, 4 - содержание,

5 - правило, 6 - менее, чем, 7 - менее чем или равняется, 8 - равняется, 9 - не равняется, 10 - более чем, 11 - более чем или равняется, 12 - переход к указанному значению, если UI менее, чем 5, 13 - переход к указанному значению, если UI меньше или равняется 5, 14 - переход к указанному значению, если UI равняется 5, 15 - переход к указанному значению, если UI не равняется 5, 16 - переход к указанному значению, если UI превышает 5, 17 - переход к указанному значению, если UI больше или равняется 5, 18 - предусмотреть промежуток с каждой стороны оператора

16.3. Пример программы

Ниже рассматривается пример упрощенной программы для облегчения понимания содержания задачи пользователя МТ.

Вводится допущение о том, что токарный станок с ЧПУ оснащен устройством для загрузки прутков. В соответствии с запрограммированным графиком на этом станке обрабатываются заготовки трех типов.

1) Присвоить соответствующим заготовкам № файла и № программы.

Workpiece A	A. MIN, 0100
↑ Workpiece B	B. MIN, 0200
Workpiece C	C. MIN, 0300

I - заготовка

2) После определения наименования файла и наименования (номера) программы подготовить программу для обработки заготовки в соответствии с чертежом детали.

3) Перед тем, как приступить к фактической обработке на станке, нужно определить количество обрабатываемых заготовок и последовательность их обработки, т.е. график работы.

Количество заготовок А : 20 штук

Количество заготовок В : 15 штук

Количество заготовок С : 25 штук

Последовательность обработки А - В - С

4) Программист может приступить к составлению программы последовательности.

Определить наименование файла и программу последователь-

ности. Наименование файла определяется в виде НАГТ-1 Г в связи с тем, что предусматривается обработка деталей типа валов.

5) Подсчет готовых деталей производится с использованием общего переменного параметра.

Workpiece A V1
 Workpiece B V2
 Workpiece C V3

I - заготовка

6) Составить программу графика в соответствии с циклограммой на рис.182.

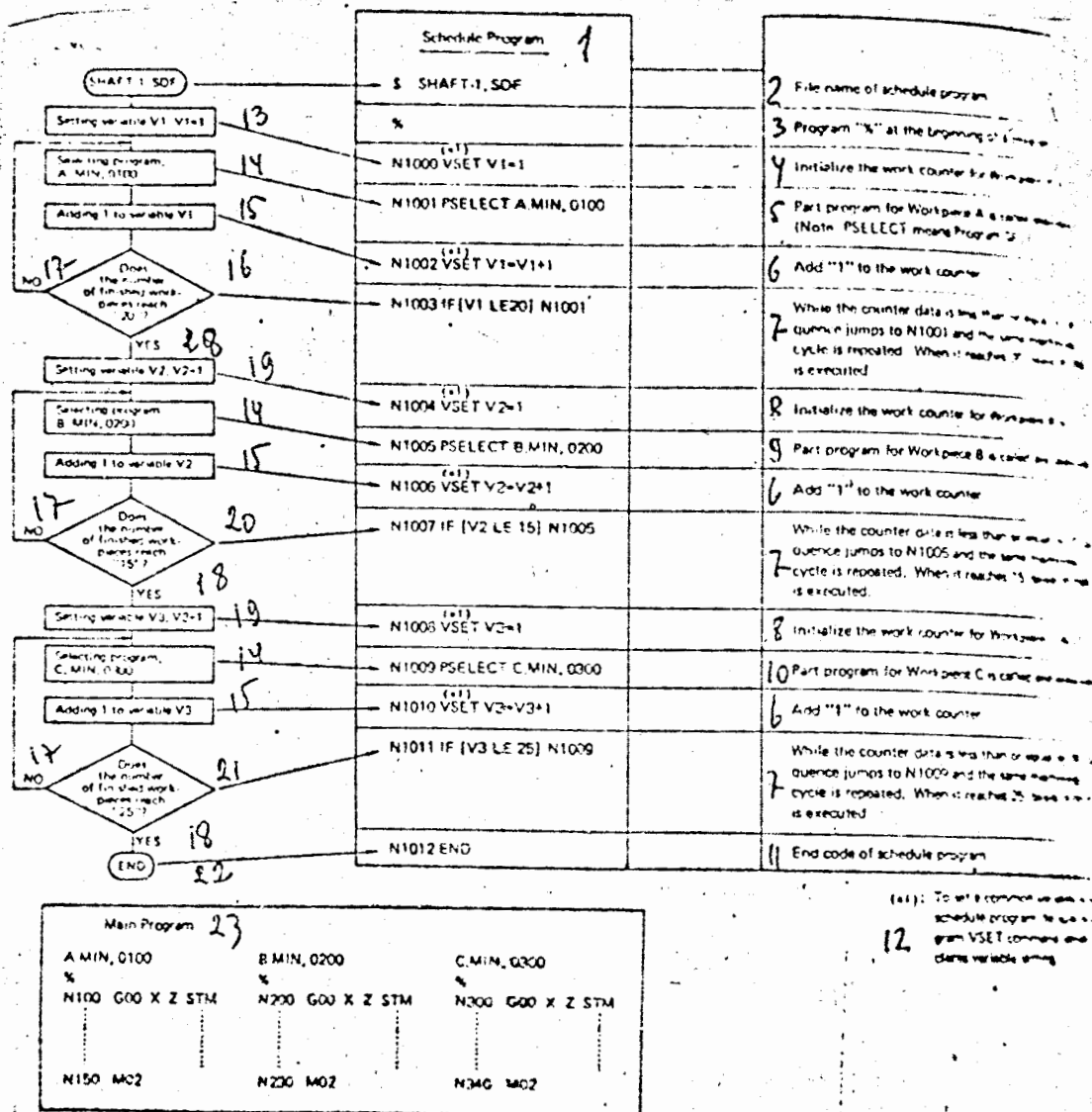


Рис. 182:

I - программа последовательности, 2 - наименование файла про-

граммы последовательности, 3 - программирование на начальном этапе, 4 - инициализация подсчета деталей, 5 - управляющая программа для указанной заготовки, 6 - добавление I на счетчике деталей, 7 - если величина подсчета менее указанной, то производится перенос последовательности на указанное значение, а цикл повторяется, 8 - инициализация подсчета деталей, 9 - управляющая программа для заготовки B, 10 - управляющая программа для заготовки C, 11 - коц окончания программы графика, 12 - для настройки общего переменного производится использование указанных данных, 13 - настройка переменного параметра, 14 - выбор программы, 15 - добавление I указанному переменному, 16 - достигло ли количество обработанных деталей цифры 20, 17 - нет, 18 - да, 19 - настройка указанного переменного параметра, 20 - достигло ли число обработанных деталей цифры 15, 21 - достигло ли число обработанных деталей цифры 25, 22 - конец, 23 - главная программа

16.4. Сравнение задачи №1 для пользователя и задачи №2 для пользователя.

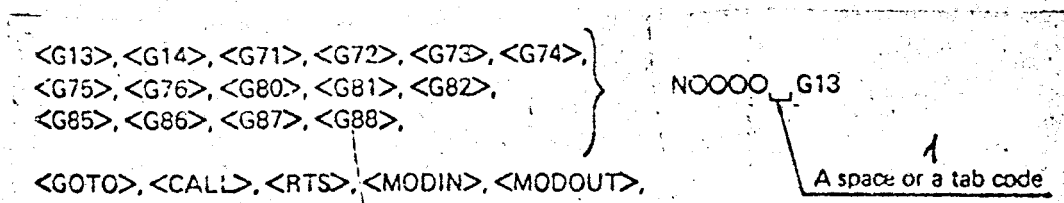
Function and Contents		User Task 1	Таблица 24 User Task 2
1		2	3
Usable Program	5	Main program 13 Schedule program 14	Main program 13 Sub program 19 Schedule program 14 System sub program 20
Control Statement Function	6	< GOTO Statement > < IF Statement > 15	< GOTO Statement > < IF Statement > < CALL Statement > < RTS Statement > < MODIN Statement > 15 < MODOUT Statement >
Variable Function	7	Common variables 16 Local variables 17	Common variables 16 Local variables 17 System parameters 21 I/O parameters 22
8 rational Function	Calculation expression 9	+, -, *, /, (four rules) 18	+, -, *, /, (four rules) 18
	Comparison expression 10	< LT >, < LE >, < EQ >, < NE >, < GT >, < GE >	< LT >, < LE >, < EQ >, < NE >, < GT >, < GE >
	Boolean expression 11		< OR >, < AND >, < EOR >, < NOT >
	Function 12		< SIN >, < COS >, < TAN >, < ATAN >, < ATAN2 >

1 - функция и содержание, 2 - задача №1 для пользователя,

3 - задача №2 для пользователя, 5 - используемая программа, 6 - функция сообщения для управления, 7 - переменная функция, 8 - функция обработки, 9 - расчетное выражение, 10 - выражение для сравнения, 11 - булево выражение, 12 - функция, 13 - главная программа, 14 - программа графика, 15 - сообщение, 16 - общие переменные параметры, 17 - местные переменные параметры, 18 - правило четырех действий, 19 - подпрограмма, 20 - подпрограмма системы, 21 - параметры системы, 22 - параметры ввода-вывода

Дополнения

Программировать промежуток или код разделения после кодов G и перечисленных ниже сообщений для управления. В кадре нужно программировать код G на начальном этапе или сразу же после наименования последовательности в кадре с промежутком или с введением кода промежутка.



I - промежуток или код промежутка

Раздел I7. Задача №2 для пользователя

I7.I. Описание задачи для пользователя

Функция задачи для пользователя разработана таким образом, чтобы пользователь мог наиболее эффективно использовать возможности системы OCP 5000I.

В запоминающем устройстве находятся операции и функции, выполненные в виде единой группы инструкций с учетом наименования программы или подпрограммы. Находящиеся в запоминающем устройстве подпрограммы имеют доступ от главной программы после указания наименования, представляющего группу инструкций, операций и функций, выполняемых с помощью этой программы.

Группа операций и функций, введенных в память, называется программой для задачи пользователя, а соответствующие команды определяются вызовом команд для задачи пользователя.

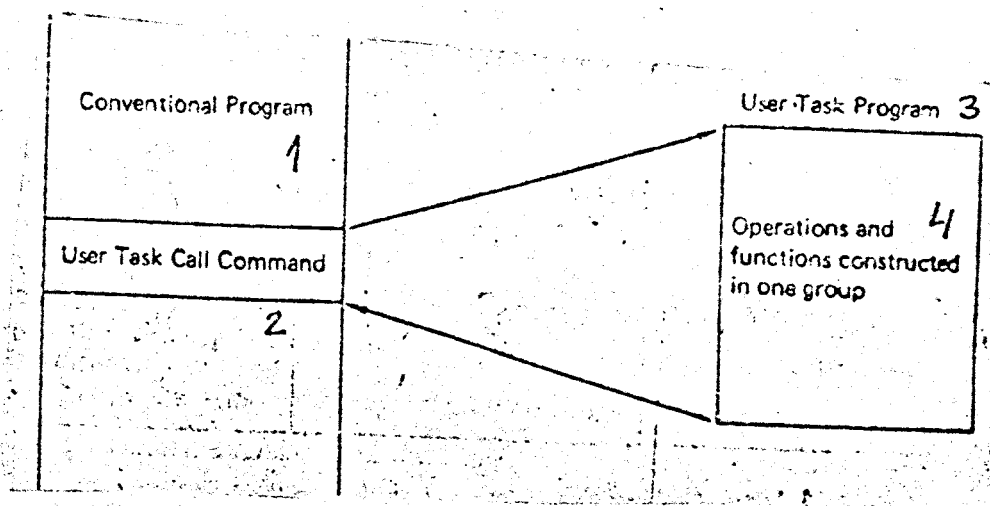


Рис. 183:

1 - обычная программа, 2 - команда вызова задачи пользователя, 3 - программа задачи пользователя, 4 - операции и функции, сведенные в одну группу

Отличительной особенностью является возможность применения различных оперативных функций и переменных параметров в программе задачи пользователя, а также возможность применения сообщений для управления с тем, чтобы расширить гибкость программирования.

Функция задачи пользователя может эффективно применяться в различных областях. Ниже указаны некоторые примеры, связанные с преимуществами, возникающими в результате введения функции задачи пользователя:

1) Повторное программирование одинакового контура при обработке детали типа шкива.

2) Обработка зубчатых колес и фланцев с одинаковым контуром.

При использовании общих и аналогичных элементов профиля обрабатываемых деталей в соответствии с функциями групповой технологии для обозначения этих элементов можно применять переменные параметры. Программирование задачи пользователя производится с помощью переменных параметров и фактических размеров конкретной обрабатываемой заготовки с учетом данных в главной программе обработки. При этом обработка деталей с одинаковым профилем может производиться с помощью одной программы для задачи пользователя.

3) Автоматический цикл с управлением периферийным оборудованием и функциями

В программу для задачи пользователя включены инструкции, необходимые для обеспечения связи между циклом работы станка и циклом работы питателя для прутковых заготовок или загрузочного устройства, разгрузочного устройства, с командами цикла измерения. Кроме того, обеспечивается связь с инструкциями, определяющими блокировку на станке и работу робота или другого периферийного оборудования. Функция задачи пользователя включается при работе станка в сочетании с периферийным оборудованием или другими операциями.

Фирма "Окума" поставляет перечисленное выше периферийное оборудование для обеспечения указанных функций. При этом пользователь может применять специальные функции или операции за счет использования функции задачи пользователя.

4) Детали с одинаковым профилем

Если на чертеже детали не указываются размеры точек, в которых пересекаются дуги окружности, или дуги окружности с коническими участками, однако имеется возможность рассчитать эти точки с помощью ряда уравнений, то программирование задачи пользователя для этих деталей может выполняться с помощью указанных уравнений.

5) Специальные постоянные циклы для пользователя

Применение в тех случаях, когда для пользователя необходимы специальные постоянные циклы.

Как указывалось выше, при эффективном применении функции задачи пользователя можно осуществить различные операции и функции. Кроме того, можно значительно упростить сложные программы за счет применения этой функции при обеспечении точного и быстрого программирования.

17.2. Типы функций задачи для пользователя

17.2.1. Связь между файлами программ и функцией задачи пользователя

В инструкции по эксплуатации системы OCP 5000L, раздел 4 "Применение" рассматриваются типы файлов с программами. На рис. 184 приводятся итоговые данные, определяющие связь между файлами программ и функциями.

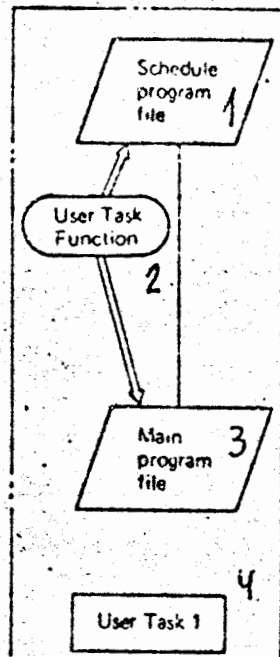


Рис.184:

1 - файл программы графика, 2 - функция задачи пользователя, 3 - файл главной программы, 4 - задача для пользователя

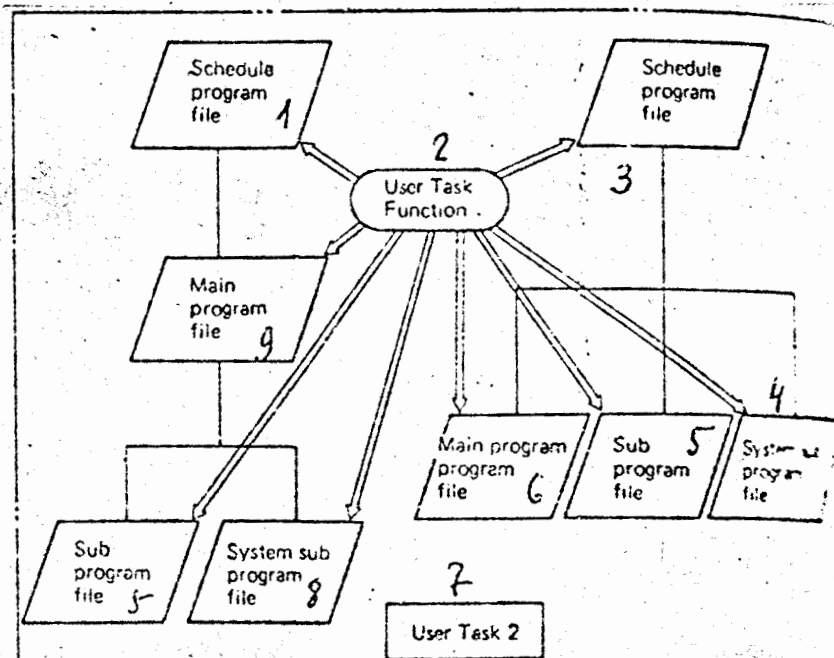


Рис.185:

1 - файл с программой графика, 2 - функция задачи пользователя, 3 - файл программы графика, 4 - система программирования с файлом

лом, 5 - файл подпрограммы, 6 - файл главной программы, 7 - задача №2 для пользователя, 8 - файл подпрограммы системы, 9 - файл главной программы

Программы на рис.184 представляют собой программу графика и главную программу. Функция программы пользователя может вводиться в программы указанных двух типов. Функция программы пользователя, применяемая в подобной схеме, называется задачей №1 для пользователя.

Вариант, представленный на рис.185, состоит из файлов программ и функций задач пользователя, которые могут применяться в любом типе программ. Функция задачи пользователя, применяемая в подобной схеме, называется задачей №2 для пользователя

В данной инструкции рассматривается задача №2 для пользователя, причем функции, характерные для задачи №1 для пользователя, введены в задачу №2 для пользователя.

17.3. Основные функции для задачи пользователя

Ниже перечислены три основные функции для задачи пользователя:

1) Функции сообщения для управления

Сообщение для контроля последовательности исполнения при запрограммированной очередности, например IF, GOTO и CALL, не вызывает затруднений.

2) Переменные параметры

При обычном программировании числовые данные непосредственно следуют за знаками адресования, например X, Y и Z. Вместо цифровых данных с знаком адресования могут быть связаны алфавитно-цифровые значения, а фактические цифровые значения связаны с переменными параметрами в соответствующих программах. Эта особенность обеспечивает гибкость и универсальность программы.

Пример

Word used in conventional program

X 135

1

When a variable is used:

2 $X = XP1$

$XP1 = 135$

3 Variable name
(alphanumeric)

4 Numerical data
is assigned

1 - слово, используемое в обычной программе, 2 - при использо-

вании переменного параметра, 3 - переменное наименование (ал-
фавитно-цифровое), 4 - присвоенное цифровое обозначение

3) Функция арифметических действий

Эта функция позволяет непосредственно программировать ариф-
метические выражения в виде данных слова.

Пример

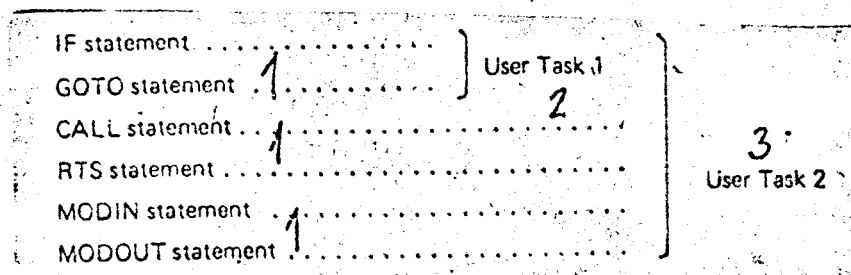
Word used in conventional program	1	2	When arithmetic operation function is used:
X 135			X = 100 + XP2 XP2 = 35

1 - слово, используемое в обычной программе, 2 - при использо-
вании функции арифметического действия

Ниже приведены подробные описания различных базовых
функций для задачи пользователя.

17.3.1. Сообщение для управления

Предусмотрено шесть сообщений для управления, используе-
мых в функции задачи для пользователя.



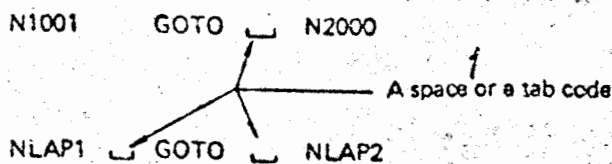
1 - сообщение, 2 - задача №1 для пользователя, 3 - задача №2
для пользователя

Программирование перечисленных выше сообщений для управ-
ления осуществляется либо в начале кадра, либо сразу же после
запрограммированного наименования последовательности в начале
кадра. Обязательно предусмотреть промежуток или код пропуска
после наименования последовательности или использовать сооб-
щение для управления в качестве разделения. В противном случае
появляется тревожное сообщение.

Для сообщения IF не обязательно предусматривать интервал

или промежуток, так как затем следует скобка.

Пример:



I - промежуток или код интервала

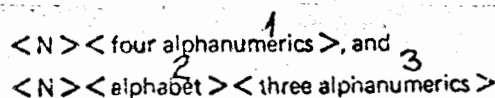
После элемента, содержащего два или более знаков адресования (от А до Z), например, в наименовании последовательности, должен указываться код управления с последующим промежутком или кодом интервала.

Примечание I. Наименование последовательности

Наименование последовательности представляет собой код для идентификации соответствующих кадров в программе, причем четыре алфавитно-цифровых знака следуют за знаком адресования

N.

Предусмотрены два типа наименований последовательности:

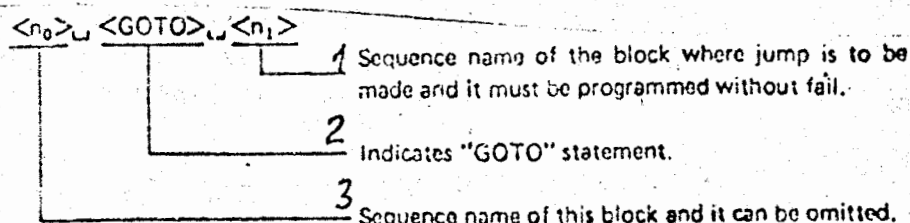


I - четыре алфавитно-цифровых знака, 2 - алфавитный знак, 3 - три алфавитно-цифровых знака

Наименование последовательности, указанное по этой схеме, может относиться к обоим вариантам наименования последовательности.

I7.3.III. Сообщение GOTO, переход без условий

I) Формат программы



I - наименование последовательности в кадре, на котором осуществляется переход и предусматривается обязательно программирование, 2 - индикация сообщения GOTO, 3 - наименование последовательности в указанном кадре может быть пропущено

Примечание. Наименование последовательности $\langle n_1 \rangle$ для перехода должно быть в программе с указанием подобного сообщения для управления.

2) Функция

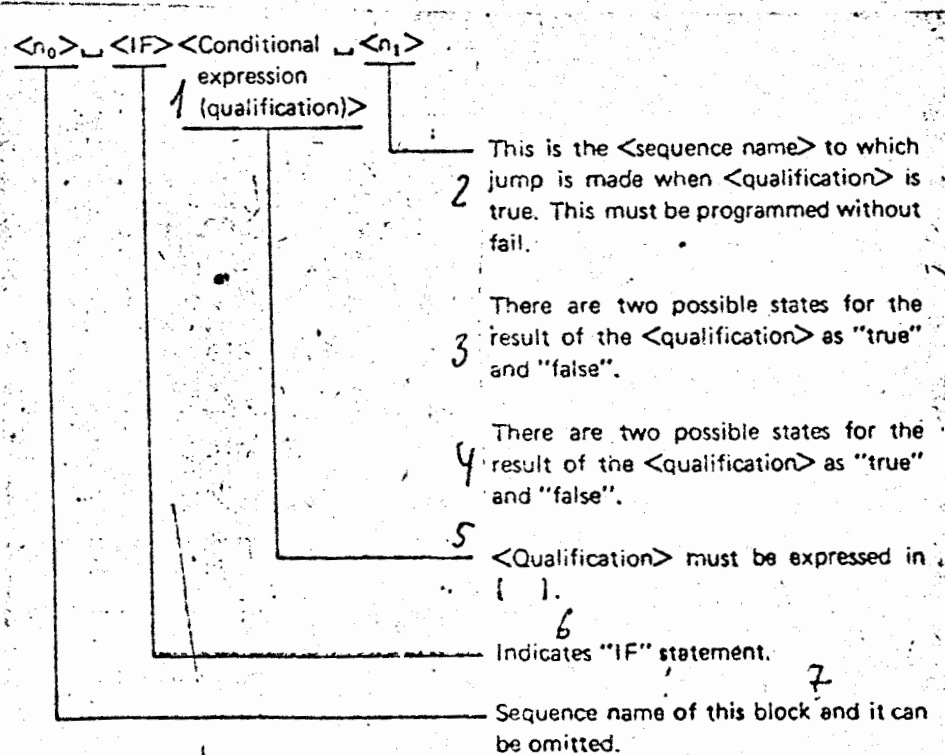
При осуществлении сообщения для управления переход к запрограммированному кадру $\langle n_1 \rangle$ осуществляется без условий.

Замечания

На ставках с двумя револьверными головками невозможен переход от программы для револьверной головки А к программе для револьверной головки В. Невозможен также переход для последовательности от программы револьверной головки В к программе для револьверной головки А.

17.3.1.2. Сообщение IF, переход с условиями

1) Формат программы



1 - обусловленное выражение (квалификация), 2 - это соответствует наименованию последовательности, в которой осуществляется переход при правильной квалификации. Программирование является обязательным, 3 - существуют два возможных варианта при действительной и ложной квалификации, 4 - существуют два возможных состояния для действительной и ложной квалификации, 5 - необходимость выражения квалификации, 6 - обозначение сообщения, 7 - наименование последовательности в кадре можно опустить.

2) Функция

а) В случае правильной квалификации выполняется последовательность переходов в соответствии с <ПГ>.

б) В случае ложной квалификации выполняется приведенная ниже последовательность.

Example:

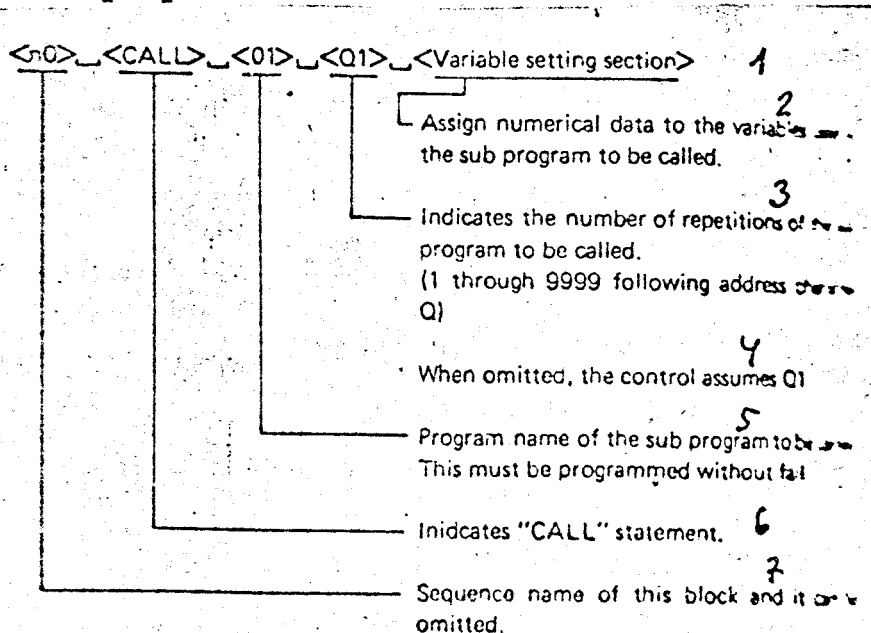
1 N1000 IF (V1 EQ 10) N2000 2
Means "equal (=)".

1 - пример, 2 - обозначает равенство

Осуществляется переход к №2000, если переменное V1 равняется 10 (V1 = 10). Если это переменное не равняется 10, то выполняется следующий кадр.

17.3.1.3. Вызов сообщения - вызов программы

1) Формат программы



1 - секция настройки переменного, 2 - обозначение числового значения переменной величины для вызова подпрограммы, 3 - индикация числа повторений вызываемой программы (если после знака адреса Q указываются значения 1-9999, 4 - при отсутствии знака в системе управления считывается Q1, 5 - наименование программы для выполняемой подпрограммы. Обязательно программировать указанное значение, 6 - индикация сообщения вызова, 7 - наименование последовательности в указанном кадре,

может быть опущено

2) Функция

Осуществляется вызов и исполнение подпрограммы, обозначенной ОI. Если переменные параметры находятся в секции настройки переменных, то все они подлежат регистрации.

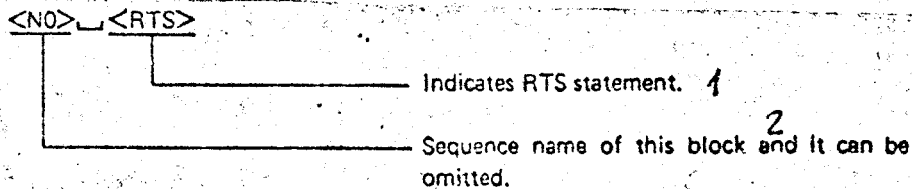
Пример

```
N1000 CALL 01234 XPI = 150 ZPI = 100 LF
```

На основании перечисленных выше запрограммированных команд осуществляется вызов и исполнение подпрограммы 01234. Одновременно регистрируются переменные XPI и ZPI.

17.3.1.4. Сообщение RTS - конец подпрограммы

1) Формат программы



1 - индикация сообщения RTS, 2 - наименование последовательности для кадра, может быть опущено

2) Функция

Сообщение RTS обязательно программируется в конце подпрограммы, при этом определяется окончание вызываемой подпрограммы, а выполняемая последовательность перебрасывается на кадр, расположенный непосредственно после кадра с сообщением "вызов".

При выполнении сообщения RTS стираются переменные параметры, зарегистрированные командами в кадре, в котором содержится сообщение вызова, а также параметры, зарегистрированные в подпрограмме (см. приведенный ниже пример).

Пример

Main Program

```
1 N1000 CALL_01234_XP1 = 150_ZP1 = 100
  N1001 G00 X Z
  :
  :
  :
```

2

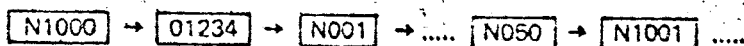
Sub Program

```
01234
N001 G00 X Z
:
:
:
N050 RTS
```

1 - главная программа, 2 - подпрограмма

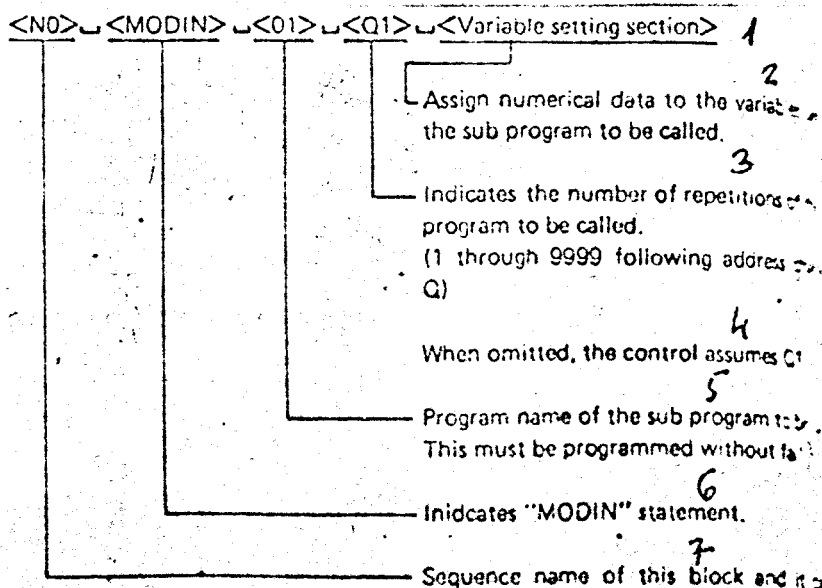
При выполнении кадра N1000 в главной программе осуществляется переход к подпрограмме 01234. Подпрограмма выполняется от N001, а после считывания системой управления сообщения RTS в кадре N050 производится перебор к кадру N1001 в главной программе и к командам этого кадра после исполнения предшествующих команд. В момент перехода от подпрограммы к главной программе стираются переменные XP1 и ZP1.

Последовательность выполнения программы



17.3.1.5. Сообщение MODIN

1) Формат программы



1 - секция настройки переменных, 2 - обозначение числового

значения переменного параметра в вызываемой подпрограмме, 3 - индикация количества повторений вызываемой подпрограммы (от 1 до 9999 после знака адресования 0), 4 - при отсутствии системы управления учитывает значение 01, 5 - наименование программы в вызываемой подпрограмме, 6 - индикация указанного сообщения, 7 - наименование последовательности соответствующего кадра, возможен пропуск

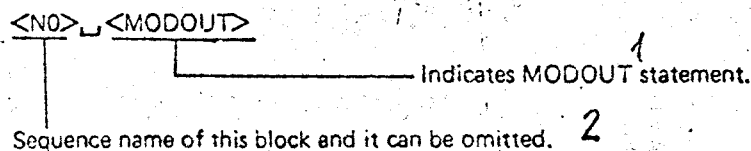
2) Функция

Вызов обозначенной подпрограммы осуществляется каждый раз, когда выполняется команда на перемещение вдоль оси. Таким образом, вызов и исполнение предусмотренной подпрограммы осуществляется каждый раз, когда выполняется команда *I в программе вызова подпрограммы. Эта функция остается действующей до тех пор, пока не произойдет считывание сообщения MODOUT (подробное описание см. в п.3.1.6.).

*I - команда на перемещение вдоль оси соответствует командам G00, G01, G02, G03, G35, в которых содержатся одно или два слова X и Z .

17.3.1.6. Сообщение MODOUT

1) Формат программы



1 - индикация указанного сообщения, 2 - наименование последовательности указанного кадра, возможен пропуск

2) Функция

Это сообщение обеспечивает отмену режима MODIN

3) Пример программы с сообщениями MODIN и MODOUT

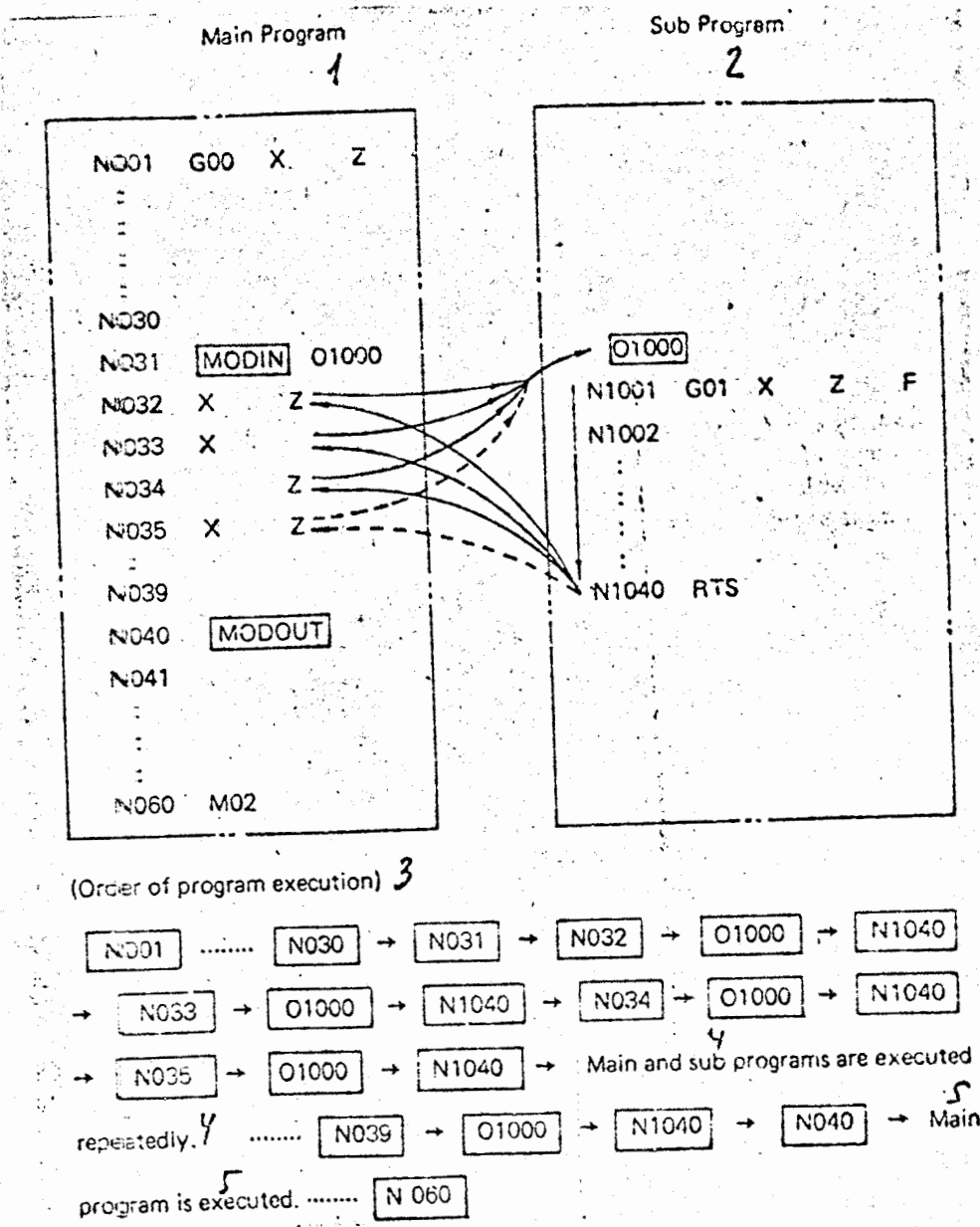


Рис.186:

1 - главная программа, 2 - подпрограмма, 3 - последовательность выполнения программ, 4 - выполнение главной программы и подпрограммы с повторяющейся очередностью, 5 - исполнение главной программы

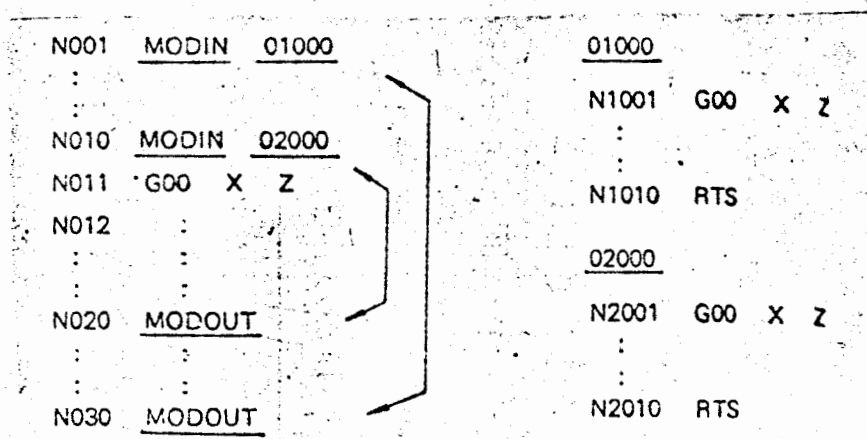
Программа начинается с N001 в главной программе, причем исполнение команд вплоть до N030 осуществляется по обычной

схеме. После выполнения команд до № 031 система управления находится в режиме MODIN с вызовом подпрограммы 01000. Однако в этом кадре не происходит исполнение подпрограммы. После завершения команд на перемещение вдоль оси в кадре №032 производится вызов и исполнение подпрограммы 01000 вплоть до № 1040 в подпрограмме. Сообщение RTS обеспечивает переход к главной программе, а затем выполняется кадр № 033 в главной программе. Аналогичные функции повторяются вплоть до кадра 039 в главной программе. Сообщение MODOUT в кадре № 040 отменяет режим MODIN, а исполнение команд в кадрах после №041 производится по обычной схеме.

Примечание I

Расположение и эффективный диапазон допустимого уровня режимов MODIN /MODOUT соответствует 8.

Пример



Приведенный выше пример содержит два групповых уровня. Режим MODIN является действительным от № 001 до №030 в подпрограмме 01000 и до № 020 в подпрограмме 02000.

Последовательность операции

1) В кадрах от №001 до №009 осуществляется вызов подпрограммы 01000 при каждом выполнении команды на перемещение вдоль оси.

2) В кадрах от №010 до № 020 вызов подпрограммы 01000 осуществляется первый раз после выдачи команды на перемещение вдоль оси, затем осуществляется последовательный вызов и исполнение подпрограммы 02000. В том случае, когда в подпрограмме 02000 содержатся команды на перемещение вдоль осей, например, команда № 2001, как показано в приведенном примере, то испол-

нения программы 01000 осуществляется после завершения указанных команд на перемещение вдоль оси. После завершения подпрограммы 02000 осуществляются команды в главной программе.

3) Вызов и исполнение подпрограммы 01000 в кадрах после № 021 и до №030 осуществляется каждый раз при выполнении команды на перемещение вдоль оси.

Примечание 2

Отмена режима MODIN осуществляется с помощью сообщения MODOUT, которое запрограммировано в той же программе. Таким образом, если режим MODIN активизируется в главной программе, то его нельзя отменить сообщением MODOUT в подпрограмме, а если режим MODIN активизируется в подпрограмме, то его нельзя отменить сообщением MODOUT в главной программе.

Примечание 3

Уровень размещения в подпрограмме

Существует восемь уровней размещения подпрограмм с учетом вызова подпрограммы с помощью сообщения CALL или MODIN

17.3.2. Переменные

В задачу №2 для пользователя предусмотрены переменные трех типов:

- Общие переменные
- Местные переменные
- Параметр системы

Использование и характеристики этих переменных и параметров различаются.

17.3.2.1. Общие переменные

Термин "Общий" для обозначения общих переменных можно понимать буквально, так как он является общим для главной программы и подпрограмм. Если одно и то же переменное используется для двух или более программ, то номер переменного используемого в этих программах должен всегда быть одинаковым. Таким образом, общее переменное, найденное в результате расчета для одной программы, должно использоваться во всех других программах.

1) Формат программы

<V> <one or two digits> = numerical data or expression

1 - одна или две цифры, 2 - цифровое значение или выражение

Обозначение общего переменного осуществляется с помощью двух цифр после знака U. Используемые общие переменные имеют обозначение от U1 до U32.

В программе общее переменное используется в следующем виде:

```
N101 V5 = 10, or  
N101 V5 = V5 + 1
```

Примечание 1. Общие переменные используются совместно для револьверных головок А и В.

Примечание 2. Общие переменные действительны как для главной программы, так и для подпрограммы.

Примечание 3. На общие переменные не влияет перенастройка системы управления или отключение питания. При этом все данные сохраняются до их повторной настройки или загрузки ленты с управляющей программой.

Примечание 4. Настройка или изменение общих переменных осуществляется введением параметра помимо ввода или изменения в соответствии с программой. Подробные сведения, относящиеся к настройке параметра, приведены в п.4.3. в Инструкции по эксплуатации системы OSP 5000.

17.3.2.2. Местные переменные

Как указывает термин "местный" местные переменные являются параметрами, которые пользователь вводит при необходимости облегченного распознавания для отличия друг от друга. Для револьверных головок А и В используется до 127 местных переменных.

1) Формат программы

< Letter > < Letter > < two alphanumerics > = Numerical data or expression
1 1 2 3

1 - буква, 2 - два буквенно-цифровых знака, 3 - числовые параметры или выражения

"0", "N" and "V" cannot be used. 1

Example "DIA1" "ITH5"

1 - пример, 2 - не допускается использование указанных знаков

Не допускается использование такого же наименования, как у относительных операторов, перечисленных в п. 2.3.2. или использования расширенного адресного знака * I.

* I - расширенные знаки адресования используются для автоматизированного программирования IAP, для обработки по шаблону, для постоянных циклов и других расширенных функций, которые не охватываются обычными знаками адресования от A до Z. Ниже перечислены расширенные знаки адресования.

<AA>, <AB>, <DA>, <DB>, <FA>, <FB>, <IA>, <IB>, <KA>, <KB>, <LA>, <LB>, <RA>, <RB>, <SA>, <SB>, <TA>, <TB>, <UA>, <UB>, <WA>, <WB>, <XA>, <XB>, <ZA>, <ZB>,

2) Характеристики местных переменных

а) местные переменные могут использоваться независимо для револьверных головок A и B. При перенастройке системы управления производится стирание местных переменных.

б) Если в главной программе вводится новое местное переменное с указанием данных, относящихся к новому наименованию местного переменного, то в запоминающем устройстве регистрируется наименование этого местного переменного и соответствующее значение.

В том случае, когда наименование местного переменного указывается без последующих данных, включается тревожное сообщение.

в) Если новые данные присваиваются местному переменному, которое ранее зарегистрировано с другими данными, то прежние данные обновляются.

```
Main Program 1
:
N0010 DIA1 = 160
:
:
N0049
N0050 DIA1 = 200
:
```

2 In N0010, numerical data "160" is assigned to local variable name "DIA1", which remains effective up to sequence N0049. In N0050, new numerical data "200" is assigned to the same local variable name "DIA1". This clears the old data "160" and it is substituted with the new data "200".

I - главная программа, 2 - в кадре N0010 цифровое значение 160 относится к наименованию местного переменного DIA1 и со-

тается эффективным вплоть до последовательности № 0049. В кадре № 0050 прежнему наименованию местного переменного DIAI присваивается новое цифровое значение 200, а прежнее значение 160 стирается для замены новым значением 200.

г) Для револьверных головок А и В соответственно может использоваться до 127 местных переменных параметров.

д) Если в главной программе запрограммирована команда вызова подпрограммы (сообщение CALL), а подпрограмма имеет набор местных переменных в кадре с сообщением CALL, то все переменные, имеющие цифровые значения в этом кадре регистрируются в виде местных переменных, а их числовые значения вводятся в запоминающее устройство.

Даже в том случае, когда местные переменные, имеющие такое же наименование, как и предыдущие переменные, зарегистрированы до введения запрограммированного сообщения CALL, производится их повторная регистрация в виде нового переменного.

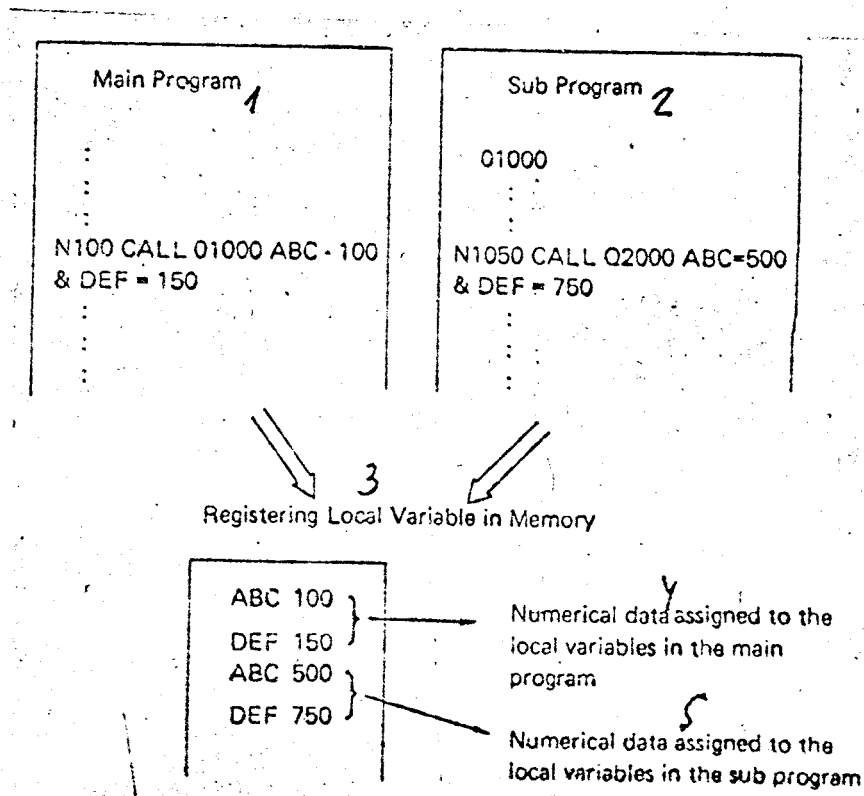


Рис.187:

1 - главная программа, 2 - подпрограмма, 3 - регистрация мест-

ных переменных в запоминающем устройстве, 4 - числовые данные, относящиеся к местным переменным в главной программе, 5 - числовые данные, относящиеся к местным переменным в подпрограмме.

Как показано выше, переменные, имеющие такое же наименование, как и переменные, зарегистрированные ранее, повторно регистрируются в виде других переменных.

е) При использовании местных переменных в вызываемой подпрограмме применяются числовые данные, относящиеся к последнему местному переменному в том случае, когда с одинаковым наименованием, зарегистрированным в запоминающем устройстве, связано несколько местных переменных.

Производится стирание местных переменных, находящихся в кадре с сообщением CALL (вызов) после выполнения сообщения RTS в вызываемой подпрограмме.

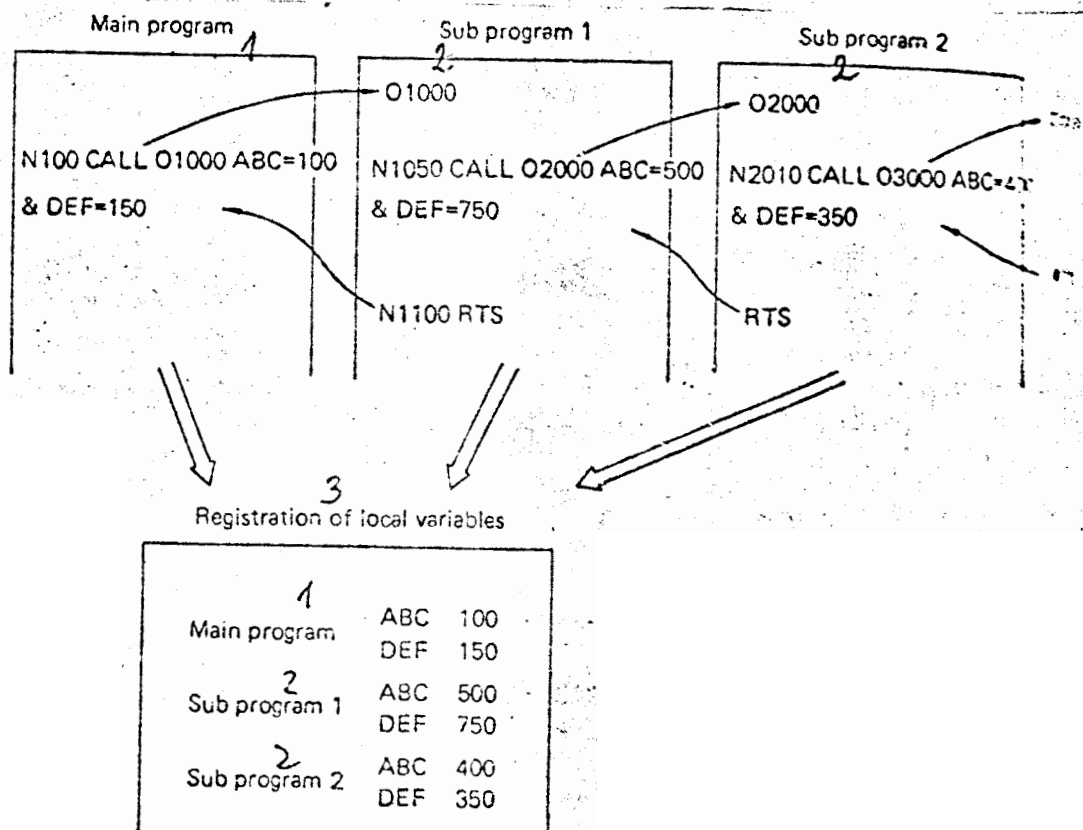


Рис. 188:

1 - главная программа, 2 - подпрограмма, 3 - регистрация местных переменных

В приведенном выше примере при исполнении кадра №2010 в подпрограмме 2 происходит регистрация местных переменных ABC и DEF для исполнения в подпрограмме 03000. Если в подпрограмме 03000 содержатся наименования местных переменных ABC и DEF, то осуществляется вызов числовых значений, зарегистрированных в последнюю очередь, т.е. ABC = 400 и DEF = 350. В конце подпрограммы, т.е. в конце подпрограммы 03000 при осуществлении сообщения RTS в подпрограмме 03000 регистрируются местные переменные, содержащиеся в подпрограмме 02000 и стираются значения ABC = 400 и DEF = 350 в запоминающем устройстве.

ж) Если в подпрограмму вводится новое местное переменное, то в запоминающем устройстве регистрируется его наименование и числовое значение. Эти переменные эффективны только в подпрограмме, в которую они введены, а после исполнения сообщения RTS в соответствующей подпрограмме производится их стирание.

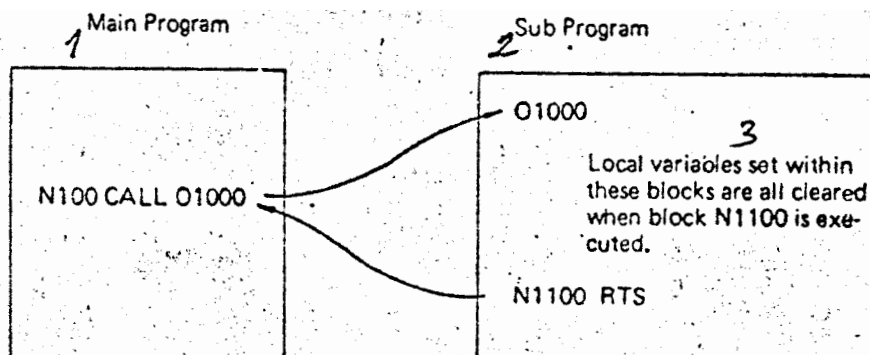


Рис. 189:

1 - главная программа, 2 - подпрограмма, 3 - местные переменные, введенные в указанные кадры, стираются при исполнении кадра №1100

з) Если числовые значения относятся к наименованию местного переменного, которое уже имеет другие числовые данные в процессе выполнения подпрограммы, происходит обновление числовых данных. Если в запоминающем устройстве зарегистрировано несколько местных переменных, имеющих одинаковое наименование, то обновляются числовые данные местного переменного, зарегистри-

стрированно в последнюю очередь.

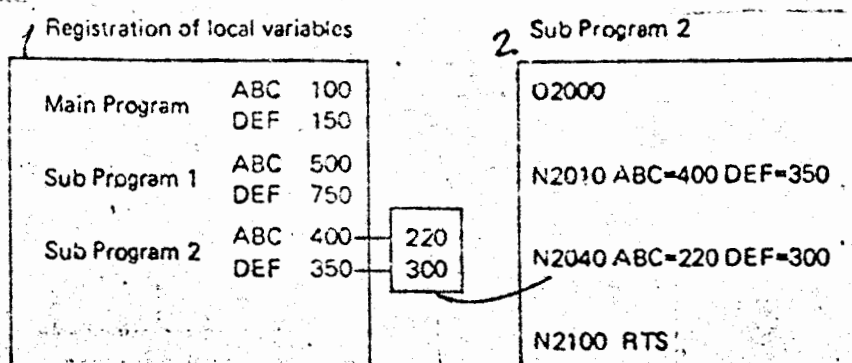


Рис.190:

1 - регистрация местных переменных, 2 - подпрограмма

В процессе выполнения подпрограммы 2 в запоминающем устройстве регистрируются местные переменные ABC = 400 и DEF = 350 когда выполняется кадр № 2010. Эти цифровые значения эффективны вплоть до кадра № 2039. Однако при исполнении кадра 2040 производится регистрация числовых значений местных переменных ABC и DEF в подпрограмме 2 с обновлением до значений 220 и 300 соответственно. Вместе с тем, не обновляются значения, зарегистрированные в подпрограмме 1 и в главной программе.

17.3.2.3. Параметры системы

Параметром системы называется параметр, характерный для соответствующих систем.

Параметры системы вводятся независимо для соответствующих револьверных головок и не стираются при перенастройке системы управления.

Предусмотрены следующие параметры системы:

- Параметр нулевой настройки
- Параметр нулевого сдвига
- Параметр сдвига режущего инструмента
- Параметр компенсации радиуса закругления вершины резца
- Параметр с данными зацева режущего инструмента
- Параметр изменяемых предельных значений
- Параметр барьера для патрона
- Параметр закругления

Эти параметры могут вводиться, меняться и использоваться в программе в соответствии с принятым форматом. Они могут эффективно использоваться в соответствующих программах, например, в программе измерения детали, в программе калибровочной проверки инструмента и в программе измерения после обработки.

Введение этих параметров может производиться с помощью режима "нулевая настройка", "данные, относящиеся к инструменту" и "режим параметра". Подробные сведения, относящиеся к операциям настройки, приведены в инструкции по эксплуатации системы OSP 5000L.

Ниже подробно рассматриваются параметры системы
Формат основной программы с параметром системы

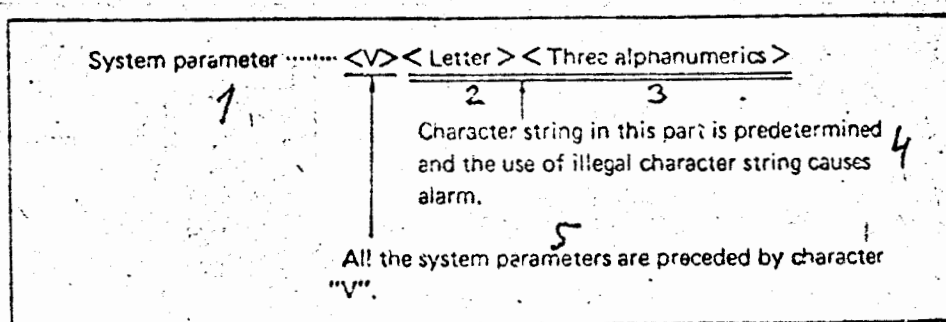


Рис. I91:

1 - параметр системы, 2 - буквенное обозначение, 3 - три алфавитно-числовых знака, 4 - цепочка знаков в этой части определяется заранее, а при использовании ошибочной цепочки знаков включается тревожная сигнализация, 5 - перед всеми параметрами системы указывается знак "V"

1) Параметр нулевой настройки

Наименование параметра

< VZOFZ >

Нулевая настройка по оси Z

< VZOFX >

Нулевая настройка по оси X

При настройке параметра вводится значение VZOFZ = 12364, 256

2) Параметр нулевого сдвига

Наименование параметра

VZSHZ

Нулевой сдвиг по оси Z

VZSHX

Нулевой сдвиг по оси X

При настройке параметра программируется VZSHZ = 50

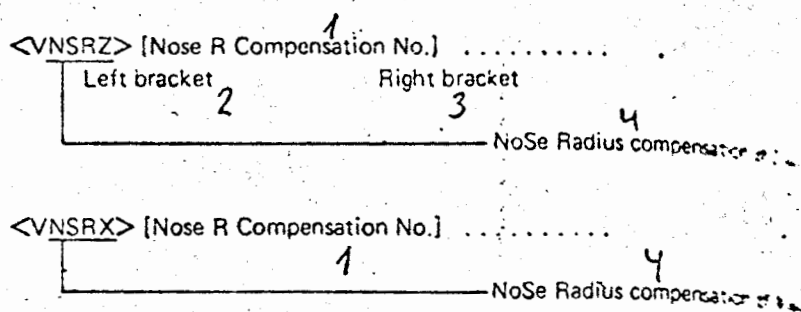
Параметр нулевого сдвига определяет величину сдвига, связанную с функцией нулевой настройки с вызовом от C50, причем при перенастройке системы управления стирается величина сдвига.

3) Параметр сдвига режущего инструмента

Наименование параметра (<VTOFZ> (номер сдвига инструмента)
левая скобка правая скобка
Сдвиг инструмента по оси Z
<VTOFX> (номер сдвига инструмента)
Сдвиг инструмента по оси X

При настройке параметра программируется VTOFZ (5) = 2,634
Это показывает, что величина сдвига инструмента по оси Z устанавливается равной 2,634.

4) Параметр компенсации радиуса закругления вершины резца
Наименование параметра



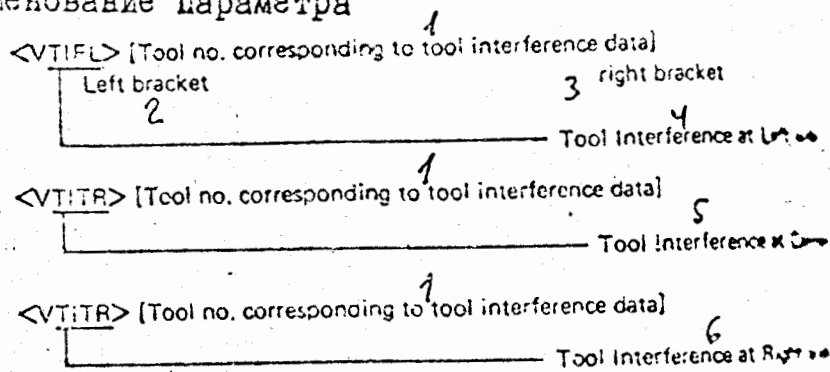
5
When setting a parameter, program as VNSRZ [4] = 0.8.

6
This indicates that nose radius compensation no. 4 is set to 0.8

1 - номер компенсации радиуса закругления вершины резца, 2 - левая скобка, 3 - правая скобка, 4 - номер компенсации радиуса закругления вершины, 5 - при настройке параметра программируется указанное значение, 6 - показывает радиус закругления с учетом компенсации

5) Параметр с данными задевания режущего инструмента

Наименование параметра



1 - номер инструмента, соответствующего данным задевания инструмента, 2 - левая скобка, 3 - правая скобка, 4 - задевание инструмента на левой стороне, 5 - задевание инструмента в средней части, 6 - задевание инструмента на правой стороне

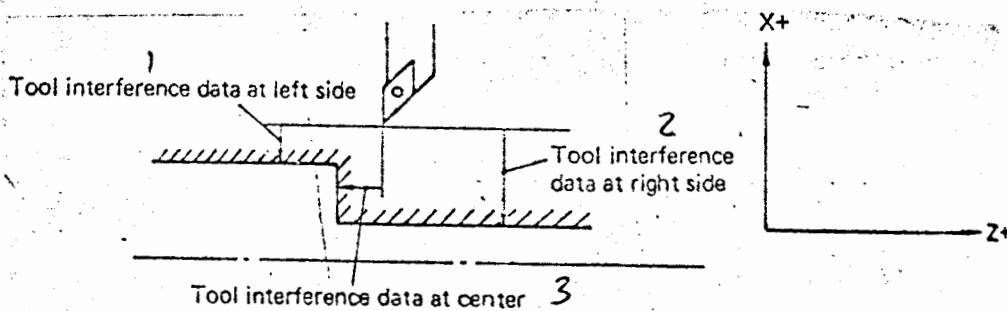


Рис.192 :

1 - данные задевания инструмента на левой стороне, 2 - данные задевания инструмента на правой стороне, 3 - данные задевания инструмента в средней части

При настройке параметра программировать VTIFL (3) = 25

6) Параметр изменяемых предельных значений

Наименование параметра

- <VPVLZ> - положительный предел по оси Z
- <VPVLX> - положительный предел по оси X
- <VNVLZ> - отрицательный предел по оси Z
- <VNVLX> - отрицательный предел по оси X

При настройке параметра программируется $VPVLZ = 2352,168$
 Это указывает, что установлено положительное направление
 переменного предельного значения по оси Z, равное $Z = 2353,168$

Цифровые данные указанных параметров относятся к началь-
 ной точке системы координат на станке (начальная точка на стан-
 ке).

7) Параметр барьера для патрона

Parameter name

- | | | | |
|---|-------------------|---|---|
| 1 | <VBPAZ> | Coordinate of Barrier Point A on Z-axis | 2 |
| | <VBPAZ> | Coordinate of Barrier Point A on X-axis | 2 |
| | <VBPBZ> | Coordinate of Barrier Point B on Z-axis | 2 |
| | <VBPBX> | Coordinate of Barrier Point B on X-axis | 2 |
| | <VBPCZ> | Coordinate of Barrier Point C on Z-axis | 2 |
| | <VBPCX> | Coordinate of Barrier Point C on X-axis | 2 |
| | <VBPDZ> | Coordinate of Barrier Point D on Z-axis | 2 |
| | <VBDPX> | Coordinate of Barrier Point D on X-axis | 2 |

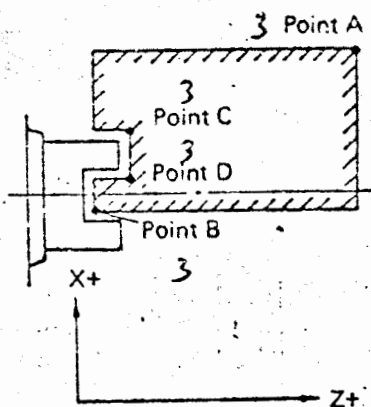


Рис.193:

I - наименование параметра, 2 - координата барьерной точки для
 указанной оси, 3 - обозначение точки

Числовые значения этих параметров относятся к началу за-
 программированной системы координат (запрограммированное нуле-
 вое значение).

8) Параметр закругления

Наименование параметра

- | | | |
|---------|-------------|----------|
| <VIMRZ> | - положение | по оси Z |
| <VIMRX> | - положение | по оси X |

17.3.2.4. Параметры ввода - вывода

Параметры ввода-вывода эффективны при передаче сигналов между центральным и периферийным оборудованием.

Входные параметры

Эти параметры обеспечивают сигналы, вводимые от периферийного оборудования, например, от панели управления, блока измерения после обработки, системы измерения режущего инструмента и от датчиков.

Эти сигналы называются входными битами данных.

Выходные параметры

Эти параметры обеспечивают сигналы, выходящие от системы управления и поступающие в периферийное оборудование, например, индикаторные лампы и тревожная индикация на панели управления.

Эти сигналы называются выходными битами данных

1) Входной параметр

Наименование параметра (см. табл. 25)

Таблица 25

<VDIN> [Input parameter no.] Data Input
 Left Bracket 1 Right Bracket 2
 3 4

5 Input Parameter No.	6 Contents of Data	7 Input Equipment
1~8	Bit data: 0 (OFF), 1 (ON) 8	Panel input 14
9	1 byte data in which data of parameters #1 through #8 are corresponded to bit 0 through bit 7. 9	Panel input 14
10	5 bits data of hexadecimal number, S0 through S1F 10	Panel input 14
11~18	Bit data: 0 (OFF), 1 (ON) 8	External input (EO)
19	1 byte data in which data of parameters #11 through #18 are corresponded to bit 0 through bit 7. 9	External input (EO) 15
21~22	Bit data: 0 (OFF), 1 (ON) 8	Panel input 14
23~24	Bit data: 0 (OFF), 1 (ON) 8	External input 15
1000~1004	Timers: 1 msec., 1 sec., 1 min., 1 day (with power turned on, timers are reset to "0" and starts counting up after that: 4-byte counter) 14	
1100~1123	Condition of external inputs: #1 through #24 12	
1124~1149	Condition of panel inputs: #1 through #16 13	
1150~1165	Condition of external outputs: #1 through #16 12	
1166~1194	Condition of panel outputs: #1 through #29 13	

1 - номер входного параметра, 2 - входные данные, 3 - левая

скобка, 4 - правая скобка, 5 - номер входного параметра, 6 - содержимое данных, 7 - оборудование ввода, 8 - биты с данными, 9 - данные для указанных параметров соответствуют бит 0- бит 7, 10 - данные для шестизначного числа, 11 - настройка таймеров (при отключении питания производится нулевой сброс таймеров с последующим возобновлением отсчета), 12 - состояние для указанных внешних выходов, 13 - состояние для указанных выходов панели, 14 - ввод для панели, 15 - внешний ввод

Примечание. Детали технических характеристик нужно согласовать с заказчиком.

2) Выходной параметр

Таблица 26

Parameter name
1 <VDDOUT>..... Data OUTPUT 2

Output Parameter No. 3	Contents of Data 4	Output Equipment 5
1~8 9	6 Bit data: 0 (OFF), 1 (ON) 7 1 byte data in which data of parameters #1 through #8 are corresponded to bit 0 through bit 7.	Panel output 8 Panel output 8
11~18 19	6 Bit data: 0 (OFF), 1 (ON) 7 1 byte data in which data of parameters #11 through #18 are corresponded to bit 0 through bit 7.	External output (EC) External output (EC) 9
21~22	6 Bit data: 0 (OFF), 1 (ON)	Panel output 8
23~24	6 Bit data: 0 (OFF), 1 (ON)	9 External output (EC)

1 - наименование параметра, 2 - выход данных, 3 - номер выходного параметра, 4 - содержание данных, 5 - оборудование выхода, 6 - биты данных, 7 - данные для указанных параметров соответствуют бит 0 - бит 7, 8 - выход для панели, 9 - внешний выход,

Примечание. Детали технических характеристик нужно согласовать с заказчиком.

17.3.3. Оперативная функция

Возможно выполнение арифметических действий с использованием переменных. Программирование при этом осуществляется по

аналогии с общим арифметическим выражением .

<Address character> 1, <Extended address character> 2, <Variables> 3 = <Expression> 4

1 - знак адресования, 2 - расширенный знак адресования, 3 - переменные, 4 - выражение

Выражение, в правой части предусматривающее арифметические действия, состоит из переменных параметров, сравнительного выражения и операторов.

Ниже приведены типы и особенности этих функций.

17.3.3.1. Арифметическое выражение (табл. 27)

Таблица 27

Operator 1	Meaning 2	Example 3	4 Rule and Remarks
+	Positive sign 5	+1234	
-	Negative sign 6	-1234	
+	Sum (addition) 7	X = 12.3 + V1	
-	Difference (subtraction) 8	X = 12.3 - V1	
.	Product (multiplication) 9	X = V10 . 10	
/	Quotient (division) 10	X = V11 / 10	

1 - оператор, 2 - значение, 3 - пример, 4 - правило или замечания, 5 - положительный знак, 6 - отрицательный знак, 7 - сумма (сложение), 8 - разность (вычитание), 9 - произведение (умножение), 10 - частное (деление)

17.3.3.2. Сравнительное выражение (табл. 28).

Operator 1	Meaning 2	Example 3	Таблица 28
LT	Less Than, < 6	IF [V1 LT 5] N100	Jump to N100 when V1 is less than 5. 12
LE	Less than or Equal to, ≤ 7	IF [V1 LE 5] N100	Jump to N100 when V1 is less than or equal to 5. 13
EQ	Equal to, = 8	IF [V1 EQ 5] N100	Jump to N100 when V1 is equal to 5. 14
NE	Not Equal to, ≠ 9	IF [V1 NE 5] N100	Jump to N100 when V1 is not equal to 5. 15
GT	Greater Than, > 10	IF [V1 GT 5] N100	Jump to N100 when V1 is greater than 5. 16
GE	Greater than or Equal to, ≥ 11	IF [V1 GE 5] N100	Jump to N100 when V1 is greater than or equal 17

1 - оператор, 2 - значение, 3 - пример, 4 - содержание,

5 - правило, 6 - меньше указанного значения, 7 - меньше или равняется, 8 - равняется, 9 - не равняется, 10 - больше указанного значения, 11 - больше или равняется, 12 - переход к №100, если У1 менее 5, 13 - переход к №100, если У1 меньше или равняется 5, 14 - переход к №100, если У1 равняется 5, 15 - переход к №100, если У1 не равняется 5, 16 - переход к №100, если У1 больше 5, 17 - переход к №100, если У1 больше или равняется 5, 18 - предусмотреть промежуток с обеих сторон оператора

17.3.3.3. Булево выражение (см. табл. 29)

Таблица 29

Символ	Meaning 2	Example 3	Rule 4
OR	Logical sum 5	VDIN [11] OR VDIN [12]	Provide a space on either side of the operator. 9
AND	Logical product 6	VDIN [11] AND VDIN [12]	
EOR	Exclusive OR 7	VDIN [11] EOR VDIN [12]	
NOT	Negation 8		

1 - оператор, 2 - значение, 3 - пример, 4 - правило, 5 - местная сумма, 6 - логический продукт, 7 - исключающий параметр "или", 8 - отрицание, 9 - предусмотреть промежуток с обеих сторон оператора

17.3.3.4. ФУНКЦИИ

Таблица 30

1	2	3	4	5
Operator	Meaning	Unit system of command	Example	Rule and Remarks
SIN	Sine 6	Deg.	X = 15•SIN [22.5]	The function name is always followed by either a variable or figures; be sure to enclose them by brackets. When two elements are entered in brackets, separate them with a comma. 24
COS	Cosine 7	"	Z = 15•COS [22.5]	
TAN	Tangent 8	"	Z = 15•TAN [12.5]	
ATAN	Arc tangent 9 (from -90 deg. to +90 deg.)		X = 15•ATAN [22.5]	
ATAN2	Arc tangent 9 10 (used for argument which is expressed by coordinates (a, b); from -180 deg. to +180 deg.)		ATAN2 [10, 15]	
SQRT	Square root 11	-	X = 15•SQRT [224.5]	
ABS	Absolute value 12	-	ABS [V15]	
BIN	BCD → BINARY (code conversion) 13	-	BIN [V10] 4 Byte	
BCD	BINARY → BCD (code conversion) 13	-	BCD [V20] 4 Byte	
ROUND	Rounding off fractions 14	22 mm 13 Inch	ROUND [V5]	
FIX	Cutting off fractions 15	"	FIX [V7]	
FUP	Counting fractions as a whole number 16	"	FUP [V15]	
DROUND	Rounding off the fractions to three decimal places (metric system) or to four decimal places (inch system). 17	"	DROUND [V20]	
DFIX	Cutting off the fractions below the third place of decimals (metric system) or below the fourth place of decimals (inch system). 18	"	DFIX [V25]	
DFUP	Count the figures below the third place of decimals (metric system) or below the fourth place of decimals (inch system) as a whole number. 19	"	DFUP [V21]	
MOD	Remainder 20	-		

I - оператор, 2 - значение, 3 - система единиц в команде, 4 - пример, 5 - правило или замечания, 6 - синус, 7 - косинус, 8 - тангенс, 9 - арктангенс (в указанных пределах), 10 - используется в качестве аргумента, выражаемого указанными координатами, 11 - квадратный корень, 12 - абсолютное значение, 13 - преобразование кода, 14 - округление дробей, 15 - отсечка дробей, 16 - учет дробей как целого числа, 17 - округление дробей до трех десятичных знаков (в метрической системе) или

четырёх десятичных знаков (в дюймовой системе), 18 - отсечка дробей после третьего десятичного знака (в метрической системе) или четвертого знака (дюймовой системе), 19 - подсчет значений после третьего десятичного знака (в метрической системе) или после четвертого десятичного знака (в дюймовой системе), 20 - остальное, 21 - градусы, 22 - миллиметры, 23 - дюймы, 24 - после наименования функции всегда следует переменный параметр или число, обязательно вводить скобки. Если в скобках указаны два элемента, то их нужно разделять запятой

17.3.3.5. Комбинация операций

1) Возможно следующее сочетание операций и функций, перечисленных выше

$$X = V1 + V2 - V3 + V4 \cdot \cos [30]$$

2) Обозначение предшествующего оператора с помощью скобок
Первоочередность оператора обозначается с помощью скобок

Пример

$$V1 = \frac{[V2 + V3] \cdot V4 + \sin \left[\frac{[V5 - V6]}{V7} + V8 \right]}$$

17.4. Примеры программ

Ниже рассматриваются три типичных примера программы. Они составлены только для целей иллюстрации и не охватывают всех функций, выполняемых с помощью задачи пользователя № 2.

Для более полного ознакомления с функцией задачи пользователя при составлении программ рекомендуется ознакомиться с приведенными примерами.

PROGRAM EXAMPLE 17 FINISH WORK WITH SIMILAR CONTOURS

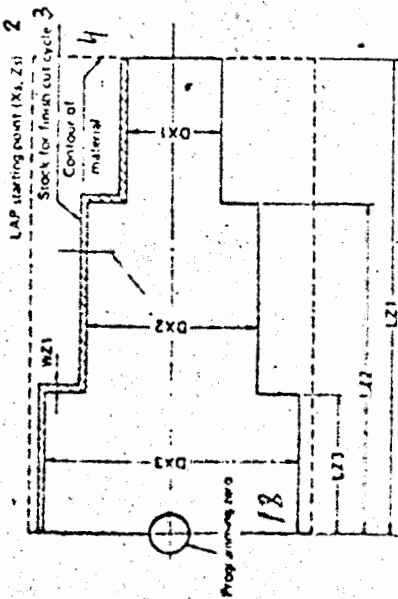


Fig. 1-1

Assuming three different workpieces having similar contours as shown above are to be cut, programs are prepared using user task function.

- (1) Assign respective workpieces with file names. **19**
- (2) Since there are three workpieces having similar contours, their contours are defined with a sub program. **20**
- (3) The elements (dimensional) used to define the contour, and the tool numbers and the cutting speeds are expressed using the local variables and the common variables, respectively. **21**
- (4) The starting point of each workpiece, LAP mode is used **28**

Table 1-1

Workpiece	Workpiece B	Workpiece C
Work A	Work B	Work C
SHAFT A	SHAFT B	SHAFT C
0101	0101	0101
0202	0202	0202
0303	0303	0303
0404	0404	0404
0505	0505	0505
0606	0606	0606
0707	0707	0707
0808	0808	0808
0909	0909	0909
1010	1010	1010
1111	1111	1111
1212	1212	1212
1313	1313	1313
1414	1414	1414
1515	1515	1515
1616	1616	1616
1717	1717	1717
1818	1818	1818
1919	1919	1919
2020	2020	2020
2121	2121	2121
2222	2222	2222
2323	2323	2323
2424	2424	2424
2525	2525	2525
2626	2626	2626
2727	2727	2727
2828	2828	2828
2929	2929	2929
3030	3030	3030
3131	3131	3131
3232	3232	3232
3333	3333	3333
3434	3434	3434
3535	3535	3535
3636	3636	3636
3737	3737	3737
3838	3838	3838
3939	3939	3939
4040	4040	4040

The sub program defining the contour, prepared using local and common variables, can be programmed as below according to Table 1-1.

```

$ SHAFT-ABC.SUB
N01000
M001 G81
N1001 G00 X=DX1 Z=LZ1+2
N1002 G01 Z=LZ7 F0.2
N1003 X=DX2
N1004 Z=LZ3
N1005 X=DX3 Z0
N1006
N1007 G80
N1010 G70 X=0 Z=0
N1011 G36 X=DX5 Z=LZ1 S=V3 T=V1 M08
N1012 G85 NLAPI D4 F0.35 U=UDX1 W=WLZ1
N1020 G00 X800 Z400
N1021
N1022 G87 NLAPI S=V4 T=V2
N1023 G00 X800 Z400 M05 M09
N1024 RTS
    
```

The main program for each workpiece is programmed as below using the sub program.

```

Workpiece A 16
$ SHAFT-A.MIN
N0100 CALL 01000 V1=0101 V2=0202 V3=100 V4=120 LZ1=300
LZ2=150 LZ3=60 $ DX1=50 DX2=50 DX3=50 WLZ1=01
UDX1=0.2 XS=100 ZS=210
    
```

```

Workpiece B 16
$ SHAFT-B.MIN
N0101 G00 X800 Z400
N1002 CALL 01000 V1=0303 V2=0404 V3=110 V4=130 LZ1=250
LZ2=170 LZ3=100 $ DX1=50 DX2=70 DX3=120 WLZ1=0.15
UDX1=0.25 XS=140 ZS=250
N1003 M02

Workpiece C 16
$ SHAFT-C.MIN
N0101 G00 X800 Z400
N1002 CALL 01000 V1=0505 V2=0606 V3=90 V4=150 LZ1=300
LZ2=200 LZ3=120 $ DX1=50 DX2=50 DX3=150 WLZ1=0.2
UDX1=0.3 XS=170 ZS=300
N1003 M02
    
```

File name of the main program ... Program \$ receiving the file name
Specify % code at the beginning of a program.
Program name ... 0100 in this example
With the command in N1002, the sub program (0100) is called and the value of the common variables for the sub program in the sub program is transferred to all set. When the command in this block is executed, the value of the common variables is transferred to the sub program. The value of the common variables is transferred to the sub program.

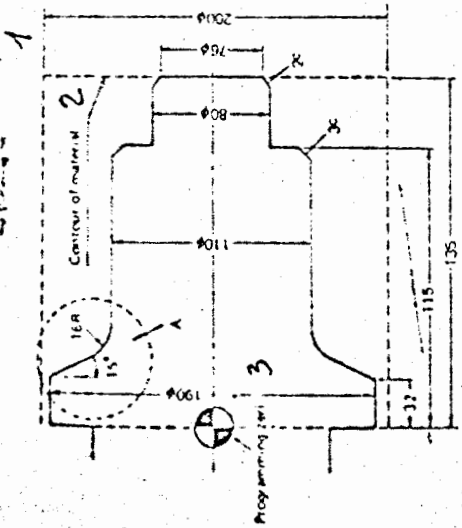


Fig. 2-1

When cutting the contour containing a circular arc and a taper and when the point(s) of intersection is not indicated on the part drawing, the user task function setting operation function is effectively used for preparing a program for such contour.

With entering section A, have the control calculate the points of intersection using the variables and the operation function of the user task.

Setting operation variables

V10 = Taper angle (15°)

V11 = Arc radius (16 mm)

Setting local variables

X01 = Diameter of point "a"

X02 = Diameter of point "b"

X03 = Diameter of point "c"

ZL1 = Z coordinate of point "a"

ZL2 = Z coordinate of point "b"

ZL3 = Z coordinate of point "c"

DIS1 = Distance DX-DX1

DIS2 = Distance between the center of arc and point "b" (along X axis)

DIS3 = Distance between point "a" and point "b" (along Z axis)

DIS4 = Distance between point "b" and point "c" (along Z axis)

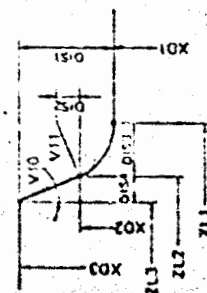


Fig. 2-2

The points that must be calculated are Z coordinate of point 'a' and Z coordinate of point 'b'. To obtain them, we solve the set of eqs.

```

%
O1000
DIS4 = (DIS1 + DIS2 - V11) * TAN(V10)
X02 = X01 + 2 * (V11 - DIS2)
ZL1 = ZL3 + DIS4
ZL1 = ZL2 + DIS3
  
```

- 16 (2) Since the pattern appearing in section A can be used in common with other workpieces, it is advisable to program such contour as a sub program.
- 17 Name such sub program "RADIUS-TAPER SUB" (file name).
- 18 Setting of variables <X0>, <ZL1>, and <ZL2> is made within this sub program, and other variables are set in the main program.

```

Sub Program
%
$ RADIUS-TAPER SUB
%
O1001
OR101
N1000 X02=X01+2*(V11-DIS2) ZL2=ZL3+DIS4 $ ZL1=ZL2+DIS3
N1001 G01 Z=ZL1
N1002 G02 X=X02 Z=ZL2 L=V11
N1003 G01 X=X03 Z=ZL3
N1004 RTS
  
```

- 20 Variables are set in block N1000
- 21 Z coordinate of point "a" is commanded in block N1001.
- 22 X and Z coordinates of point "b" and arc radius are commanded in block N1002
- 23 X and Z coordinates of point "c" are commanded in block N1003.
- 24 RTS in block N1004 indicates the end of the sub program.

```

%
$ FLANGE MAIN
%
O1000
N101 V10=15 V11=16 X01=110 X03=190 ZL1=37
N102 G00 X800 Z X0
N103 G00 X16 Z137
N104 G12 G01 Z137 F02
N105 G15 X10 L7
N106 G15 X115
N107 G15 X=X01 L3
N108 CALL OR101 O101=X03-X01/2 O102=V11-SIN(V10)
$ DIS1=V11-COS(V10) DIS1=(DIS1-O102-V11)*TAN(V10)
Z Z
N110 G40 G00 X270 110
N111 G90
N120 G00 X800 Z370
N121 G06 X20 Z137 $117 TO(0,0) (163) M04
N122 G03 HLAPI D6 F035 L02 W01
N123 G00 X800 Z330
N124 G00 X120 L02X02
N125 G87 NLAP2
N126 G00 X900 Z370
N127 M05 M09
N128 X02
  
```

Note: The sub program OR101 is called for by the commands in block N108 for defining the contour consisting of circular arc and taper. Variables used for defining such contour are all set in this block.

25
26

27

Рис.195. Пример 2 программы

I - обработка валов с одинаковым контуром, 2 - начальная точка для автоматизированного проектирования, 3 - припуск для цикла чистовой обработки, 4 - контур заготовки, 5 - обозначение на чертеже, 6 - общие и местные переменные, 7 - наименование файла для заготовки, 8 - тип заготовки, 9 - заготовка, 10 - вал, 11 - инструмент для черновой обработки, 12 - инструмент для чистовой обработки, 13 - скорость резания при цикле черновой обработки, 14 - скорость резания при цикле чистовой обработки, 15 - программирование подпрограммы для определения контура с использованием местных и общих переменных производится в соответствии с приведенными ниже указаниями, 16 - заготовка, 17 - вводится допущение о том, что три различные заготовки имеют аналогичный контур, а составление программы производится с помощью функции задачи пользователя, 18 - нулевая точка при программировании, 19 - указывать соответствующие наименования файла для заготовки, 20 - в связи с одинаковым профилем заготовок он определяется подпрограммой, 21 - наименование файла для подпрограммы, 22 - элементы (размеры), используемые для определения профиля и наименований или номеров инструментов, а также скоростей резания выражаются с помощью соответственно местных и общих переменных, 23 - продольный размер, 24 - диаметр, 25 - припуск на чистовую обработку в продольном направлении, 26 - припуск на чистовую обработку по диаметру, 27 - координата для начальной точки автоматизированного проектирования, 28 - для обработки заготовки используется режим автоматического программирования, 29 - программирование главной программы для каждой заготовки производится с помощью указанной подпрограммы, 30 - наименование файла для главной программы, 31 - указанная программа предшествует наименованию файла, 32 - указывать код в начале программы, 33 - наименование программы, 34 - вызов и исполнение подпрограммы производится с помощью указанной команды

К рис. I95 :

I - обрабатываемый контур требует уточнения, 2 - контур заготовки, 3 - нулевая точка программирования, 4 - если обрабатываемый контур содержит дугу окружности и конус и если точки пересечения не указаны на чертеже детали, то для подготовки программы обработки подобного контура весьма эффек-

тивным является использование оперативной функции задачи пользователя, 5 - при расширении указанной секции необходимо рассчитать точки пересечения на контуре с использованием переменных параметров и функции операции для задачи пользователя, 6 - ввод с помощью общего переменного, 7 - угол конуса, 8 - радиус дуги, 9 - ввод с помощью местных переменных, 10 - диаметр для указанной точки, 11 - координата для указанной точки, 12 - расстояние, 13 - расстояние от центра дуги до указанной точки, 14 - расстояние между указанными точками, 15 - точки, которые рассчитываются в виде указанных координат. Для нахождения используются переменные, представленные справа, 16 - шаблон на участке А может использоваться в качестве общего для других заготовок, поэтому рекомендуется программировать подобный контур в виде подпрограммы, 17 - указать наименование подпрограммы (наименование файла), 18 - настройка указанных переменных осуществляется в пределах подпрограммы, а другие переменные вводятся в главную программу, 19 - подпрограмма, 20 - переменные вводятся в указанный кадр, 21 - команда на координату точки в указанном кадре, 22 - координаты точки и радиуса вводятся командой в указанном кадре, 23 - координаты точки вводятся в указанном кадре, 24 - индикация окончания подпрограммы в указанном кадре, 25 - главная программа представлена ниже, 26 - главная программа, 27 - примечание. Вызов подпрограммы осуществляется с помощью команды в указанном кадре для определения контура, состоящего из дуги окружности и конуса. В этом кадре вводятся переменные параметры, определяющие указанный контур

К рис.196:

1 - обрабатываемый контур состоит из повторяющихся элементов, 2 - в этом примере программируется контур, состоящий из одинаковых повторяющихся профилей, например при обработке шкива, 3 - сделано допущение об использовании одинакового контура на различных шкивах. Для упрощения программирования обработки подобных шкивов профиль зоны А определяется с помощью переменных, 4 - сдвиг инструмента эффективен в указанной точке (программа составляется с использованием указанной точки в качестве контрольной), 5 - обозначение переменного, 6 - наи-

менованье, 7 - числовое значение для рассматриваемого примера, 8 - шаг между канавками шкива, 9 - высота канавки шкива, 10 - начальная точка резания, 11 - ширина канавки, 12 - величина конуса в указанном направлении, 13 - ширина резания, 14 - программировать цикл обработки канавки шкива в виде подпрограммы с использованием переменных. Вызов этой программы осуществляется после завершения команд на перемещение вдоль оси, поэтому используется ступенчатый режим, который можно использовать в любом положении в случае вызова, 15 - наименование файла подпрограммы, 16 - подпрограмма, 17 - в указанном этапе выполняется программа обработки одной канавки шкива. С помощью этой подпрограммы можно составить программу обработки всего шкива, 18-программа используется в качестве главной программы, 19 - наименование программы, 20 - главная программа, 21 - при указанном сообщении в кадре №004 система управления переключается на указанный режим, предусматривающий вызов и исполнение подпрограммы после каждого завершения команды на перемещение вдоль оси. В этом кадре введены также переменные, используемые для указанной подпрограммы, 22 - вызов и исполнение программы OPPI в указанных кадрах производится каждый раз после завершения команд на перемещение вдоль осей в этих кадрах для обработки канавок в шкивах, 23 - при использовании сообщения вызова вместо указанных сообщений обеспечивается обработка канавок шкива. Однако в этом случае сообщение нужно повторять каждый раз при вызове подпрограммы

17.5. Дополнения

17.5.1. Последовательность возврата в программе с использованием задачи пользователя

В принципе последовательность возврата может осуществляться по аналогии с обычной программой, причем отсутствуют ограничения для ввода функции последовательности возврата.

Если переменные введены в кадр, предшествующий кадру с последовательностью возврата, то все введенные данные регистрируются в запоминающем устройстве.

17.5.2. Режим одновременного управления по четырем осям
Выбор револьверной головки может осуществляться только в главной программе.

Коды G13 и G14 для револьверных головок обеспечивают вызов одинаковой подпрограммы. На рис.197 показан пример сочетания одновременного управления по четырем осям, автоматического программирования LAR, компенсации радиуса закругления при вершине резца и подпрограммы.

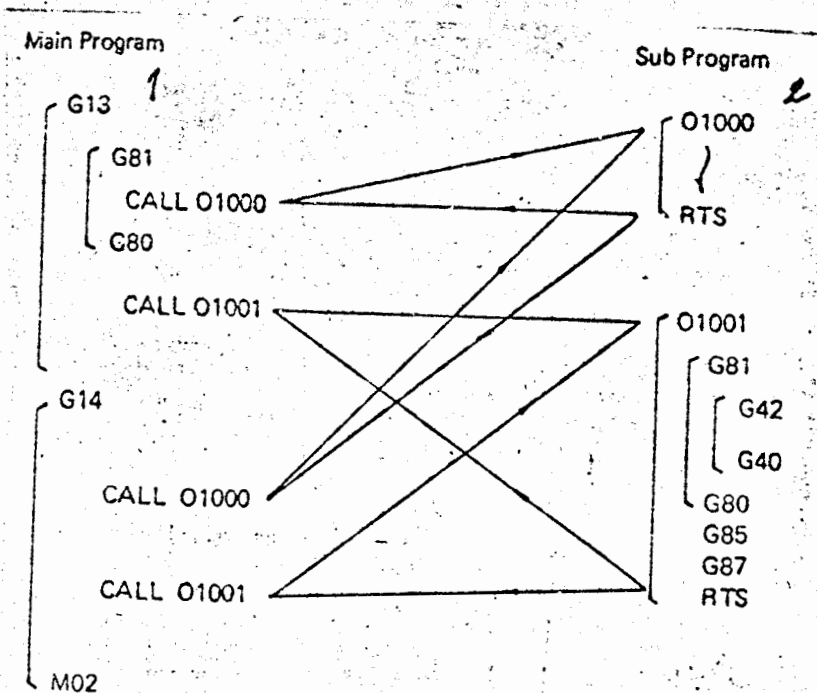


Рис.197:

1 - главная программа, 2 - подпрограмма

17.5.3. Правила работы и оценка результатов

1) Тип данных

Предусмотрены три типа данных: комплексные, реальные и логические.

а) Данные типа целых чисел

Данные типа целых чисел обеспечивают точное выражение целых чисел, причем предусматриваются как отрицательные, так и положительные целые числа.

б) Реальные данные

Эти данные реального типа достаточно точно выражают реальные значения и могут быть нулевыми, положительными или

отрицательными реальными числами.

в) Логические данные

Данные логического типа могут быть либо правильными (1), либо ложными (0).

2) Константы

Применяются константы двух типов ("целые константы" и "реальные константы").

Если перед константой указан знак "+" или "-", то они называются константами с знаками.

Подобные константы называются "образующими константами".

а) Целые константы

Целые константы носят характер целых чисел. Они выражаются несколькими знаками и рассматриваются как десятичные числа.

б) Реальные константы

Реальные константы относятся к реальному типу. Они выражаются несколькими знаками, в том числе десятичной запятой, и рассматриваются как десятичные числа.

3) Типы переменных и оценка значений

При введении переменного используется сообщение о вводе

$$V = e$$

где V - наименование переменного

e - константа, наименование переменного, выражение и функция

При указанной настройке производится оценка значения "e" и изменение значения V в соответствии с правилом.

Примечание: типы сокращений

[I] - целый тип

[R] - реальный тип

{ I } [R] - изменение на реальный тип

{ R } [I] - изменение на целый тип

1 Variable name of V	2 Unit system	3 Type of "e"	4 Evaluation of value
System parameter 5	1 μm 1/10000 inch	[I]	Not changed 10
		[R]	[R] → [I] (inch system value is converted into metric system value) (rounding off fractions) 12
	mm Inch	[I]	[I] • 1000 (metric system) 13 [I] • 10000 (inch system) 14
		[R]	[R] • 1000 → (metric system) (rounding off fractions) 13 [R] • 10000 → (inch system) (rounding off fractions) 14
Common variables 6	-	[I]	[I] → [R]
		[R]	Not changed 10
Local variables 7	-	[I]	[I] → [R]
		[R]	Not changed 10
Extended address character 8	-	[I]	[I] → [R]
		[R]	Not changed 10
I/O parameter 9	-	[I]	Not changed 10
		[R]	[R] → [I] (rounding off fractions) 12

I - наименование переменного, 2 - система единиц, 3 - тип, 4 - оценка значения, 5 - параметр системы, 6 - общие переменные, 7 - местные переменные, 8 - знак расширенного адреса, 9 - параметр ввода-вывода, 10 - не меняется, 11 - величина из дюймовой системы преобразуется в метрическую систему, 12 - округление дробей, 13 - метрическая система, 14 - дюймовая система

4) Правило выражения действия и оценки значений

Пример

Выражение: $C = A + B$

Результат действия (элемент 1) (элемент 2)

Примечание. Обозначение сокращений

- [I] - целый тип
- [R] - реальный тип
- [I][R] - изменение на реальный тип
- [R][I] - изменение на целый тип
- логический тип

Таблица 32

Type of expression 1	Operator 2	Meaning 3	Type of element 4 1 "A"	Type of element 4 2 "B"	Type of result of operation 5
6 Arithmetic expression	+	Addition 9	[I]	[I]	[I]
			[I] → [R]	[R]	[P]
	-	Subtraction 10	[R]	[I] → [R]	[R]
			[R]	[R]	[R]
		Positive sign 11	[I]	/	[I]
		Negative sign 12	[R]		[R]
	•	Multiplication 13	[I] [R]	[I] → [R]	}
	/	Division 14	[R]	[R]	
7 Comparison expression	LT	Less than < 15			}
	LE	Less than or equal to ≤ 16	[I]	[I]	
	EQ	Equal to = 17	[I] → [R]	[R]	
	NE	Not equal to ≠ 18	[R]	[I] → [R]	
	GT	Greater than > 19	[R]	[R]	
	GE	Greater than or equal to ≥ 20			
8 Logical expression	EOR	Exclusive OR 21	[I]	[I]	}
	OR	Logical sum. 22	[R] → [I]	[R] → [I]	
	AND	Logical product 23	(Cutting off fractions) 25	(Cutting off fractions) 25	
	NOT	Negation 24	[I] or [R] → [I] (Cutting off fractions) 25	/	[I]

I - тип выражения, 2 - оператор, 3 - значение, 4 - тип элемента, 5 - тип результата действия, 6 - арифметическое выражение, 7 - сравнительное выражение, 8 - логическое выражение, 9 - суммирование, 10 - вычитание, 11 - положительный знак, 12 - отрицательный знак, 13 - умножение, 14 - деление, 15 - менее чем, 16 - менее или равняется, 17 - равняется, 18 - не равняется, 19 - более, 20 - более или равняется, 21 - исключающее "ИЛИ", 22 - логическая сумма, 23 - логическое произведение, 24 - отрицание, 25 - отсечка дробей

5) Правило действия для функции и оценки значений

Примечание. Обозначение сокращений

- [I] - целый тип
- [R] - реальный тип
- [I] [R] - изменение на реальный тип
- [R] [I] - изменение на целый тип

Таблица 33

Function name 1	Meaning 2	Unit system 3	Type of element 4	Type of element 2y	Type of result of operation 5
SIN	Sine 6	1 μm 19 1/10000 inch	[I] → [R] [R]		[R]
COS	Cosine 7				
TAN	Tangent 8	mm 20 inch 21	[I] → [R] [R]	(degree)	[R]
ATAN	Arc tangent 9	1 μm 19	[I] → [R]		[R] · 1000 (1/1000 degree) (metric system) 22
ATNN2		1/10000 inch	[R]		[R] · 10000 (1/10000 degree) (inch system) 23
		mm 20 inch 21	[I] → [R] [R]		[R]
SORT	Square root 10	-	[I] → [R] [R]	-	[R]
ABS	Absolute value 11	-	[I] → [R] [R]	-	[R]
BIN	BCD → BIN	-	[I] [R] → [I] (Cutting off fractions)	2y	[I]
BCD	BIN → BCD	-	[I] [R] → [I] (Cutting off fractions)	2y	[I]
ROUND	Rounding off fractions 12	-	[I] [R]	-	[I] (Not changed) 25 [I]
FIX	Cutting off fractions 13	-	[I] [R]	-	[I] (Not changed) 25 [I]
FUP	Counting fractions as a whole number 14	-	[I] [R]	-	[I] (Not changed) 25 [I]
ROUND	Rounding off the fractions to three decimal places (metric) or to four decimal places (inch system). 15	1 μm 19	These functions cannot be used.	26	-
DFIX	Cutting off the fractions below the third place of decimals (metric) or below the fourth place of decimals (inch system). 16	1/10000 inch			
DFUP	Count the figures below the third place of decimals (metric) or below the fourth place of decimals (inch system) as a whole number. 17	mm 20 inch 21			
MOD	Remainder 18 (a - Fix (a/b) · b)	-	[I] [I] → [R] [R] [R]	[I] [R] [I] → [R] [R]	[I] [R] [R] [R]

I - наименование функции, 2 - значение, 3 - система единиц, 4 - тип элемента, 5 - тип результата действий, 6 - синус, 7 - косинус, 8 - тангенс, 9 - котангенс, 10 - квадратный ко-

резь, 11 - абсолютное значение, 12 - округление дробей, 13 - отбрасывание дробей, 14 - подсчет дробей как целого числа, 15 - округление дробей до трех десятичных знаков (метрическое значение) или четырех десятичных знаков (дюймовая система), 16 - отбрасывание дробей после третьего десятичного знака (метрическая система), или четвертого десятичного знака (дюймовая система), 17 - подсчет в виде целого числа цифр после третьего десятичного знака (метрическая система) или четвертого десятичного знака (дюймовая система), 18 - остаток, 19 - микроны, 20 - миллиметры, 21 - дюймы, 22 - градусы в метрической системе, 23 - градусы в дюймовой системе, 24 - отбрасывание дробей, 25 - не меняется, 26 - невозможность использования указанных функций

Раздел 18. Графический дисплей на электронно-лучевой трубке

18.1. Описание

Особенности графического дисплея на электронно-лучевой трубке обозначают программированную траекторию резца на дисплее в виде сплошных, пунктирных и точечных линий.

Ниже обозначается стандартная обработка, фиксация станка и режим обработки без детали.

- | | |
|--------------------------|------------------|
| - Подача при резании | сплошные линии |
| - Ускоренное перемещение | пунктирные линии |
| - Ручная подача | точечные линии |

Существует два варианта построения графиков траектории режущего инструмента: график для низкой скорости и график для высокой скорости. В режиме построения траектории резца при низкой скорости параметры отображаются на дисплее с той же скоростью, с которой осуществляется контроль перемещения вдоль осей, а при построении траектории резца с высокой скоростью индикация на дисплее осуществляется мгновенно.

18.2. Операции и команды

Функции графического дисплея являются активными в автоматическом, ручном режиме и при ручном вводе данных.

Для настройки режима графического дисплея нажать кнопку FS "Расширение" в режиме ручного ввода данных или дважды нажать кнопку FS для того, чтобы ввести ручной или автоматический режим.

Строка консоли на экране дисплея имеет следующий вид:

F1	F2	F3	F4	F5	F6	F7	F8
GRAPHIC	SCALE	G-ZERO	DELETE	PAGE	TURRET		EXTEND
1	2	3	4	5	6		7

Рис.198:

1 - графический, 2 - шкала, 3 - нулевой, 4 - исключение, 5 - страница, 6 - револьверная головка, 7 - расширение

При нажатии клавиши F1 "графический" устанавливается режим на графическом дисплее.

1) Кнопка F1 GRAPHIC (графический).

На дисплее устанавливается необходимый режим, а в систему управления поступают команды графического режима.

2) Клавиша F2 SCALE (шкала)

Используется для ввода единицы длины на графический дисплей. Величина шкалы появляется на нижней строке в секции с данными дисплея. Стрелка обозначается пунктирной линией ниже строки. Длина пунктирной линии соответствует единице длины, которая вводится командой шкалы.

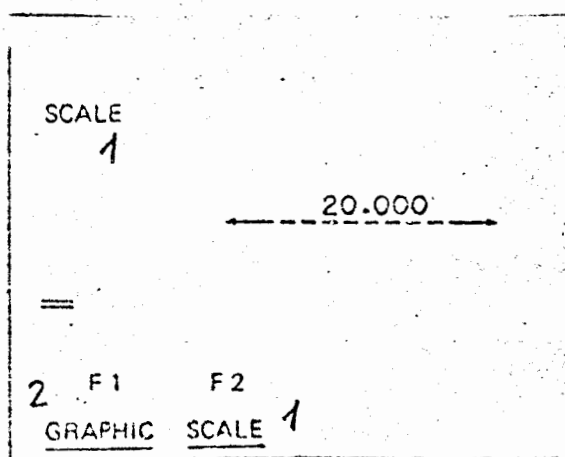


Рис.199:

1 - шкала, 2 - графический,

После нажатия клавиши F2 в строке консоли появляется инструкция SC. Звести величину устанавливаемой шкалы и нажать клавишу "Запись - исполнение". При этом вводится требуемая единица длины.

После настройки величины шкалы происходит повторное стирание экрана.

Пример

```
SC20 [WRITE/EXECUTION]
```

Данные эффективны для обеих револьверных головок (на станках с двумя револьверными головками и по обеим координатным осям X и Z).

3) Клавиша F3 G - ZERO

При этой команде определяется расстояние между начальной точкой графического дисплея и начальной точкой программы.

При нажатии на клавишу F3 в строке консоли появляется инструкция GO. После ввода требуемого расстояния нажать клавишу "Запись-исполнение".

Если предусмотрен сдвиг только по одной оси:

```
=GO X=10,Z=20 [WRITE/EXECUTION]
```

Если предусмотрен сдвиг по двум осям:

```
=GO X=10 [WRITE/EXECUTION]
```

```
=GO Z=20 [WRITE/EXECUTION]
```

После того, как установлена начальная точка на графическом дисплее, происходит стирание изображения на экране.

Настройка величины сдвига между начальной точкой на графическом дисплее и начальной точкой программирования производится независимо для соответствующих револьверных головок, поэтому для введения величины сдвига для индикации траекторий режущих инструментов револьверной головки В (только на моделях станков с двумя револьверными головками) нажать клавишу F6 (револьверная головка).

4) Клавиша F4 "Стирание"

Используется для стирания представленной траектории резца.

5) Клавиша F5 "Страница" используется для изменения страницы на дисплее, для индикации револьверной головки А и В .

6) Клавиша F6 "Револьверная головка"

Используется для выбора револьверной головки с целью обозначения траекторий режущих инструментов (только для станков с двумя револьверными головками).

Примечание. Положение осей на графическом дисплее может изменяться при вводе соответствующих параметрических данных для бита №5, как показано на рис.200.

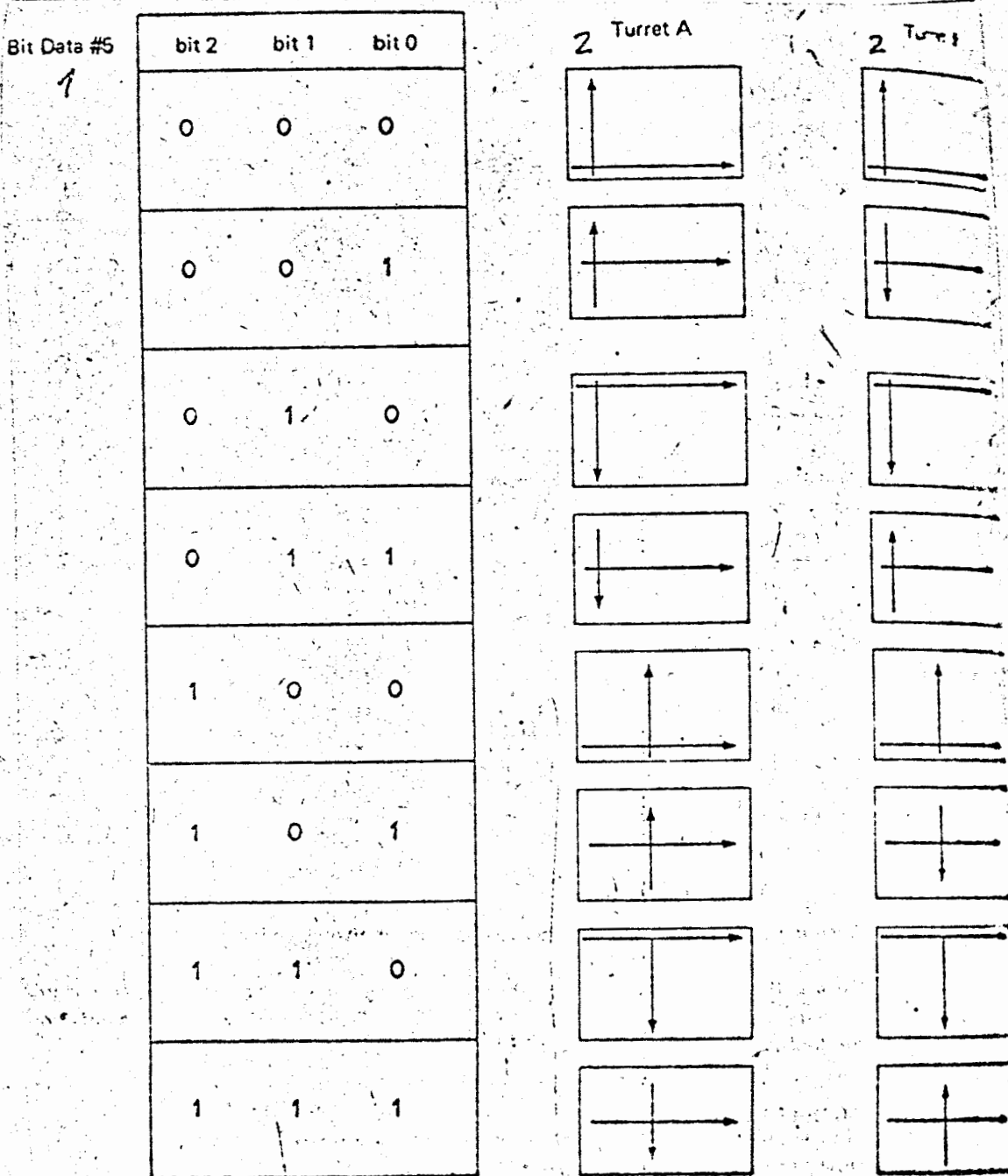


Рис.200:

1 - бит с данными, 2 - револьверная головка

18.3. Режим построения траекторий резца при большой скорости

Если введены функции блокировки ставка и обработки без детали, допускается построение траектории режущего инструмента при большой скорости. В этом режиме возможен выбор ввода и отключения функции единичных кадров.

Следует отметить, что режим СОС несколько отличается с точки зрения построения траекторий от фактического перемещения вдоль оси.

Раздел 19. Функция преобразования ленты (для систем ОЗР 3000L - ОЗР 5000L)

19.1. Описание

Эта функция используется для преобразования программ, предназначенных для системы ОЗР 3000L и хранимых в пузырьковой памяти, в программы для системы ОЗР 5000L.

Ниже перечислены преобразуемые элементы:

- 1) Программа одновременного управления по четырем осям
- 2) Программа постоянной скорости резания
- 3) Знаки адресования, определяющие шаг переменных параметров.
- 4) Работа без перерыва
- 5) Функция компенсации радиуса закругления при вершине резца.
- 6) Автоматическое программирование IAP.

19.2. Работа

Хранение программ, подготовленных для системы ОЗР 3000L, приведено в указаниях в п. 4.2.1. в Инструкции по эксплуатации системы ОЗР 5000L.

Преобразование программы выполняется с использованием следующих операций:

- 1) Нажать клавишу "Вспомогательное редактирование" в секции "Режим" (см. рис.201).

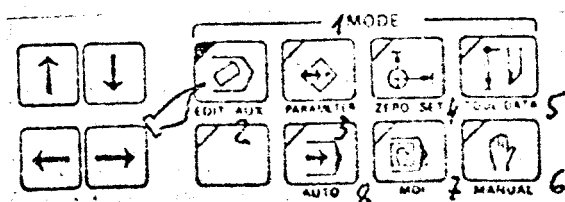


Рис. 201:

1 - режим, 2 - вспомогательное редактирование, 3 - параметр, 4 - нулевая настройка, 5 - данные, относящиеся к режущему инструменту, 6 - ручной, 7 - ручной ввод данных, 8 - автоматический

2) На экране появляется изображение, представленное на рис.202.

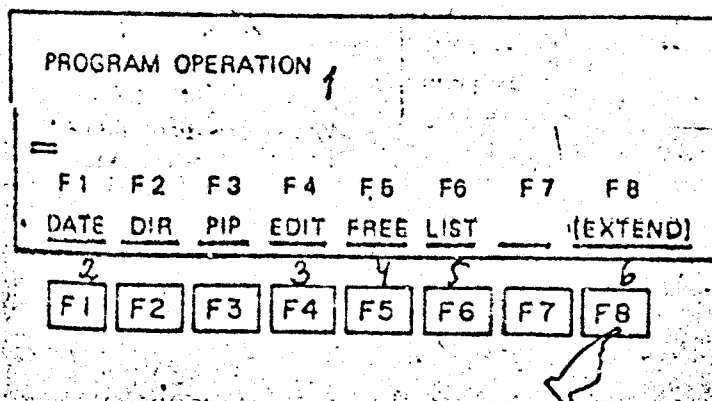


Рис.202:

1 - работа по программе, 2 - дата, 3 - редактирование, 4 - свободный, 5 - перечень, 6 - расширение

3) Нажать функциональную клавишу F8 "Расширение" два раза, так что на экране дисплея в строке функций происходят изменения, представленные на рис.203.

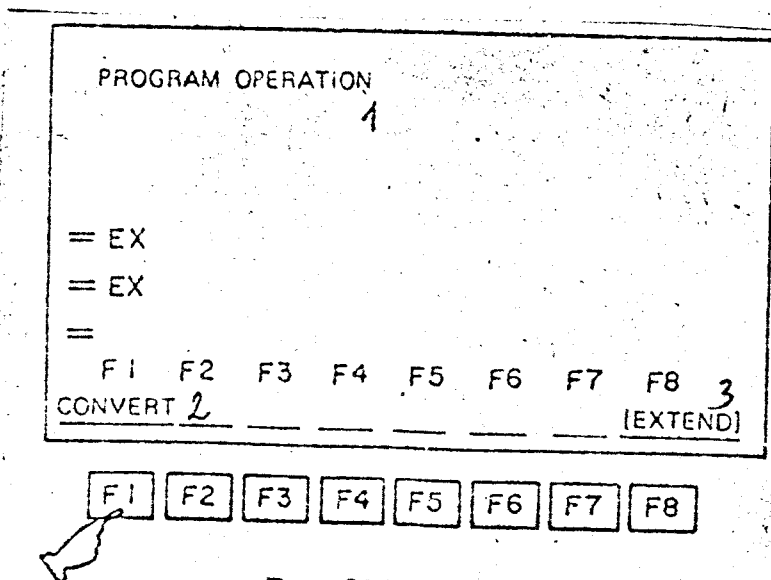


Рис.203:

1 - работа по программе, 2 - преобразование, 3 - расширение

4) Нажать функциональную клавишу F1 (преобразование)
На экране появляется знак "C="

5) Ввести "входное наименование файла" и "выходное наименование файла" с помощью переключателей клавиатуры.

6) Нажать клавишу "Запись-исполнение". Изображение на экране меняется в соответствии с рис.204.

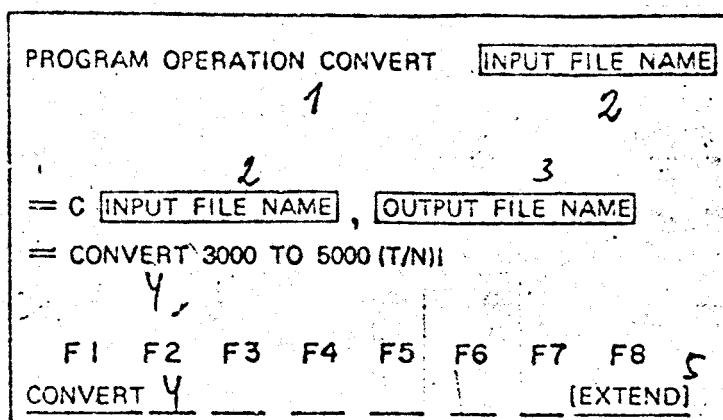


Рис.204 :

1 - преобразование работы по программе, 2 - входное наименование файла, 3 - выходное наименование файла, 4 - преобразование, 5 - расширение

7) Если требуется преобразование, нужно ввести знак "у", а затем нажать клавишу "Запись-исполнение". Если преобразование не требуется, нажать клавишу N вместо "у", а затем нажать клавишу "Запись-исполнение".

Внимание

1) Если обозначается только входное наименование файла, то формируется программа для системы ОР 5000L, имеющая такое же наименование файла, а программа для системы ОР 3000L стирается.

2) Для других случаев обозначается как входное наименование файла, так и выходное наименование файла.

3) Если входной файл обозначается наименованием, которое не удалось найти, то на экране появляется сообщение "Файл не найден" и операция преобразования ленты прекращается.

4) Если выходной файл обозначается наименованием, которое существовало до этого, то появляется сообщение "наличие наименования файла". Нажатие клавиши "Запись-исполнение" после ввода знака "у" стирает прежний выходной файл и вводится преобразование ленты.

При нажатии на клавишу "Запись-исполнение" после ввода знака N прекращается операция преобразования ленты до завершения преобразования.

5) Для проверки результатов программы преобразования нужно ознакомиться с указаниями в п.4.2.3. в Инструкции по эксплуатации системы OSP 5000I.

6) Если невозможно вычислить новое значение команды, формируемой в процессе преобразования ленты, то вводится знак "?" в зоне ввода подобных новых данных.

19.3. Содержание преобразования

19.3.1. Программа одновременного управления по четырем осям.

1) Секция одновременного управления по четырем осям:

а) Отменяется код M вызова ввода и окончания одновременного управления по четырем осям. При этом используются соответствующие коды MII и MIO.

б) Для прекращения программы и дополнительной остановки вводятся дополнительные коды M, например MOO и MOI, коды P с числовыми значениями "1", превышающими числовое значение предшествующего кода P для соответствующих револьверных головок при синхронизации исполнения кодов M.

в) Обновление кодов P, заранее запрограммированных для соответствующих револьверных головок, производится так, чтобы обеспечить выполнение синхронизации.

2) Секция независимого контроля

а) Для кодов G выбора револьверной головки (коды G13 и G14), для кода P, цифровое значение которого IO превышает числовое значение предшествующего кода P, вводятся дополнительные значения.

б) Первому коду P, указанному для режима независимой обработки после завершения одновременного управления по четырем осям, присваивается числовое значение IO, превышающее значение для кода P, имеющего наибольшее значение в предшествующей программе одновременного управления по четырем осям.

Примечание 1

Программирование кодов G13 и G14 осуществляется в одном кадре без других команд.

Примечание 2.

Программирование кодов MOO и MOI производится для револьверных головок А и В

<u>OSP3000L</u>				<u>OSP5000L</u>			
N001	M11		CR	N001			CR
N002	G13		CR	N002	G13		CR
N003	G00	X1 Z1 P1 M1 T1 S1	CR	N003	G00	X1 Z1 P10 M1 T1 S1	CR
}				}			
N014	M01		CR	N014	M01 P11		CR
N015	G14		CR	N015	G14		CR
N016	G00	X1 Z1 P1 M1 T2 S1	CR	N016	G00	X1 Z1 P10 M1 T2 S1	CR
}				}			
N024	M01		CR	N024	M01 P11		CR
N025	M10		CR	N025			CR
N026	G14		CR	N025	G14		CR
N027	G00	X2 Z2 T3 S2 M2	CR	N027	G00	X2 Z2 S2 M2 P21	CR

Рис.205. Пример программы

19.3.2. Программа для постоянной скорости резания

В программе одновременного управления по четырем осям определение револьверной головки с действующей постоянной скоростью резания (CCS) осуществляется выбором запрограммированной команды, которая действовала непосредственно до ввода одновременного управления по четырем осям, причем добавляется указанный ниже код C после кода C96 для вызова работы с постоянной скоростью резания (см. рис.206).

G110 CCS on turret A (G13)

G111 CCS on turret B (G14)

Example of Program: 2

<u>OSP3000L</u>				<u>OSP5000L</u>			
N001	M10		CR	N001			CR
N002	G13		CR	N002	G13		CR
}				}			
N009	M11		CR	N009			CR
N010	G13		CR	N010	G13		CR
N011	G00	X1 Z1 P10 S1 T1 M1	CR	N011	G00	X1 Z1 P10 S1 T1 M1	CR
N012	G96	X2 Z2 P20 S2	CR	N012	G96 G110	X2 Z2 P20 S2	CR
}				}			
N020	G14		CR	N020	G14		CR
N021	G00	X3 Z3 P10 S1 T1 M1	CR	N021	G00	X3 Z3 P10 S1 T1 M1	CR
N022	G96	X4 Z4 P20 S2	CR	N022	G96 G110	X4 Z4 P20 S2	CR
}				}			
N030	M10		CR	N030			CR

1 - для револьверной головки, 2 - пример программы

19.3.3. Знаки адресования для обозначения переменного шага

Знаки адресования, используемые для ввода команды на изменение величины шага при каждом шаге обработки резьбы с переменным шагом, преобразуются с изменением от К до Е.

<u>OSP3000L</u>					<u>OSP5000L</u>				
N001	G00	X1	Z1	CR	N001	G00	X1	Z1	CR
N002	G34	X2	Z2 K F	CR	N002	G34	X2	Z2 E F	CR
N003	G34		Z3	CR	N003	G34		Z3	CR
}				—	}				
N007	G00	X7	Z7	CR	N007	G00	X7	Z7	CR
N008	G33	X8	Z8 K F	CR	N008	G33	X8	Z8 E F	CR

Рис.207. Пример программы

19.3.4. Работа без перерыва

В процессе преобразования ленты код M31 преобразуется в код M02.

<u>OSP3000L</u>			<u>OSP5000L</u>		
N001	G13	CR	N001	G13	CR
}		—	}		
N009	M31	CR	N009	M02	CR

Рис.208. Пример программы

19.3.5. Функция компенсации радиуса закругления вершины резца

Для кодов G40, G00 вызова завершения функции компенсации радиуса закругления вершины резца при программировании для системы OSP 3000L формируется воображаемая точка (I, K на основании команд X2, Z2 в кадре с кодами G40, G00 и X1, Z при условии их наличия в предшествующем кадре с преобразованием ленты.

I - X2 - X1
Z - Z2 - Z1

1 Example of Program:

<u>OSP3000L</u>					<u>OSP5000L</u>				
N009	G01	X1	Z1	CR	N009	G01	X1	Z1	CR
N010	G40	G00	X2 Z2	CR	N010	G40	G00	X2 Z2 I K	CR

Note: ² G40 and G00 must be programmed in the same block. ³

Рис.209:

I - пример программы, 2 - примечание, 3 - необходимость программирования в одном кадре

19.3.6. Автоматическое программирование ЦАР

1) Знак адресования L, определяющий величину припуска, преобразуется в U или W. U - для продольного резания и W - для поперечного (торцевого) резания.

2) В программе для системы OSP 3000L режимы резания в цикле черновой обработки определяются вместе с командами соответствующему окончательному профилю. Для системы OSP 5000L команды, определяющие профиль и предусматривающие режимы черновой и чистовой обработки профиля, указываются отдельно от режимов чистового резания. В практических условиях резания осуществляется вызов условий чистового резания на основании команд, определяющих профиль.

<u>OSP3000L</u>						<u>OSP5000L</u>				
N001	G00	Xt	Zt	CR	N001	G00	Xt	Zt		CR
N002		Xs	Zs	Fr	Sr	Tr	Mr			CR
N003	G85								NLA01	U D
N004	G00	Xt	Zt	L	(U W)	D				CR
N005		Xa	Za	Ff	Sf	Tf	Mf			CR
N006	G01	Xb	Zb						G87	NLA01
N007		Xc	Zc						NLA01	G81
N008	G80								Xa	Zb
N009	G00	Xt	Zt						G01	Xb
									Xc	Zc
									G80	
									G00	Xt

Рис.210. Пример программы

3) Цикл нарезания резьбы по командам G33 осуществляется в комплексном постоянном цикле или в непрерывном цикле наре-

зания резьбы (коды G34 и G35) при осуществлении автоматического программирования IAP.

Для кода нарезания резьбы G33, который преобразуется в комплексный цикл нарезания резьбы, вводятся дополнительные коды G71 в зависимости от направления резания, продольного или поперечного резания.

<u>OSP3000L</u>					<u>OSP5000L</u>			
N001	G00	S1 T1 M1	CR		N001	G00	S1 T1 M1	CR
N002	Xs Zs		CR		N002	Xs Zs		CR
N003	G86		CR		N003	G88 NLA01 H D		CR
N004	D E		CR		NLA01	G81		CR
N005	G00 Xa Za		CR	→	N005	G00 Xa Za		CR
N006	G34 Zb Zb K F		CR		N006	G34 Xb Zb E F		CR
N007	Xc Zc		CR		N007	Xc Zc		CR
N008	CR Xd Zd		CR		N008	G00 Xd Zd		CR
N009	G80		CR		N009	G80		CR

Рис.211. Пример программы

<u>OSP3000L</u>					<u>OSP5000L</u>			
N001	G00		CR		N001	G00	S T M	CR
N002	Xs Zs		CR		N002	Xs Zs		CR
N003	G86		CR		N003	G71 Xb Zb I D H E1 F		CR
N004	D E0		CR	→	N007	Xt Zt		CR
N005	G33 Xb Zb (I K) F		CR					
N006	G80		CR					
N007	Xt Zt		CR					

Рис.212. Пример программы

<u>OSP3000L</u>					<u>OSP5000L</u>			
N001	G00	S T M	CR		N001	G00	S T M	CR
N002	Xs Zs		CR		N002	Xs Zs		CR
N003	G86		CR		N003	G72 Xb Zb K D H E1 F		CR
N004	G82		CR	→	N008	Xt Zt		CR
N005	D E0		CR					
N006	G33 Xb Zb (I K) F		CR					
N007	G80		CR					
N008	Xt Zt		CR					

Рис.213. Пример программы

4) Для системы OSP 5000L обработка канавок и сверления осуществляются не в режиме автоматического программирования IAP, а при комплексном постоянном цикле.

Если сверление и обработка канавок выполняются в комплексном постоянном цикле, то код G преобразуется в G73 или G74 в зависимости от направления резания, продольной или поперечной обработки.

Example of Program:

1	<u>OSP3000L</u>		<u>OSP5000L</u>
N001	G00 Xt Zt S1 T1 M1	CR	N001 G00 Xt Zt S1 T1 M1 CR
N002	Xs Ts	CR	N002 Xs Zs CR
N003	G83	CR	N003 G73 Xa Za K D Fr CR
N004	D E	CR	N007 G00 Xt Zt CR
N005	G01 Za Za Fr	CR	
N006	G80	CR	
N007	G00 Zt Zt	CR	

Example of Program:

1	<u>OSP3000L</u>		<u>OSP5000L</u>
N001	G00 Xt Zt S1 T1 M1	CR	N001 G00 Xt Zt S1 T1 M1 CR
N002	Xs Zs	CR	N002 Xs Zs CR
N003	G83	CR	N003 G74 Xa Za I D Fr CR
N004	G82	CR	N008 G00 Xt Zt CR
N005	D E	CR	
N006	G01 Xa Za Fr	CR	
N007	G80	CR	
N008	G00 Xt Zt	CR	

Note: When stock "L", or "U" or "W" is "0", finish cut cycle (G87) is not carried out.

Рис.214:

1 - пример программы, 2 - примечание, Если указанный припуск равен 0, то цикл чистовой обработки (G87) не выполняется

19.4. Ограничения по преобразованию функций на ленте при переходе от системы OSP 3000 к системе OSP 5000

Система OSP 5000 обладает функцией преобразования ленты, которая обеспечивает преобразование управляющей программы для системы OSP 3000 с сохранением совместимости с системой OSP 5000. Однако эта функция не позволяет осуществить преобразование всех управляющих программ для системы OSP 3000, так как некоторые второстепенные (не разъясняемые в инструкции по программиро-

ванию) или особые характеристики не подлежат преобразованию.

Как правило, элементы программы, не уточняемые в данном разделе, являются выходными характеристиками для первоначального варианта OSP 3000 даже после преобразования ленты. Таким образом, характеристики, которые не перечислены ниже, имеют возможность преобразования по сравнению с указанными в соответствии с изменениями системы управления от OSP 3000 до OSP 5000 и невозможны для исполнения в системе OSP 5000.

Таблица 34

Характеристики непреобразуемых функций

Specifications .1	OSP3000	OSP5000
Programming Format 2 Command Value "0" 3	7 Value "0" is omissible and just "X" or "Z" is acceptable.	8 Value "0" is not omissible and program must be made as "X0" and "Z0".
+/- Sign 4	"+" and "-" signs may be placed at any position before the next address character. See example below. 9 X100- Z1+00	The signs must be placed right before the numeral value as below: 10 X-100 Z+100
Space 5	Spaces may be placed at any position. See example below: 11 X10 0 Z 1 00	Spaces must not be placed between numerals or between numerals and characters: 12 X10 0Z 100 → (not permissible) 13
M02	Leading zero is omissible and "M2" is acceptable. 14	Leading zero is not omissible and the code must be given as "M02". 15
/ (Slash) 6	The slash code may be placed at any position in a block. The commands after the slash code are ignored. 16 N001 G00 X100 /Z100	The slash code must be placed either at the beginning of each block or right after a sequence number. Otherwise, it is interpreted as a division calculation. 17 N001 /G00 X100 Z100

Таблица 34 (продолжение I)

Inconvertible Specifications (continued):

Specifications	OSP3000	OSP5000
% Code 18	Commands may be programmed in the first sequence following the % code. 19 % N001 G00.....	The % code must be followed by an end of block command, LF or CR. 20 % N001 G00..... CR LF
Thread Cutting 21 8 times F command 22	23 Combined with M27, a thread lead is programmed following address character F in 8 times the actual lead. F000 M27	25 Codes M27 and M28 cannot be programmed since numerical values smaller than 0.001 mm are programmable.
10 times F command 22	24 Combined with M28, a thread lead is programmed following address character F in 10 times the actual lead. F000 M28	F0.00000
Designation of Reference Axis of Thread Lead 25	26 For a thread having an angle of 45 deg. or larger to Z-axis, its lead is referenced to X-axis.	Code M27 is used to change reference axis of thread lead from Z-axis to X-axis. 27
Simultaneous 4-Axis Cutting 28 G13 and G14	29 Axis movement commands and other commands can be programmed in the same block containing G13 or G14 selecting the turret used for cutting.	30 Both G13 and G14 must be specified in a block independently.
M00 and M01	31 These M codes may be programmed only for one turret.	32 These M codes must be programmed for both A and B turrets.
Mirror Image 33 G13 and G14	34 Axis movement commands and other commands can be programmed in the block containing G13 or G14.	35 Both G13 and G14 must be specified in a block independently.

Таблица 34 (продолжение 2)

<p>Nose Radius Compensation 36</p> <p>Start-Up 37</p> <p>(NO G00 X0 Z0 N1 G42 X1 Z1 N2 G01 X2 Z2 F1</p>	<p>38</p> <p>With the commands in block N1, the cutting tool is positioned so that the tool nose comes to contact both straight lines POP1 and P1P2.</p>	<p>37</p> <p>With the commands in block N1, the cutting tool is positioned so that the tool nose comes to contact the straight line P1P2 at point P1:</p>
<p>LAP 39</p> <p>Sequence Branch 40</p>	<p>41</p> <p>Sequence to which jump or branch is made after the completion of rough or finish cutting cycle is specified following address character Q.</p>	<p>42</p> <p>Since commands of contour or shape definition, rough cutting cycle, and finish cutting cycle are independently programmed, no sequence branch function is available.</p>

I - характеристика, 2 - формат программы, 3 - командное значение, 4 - знак, 5 - промежуток, 6 - косая дробь, 7-указанную величину можно пропустить при допустимых указанных значениях, I8 - нулевое значение нельзя пропустить, поэтому программа составляется для указанных значений, 9 - знаки могут размещаться в любом положении перед следующим знаком адресования. См. приведенный пример, I0 - знаки должны указываться непосредственно перед цифровым значением, указанным ниже, II - знаки размещаются в любом месте. См. приведенный пример, I2 - предусмотреть промежутки, которые не должны размещаться между цифрами или между цифрами и знаками, I3 - не допускается, I4 - предшествующие нули опускаются, а указанная команда является допустимой, I5 - предшествующие нули не опускаются, а код указывается в следующем виде, I6 - код в виде косой дроби может размещаться в любом месте кадра. Команды после этого кода не учитываются, I7 - код в виде косой дроби указывается либо в начале каждого кадра, или сразу после номера последовательности. В противном случае он рассматривается как разделитель при расчетах, I8 - код, I9 - команды могут программироваться в первой последовательности после указанного кода, 20 - после кода следует указание команды окончания кадра, 2I - нарезание резьбы, 22 - команда повторяется

указанное число раз, 23 - в сочетании с кодом M27 шаг резьбы программируется после кода адресования F с увеличением фактического шага в 8 раз, 24 - в сочетании с кодом M28 шаг резьбы программируется после знака адресования F с превышением фактического шага в 10 раз, 25 - невозможность программирования указанных кодов, так как числовые значения менее, чем 0,001 мм подлежат программированию, 25 - обозначение контрольной оси для шага резьбы, 26 - для резьб с углом 45° или более по указанной оси контрольное обозначение шага производится по оси X, 27 - код M27 используется для изменения контрольной оси или шага резьбы, 28 - одновременная обработка по четырем осям, 29 - команды на перемещение вдоль оси и прочие команды могут программироваться в едином кадре, содержащем G13 или G14 с выбором револьверной головки для резания, 30 - оба кода нужно независимо указывать в кадре, 31 - коды M могут программироваться только для одной револьверной головки, 32 - коды M должны программироваться для револьверных головок A и B, 33 - зеркальное изображение, 34 - команды на перемещение вдоль оси и другие команды могут программироваться в кадре с указанными кодами, 35 - оба указанные кода должны независимо указываться в кадре, 36 - компенсация радиуса закругления вершины резца, 37 - начало, 38 - если команды находятся в указанном кадре, то позиционирование режущего инструмента осуществляется таким образом, чтобы вершина резца подводилась к контакту с прямой линией в указанной точке, 39 - автоматическое программирование, 40 - последовательность перехода, 41 - последовательность перехода после завершения цикла чернового или чистового резания указывается после знака адресования O, 42 - в связи с независимым программированием команд на профилирование или формирование контура и в связи с независимым выбором цикла чернового и чистового резания функция ответвления не вводится

Непресобразуемые особые характеристики системы O2P 3000

- Программа параметра (задача пользователя)
- Контроль ресурса стойкости режущего инструмента
- Функции измерений (измерение заготовки, измерение инструмента, измерение после обработки)
- Робот
- Программированная функция задней бабки

- Прутковый питатель
- Множественная модель обработки (связи по оси С), команды по оси С, постоянные циклы
- График работы
- Определение перегрузки по оси Z

Невозможно преобразование других функций, предназначенных лишь для характерной модели и для станка, изготовленного по специальному заказу.

Раздел 20. Функция выдвижения и отвода шпинделя задней бабки (с подтверждением от конечного выключателя)

20.1. Коды M

M55 - отвод шпинделя задней бабки

M56 - выдвижение шпинделя задней бабки

Выбор кодов M55 или M56 не зависит от включения питания во время работы или во время перенастройки системы управления.

На станках с двумя револьверными головками программирование кодов M55 или M56 может осуществляться независимо от выбора револьверной головки с помощью кодов G13 или G14.

20.2. Тревожная сигнализация

Сообщение	Наименование
Ответ на выдвижение шпинделя задней бабки (тревожное сообщение А)	Отсутствие ответа на подтверждающий сигнал окончания выдвижения через 5с после команды M56 на выдвижение шпинделя задней бабки
Избыточное выдвижения шпинделя задней бабки (тревожное сообщение А или В)	Избыточное выдвижение шпинделя задней бабки превышает предельное положение выдвижения Тревожное сообщение А во время осуществления программы
Неправильное состояние шпинделя задней бабки (тревожное сообщение А)	Программируется команда на включение вращения шпинделя, когда шпиндель задней бабки не выдвинут для обработки в центрах или не отведен для обработки в патроне Сдвиг положения шпинделя задней бабки из выдвинутого положения при обработке в патроне или из отведенного крайнего положения при обра-

Программируется команда выдвижения и отвода (M55/M56) во время настройки на работу в патроне. Команды M55 или M56 программируются во время работы шпинделя (за исключением случая, когда станок не рассчитан на работу в центрах или при введении отключающей блокировки M157).

Раздел 2I. Программа графика

2I.1. Описание

Программа графика используется для непрерывной обработки деталей различных типов в автоматическом режиме, если загрузка и выгрузка этих деталей может автоматически выполняться роботом, автоматическим загрузчиком или другим транспортно-грузоподъемным устройством. Сочетание программы графика с программами механической обработки обеспечивает непрерывную обработку в автоматическом режиме.

В программе графика предусмотрены следующие сообщения:

1) Сообщение для выбора главной программы PSELECT

2) Сообщение для указания последовательности выполнения кадров программы графика. IF, GOTO

3) Сообщение для ввода переменных VSET

Программа графика может составляться с использованием перечисленных выше сообщений, так что различные типы заготовок могут обрабатываться в автоматическом непрерывном режиме за счет осуществления программы графика.

Операции обработки при выполнении программы графика перечислены в п.4.1.6. в Инструкции по эксплуатации.

2I.2. Программирование

1) Сообщение для выбора исполнения главной программы PSELECT

С помощью сообщения PSELECT программируется наименование главной программы, наименование выполняемой подпрограммы с указанием числа повторений вызываемой программы.

Как показано на рис.2I5, файл SHAFT.MIM содержит программу механической обработки OSHTI, а файл SHAFT.SUB предусмотрен для подпрограммы, которая вызывается с помощью указанной программы.

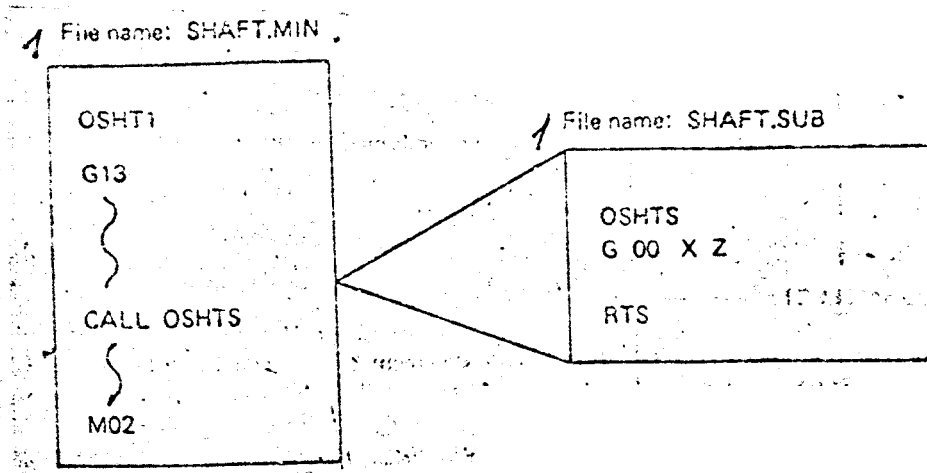
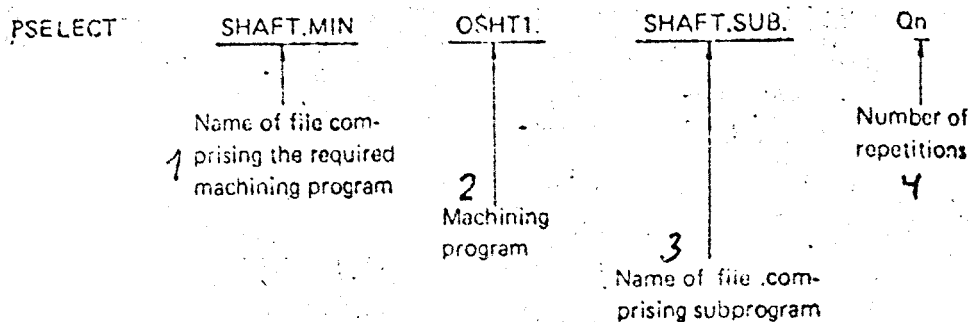


Рис. 215:

I - наименование файла

Выполнить приведенное ниже программирование для повторения программы механической обработки "п" раз.



I - наименование файла, содержащего требуемую программу обработки, 2 - программа механической обработки, 3 - наименование файла, содержащего подпрограмму, 4 - число повторений

Примечание. Если программа должна выполняться один раз, то нет необходимости программировать команду Q.

Команда 00000 обеспечивает бесконечное повторение программы.

2) Сообщение для определения последовательности выполнения кадра программы графика (IF, GOTO).

Если сообщение IF обеспечивает переход к указанному номеру или наименованию последовательности, то выполняются заданные условия. (См. рис.216).

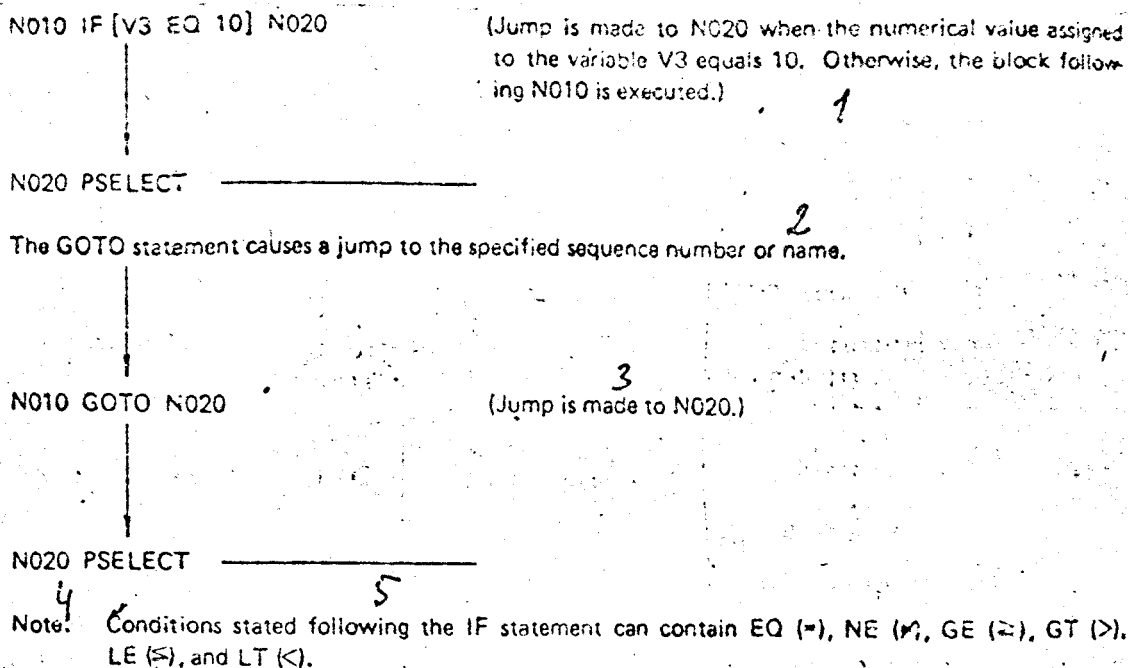


Рис. 216:

I - осуществляется переход к № 020, если числовое значение, относящееся к переменному U3, равняется 10. В противном случае выполняется кадр после N010, 2 - указанное сообщение обеспечивает переход к заданному номеру или наименованию последовательности, 3 - осуществляется переход, 4 - примечание, 5 - в условиях после сообщения IF содержится следующее:

3) Сообщение для настройки переменных VSET

Это сообщение предназначено для общих переменных U1 - U32.

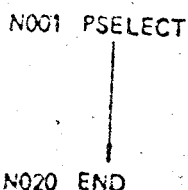
U7 = I назначение числового значения I для переменного U7

U1 = U7+I назначение числового значения U7+I для переменного U1

U6 = U10+U11 назначение числовой величины U10+U11 для переменного U6

4) Конец программы графика END

Программировать сообщение END в конце программы графика (см. рис. 217).



21.3. Пример программы

```
N001 VSET V1 = 0
N002 PSELECT SHAFT.MIN, OSHT1, SHAFT.SUB, Q100
N003 VSET V1 = V1+1
N004 IF [V1 GT 1] N007
N005 PSELECT SHAFT.MIN, OSHT2, SHAFT.SUB, Q50
N006 GOTO N008
N007 PSELECT SHAFT.MIN, OSHT3, SHAFT.SUB, Q200
N008 IF [V1 GT 2] N010
N009 GOTO N002
N010 END
```

Explanation of example program:

- 1 N001: Variable V1 is assigned with numerical value zero ($V1 = 0$).
- 2 N002: Machining program OSHT1 is executed 100 times.
- 3 N003: Variable V1 is set to "V1+1" ($V1 = V1+1$).
- 4 N004: If V1 is greater than 1 ($V1 > 1$), the sequence jumps to N007.
- 5 N005: Machining program OSHT2 is executed 50 times.
- 6 N006: Sequence jumps to N008.
- 7 N007: Machining program OSHT3 is executed 200 times.
- 8 N008: If V1 is greater than 2 ($V1 > 2$), the sequence jumps to N010.
- 9 N009: Sequence jumps to N002.
- 10 N010: The END statement indicates the end of the schedule program.

Рис.218 :

1 - разъяснения, относящиеся к примеру программы, 2 - переменное $V1$ имеет нулевое цифровое значение, 3 - указанная про-

грамма механической обработки выполняется 100 раз, 4 - введение переменного параметра, 5 - в случае превышения переменным параметром значения I происходит переход к № 007, 6 - программа механической обработки повторяется 50 раз, 7 - переход последовательности, 8 - программа механической обработки повторяется 200 раз, 9 - если UI превышает 2, то последовательность переходит к №10, 10 - сообщение EWD обозначает окончание программы графика

Раздел 22. Проверка задевания режущего инструмента

22.1. Описание

Если программируется одновременная обработка по четырем осям, то проверку вероятности задевания режущего инструмента на револьверных головках А и В осуществляют в зонах задевания предварительно с тем, чтобы исключить неприятности, связанные с задеванием инструмента во время осуществления программы.

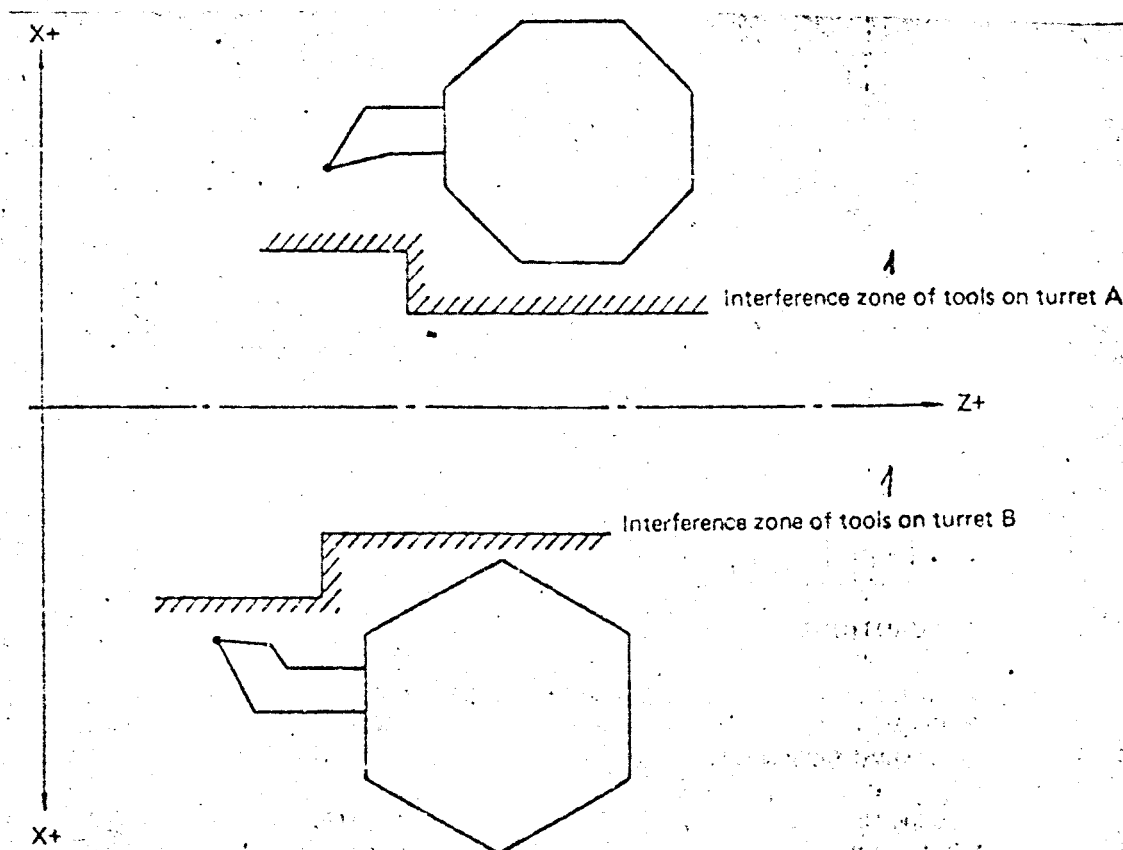


Рис. 219:

I - зона задевания инструментов указанной головки

Как показано на рис.219, в случае перекрывания зон задевания обеих револьверных головок во время исполнения программы включается тревожная сигнализация.

Подробные сведения, относящиеся к операциям настройки зоны задевания, указаны в операциях в разделе "Настройка данных для инструмента" в п.22.4.

22.2. Программирование

После того, как установлены зоны задевания, можно включить или выключить функцию проверки задевания с помощью программирования команд M25 или M24 соответственно.

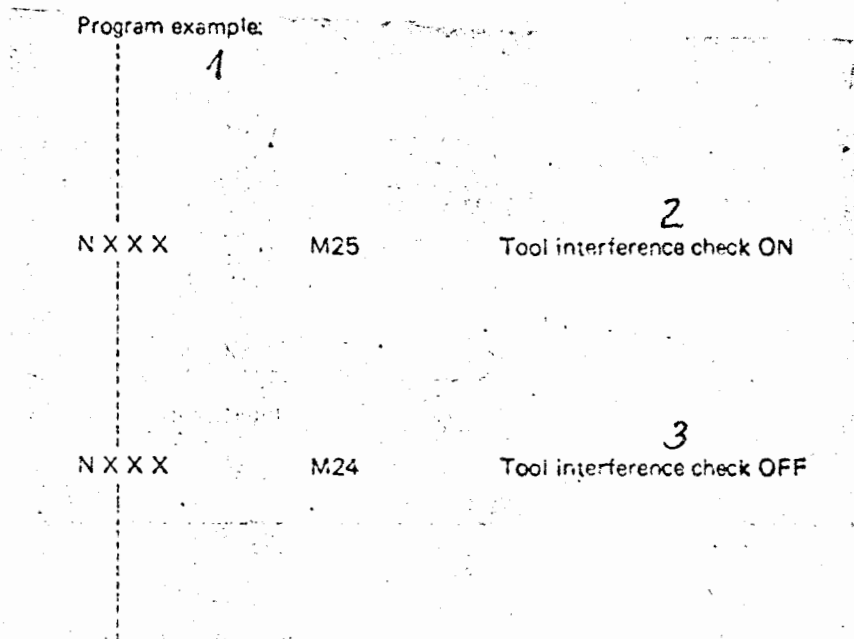


Рис.220:

1 - пример программы, 2 - включение проверки задевания режущего инструмента, 3 - выключение проверки задевания режущего инструмента

22.3. Меры предосторожности

1) При включении питания или при повторной настройке системы управления, она автоматически переключается на режим M24 "Выключение проверки задевания режущего инструмента". Для включения функции проверки задевания режущего инструмента нуж-

но повторно программировать M25.

2) При загрузке ленты № 3 с управляющей программой происходит стирание данных в зоне задевания. Убедиться в обязательном вводе данных для зоны задевания инструмента при загрузке.

22.4. Работа в режиме ввода данных для инструмента.

В режиме ввода данных для инструмента нужно установить данные для компенсирующего сдвига инструмента и данные для зоны задевания инструмента. Операции ввода данных для сдвига инструмента приведены в п. 3.2.2.3 "Инструкции по эксплуатации".

Операции ввода данных для зоны задевания инструмента (см. рис.22I).

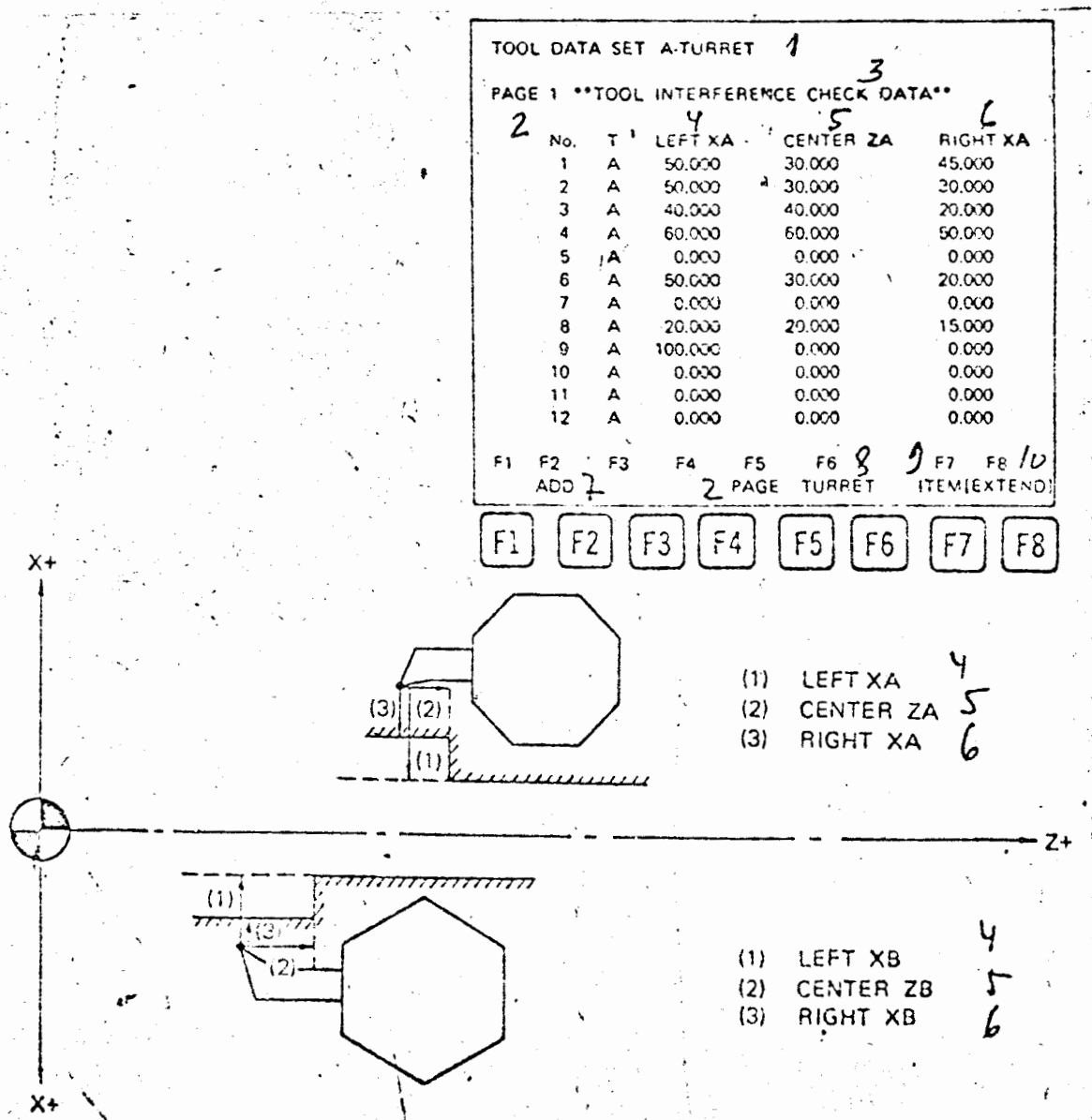


Рис. 22I:






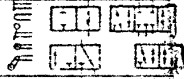




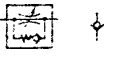





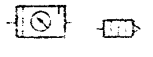

1 - ввод данных режущего инструмента для револьверной головки А,
2 - страница, 3 - данные проверки зацева режущего инструмен-
та, 4 - левая сторона, 5 - середина, 6 - правая сторона, 7 -
суммирование, 8 - револьверная головка, 9 - пункт, 10 - рас-
ширение

1) Нажать клавишу "Данные для инструмента в секции "Режим"
на панели управления системой OIP 50001.

2) Нажать функциональную клавишу F7 ITEM для выбора стра-
ницы с данными настройки режущего инструмента (см. рис.221).

3) В соответствии с размерами, приведенными на рис.221,
установить требуемые размерные параметры для каждого из ин-
струментов I-I2 на отдельных револьверных головках.

Inhaltsverzeichnis Pneumatik

	Bau- gruppe	Symbol
1 Allgemeine Informationen		
2 Technische Informationen		
3 Normzylinder		
4 Sonderzylinder		
5 Drehantriebe		
6 Pneumatische Steuerungssysteme		
7 Manuell und mechanisch betät. Wegeventile und Wahlschalter	P 400, P 401 P 402, P 403	
8 2/2-, 3/2-, 5/2- und 5/3-Wegeventile, pneumatisch betätigt	P 404, P 405 P 406, P 407	
9 2/2-Wegeventile, elektromagnetisch betätigt	P 100 P 410	
10 3/2-Wegeventile, elektromagnetisch betätigt	P 101 P 411	
11 4/2-, 5/2-, 5/3- und 5/4-Wegeventile, elektromagnetisch betätigt	P 413 P 415 P 416	
12 Strom- und Sperrventile	P 417 P 418	
13 Anschlußplatten, Batterieleisten, Drosselplatten	P 430 P 431	
14 Sicherheitssteuerungen	P 440	
15		
16 Proportionalventile, Regelverstärker, Meßwertaufnehmer	P 409 P 633	
17 Elektrische Steuerungen, Befätigungsmagnete	P 60X, P 65X P 66X, P 690	
18 Pressentechnik		
19 Druck- und Temperaturschalter	P 5XX	
20 Spezialgeräte		
21 Zubehör		
22 Service	P 75X	
23 Anwendung		

