

11

11. АРИФМЕТИКА ПАРАМЕТРОВ

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

11.1 Общие принципы

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

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

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

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

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

Параметр **R95** определен для функции управления.

11.2 Управление двойной точностью

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

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

1. Двойная точность запрещена:

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

2. Двойная точность разрешена:

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

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

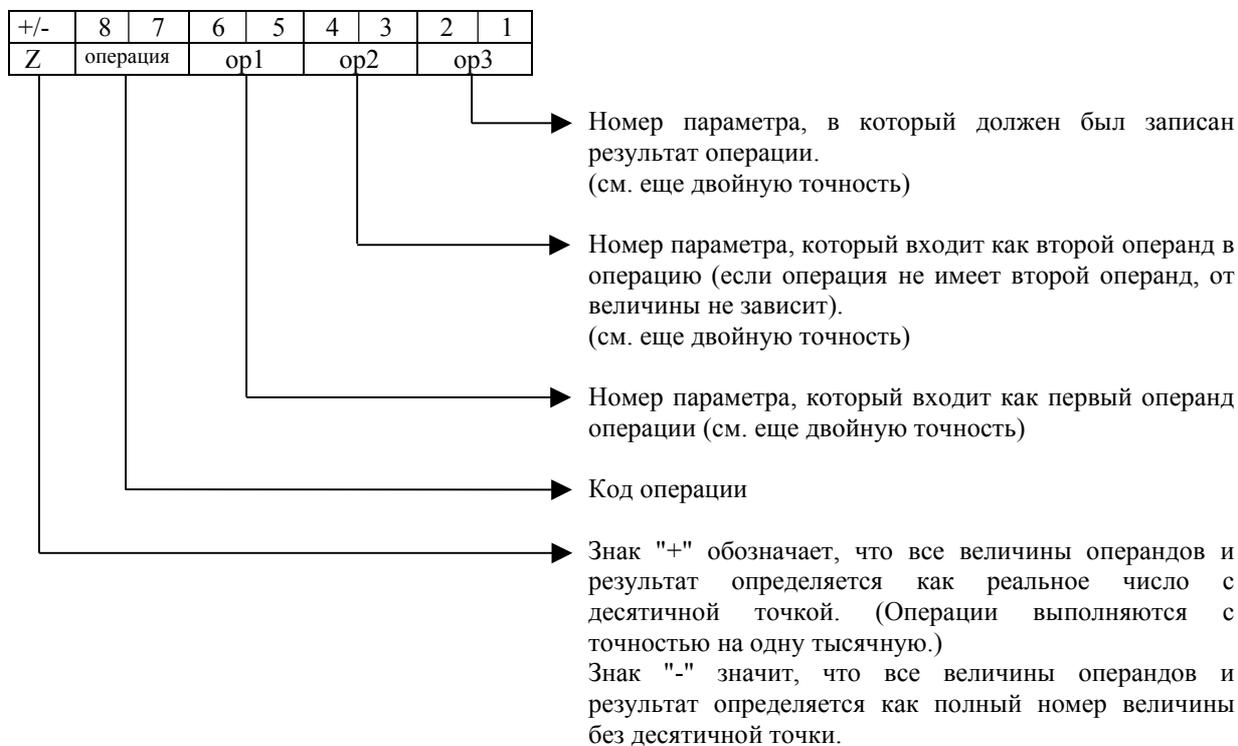
4 декада машинной константы 283: 0 арифметика параметров использует параметры 0-49 с

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

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

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

11.3 Структура параметров управления



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

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

		параметре согласно сi1* величина 73 , в противоположном случае 78 .
Меньше $op1^* < op2^* \rightarrow c\dot{i}l^*$	17	Если величина параметра согласно op1 меньше чем величина согласно op2 , будет в параметр согласно сi1* уложена величина 73 , в противоположном случае 78 .
Меньше или равно $op1^* \lceil op2^* \rightarrow c\dot{i}l^*$	18	Если величина параметра согласно op1 меньше или равна величине согласно op2 , будет в параметр согласно сi1* записана величина 73 , в противоположном случае 78 .
Сравнение $op1^* = op2^* \rightarrow c\dot{i}l^*$	19	Если величина параметра согласно op1 равна величине согласно op2 , будет в параметр согласно сi1* записана величина 73 , в противоположном случае 78 .
Условие "равно" $op1^* = op2^* \rightarrow$ продолжай	20	Если величина параметра согласно op1 равна величине согласно op2 , продолжается расчет следующей параметрической операции.
Условие "неравно" $op1^* \neq op2^* \rightarrow$ продолжай	21	Если величина параметра согласно op1 не равна величине согласно op2 , продолжается расчет следующей параметрической операции.
Конец условия \rightarrow продолжай	22	Окончание условий для расчета параметрических операций.
Перевод int \rightarrow real $real(op1) \rightarrow c\dot{i}l$	23	Перевод величин целых чисел согласно параметру op1 на реальную величину согласно сi1
Перевод real \rightarrow int $int(op1) \rightarrow c\dot{i}l$	24	Перевод реальной величины согласно op1 на целые числа величин согласно сi1
Перемещение измерение ODM $\rightarrow c\dot{i}l^*$	25	Перемещение измерением координат согласно op1* (и op2*) и запись в параметр согласно сi1* . В операнды op1 (op2) вводится порядковый номер координат (1,2,...6).
Перемещение программированной величины координат PROG $\rightarrow c\dot{i}l^*$	26	Перемещение программированной координаты согласно величине координат op1* (и op2*) и запись в параметр согласно сi1* . В операнд op1 (op2) вводится требуемый номер координат (1,2,...6).
Функция tan(x) $\tan(op1) \rightarrow c\dot{i}l^*$	27	Функция tan(x) согласно op1 и запись результата в сi1 . Операнд op1 не может иметь двойную точность и должен принимать реальное значение. Результат является реальным номером и может иметь двойную точность.
Сумма геометрической последовательности $1 + k + k^2 + \dots + k^{n-1}$ $(k^n + 1) / (k - 1) \rightarrow c\dot{i}l^*$ k...op1* n...op2*	28	Сумма геометрической последовательности $1+k+k^2+\dots+k^{n-1} = (k^n-1)/(k-1)$ Частное k определено параметром op1 . Количество членов n определено параметром op2 . Операнды op1 и op2 не могут иметь двойную точность и должны быть реальными числами. Результат принимает реальное значение и может иметь двойную точность.
Условие "больше" $op1^* > op2^* \rightarrow$ продолжай	29	Если величина параметра согласно op1 больше, чем величина согласно op2 , продолжается расчет последующей параметрической операции.
Условие "больше или равно" $op1^* \lceil op2^* \rightarrow$ продолжай	30	Если величина параметра согласно op1 больше или равна величине согласно op2 , продолжается расчет последующей параметрической операции.
Условие "меньше" $op1^* < op2^* \rightarrow$ продолжай	31	Если величина параметра согласно op1 меньше величины согласно op2 , продолжается расчет последующей параметрической операции.
Условие "меньше или равно" $op1^* \lceil op2^* \rightarrow$ продолжай	32	Если величина параметра согласно op1 меньше или равна величине согласно op2 , продолжается расчет последующей параметрической операции.
ПРОГРАММА запуск программы от op1 по c\dot{i}l	33	Запуск программы от параметра op1 до параметра сi1

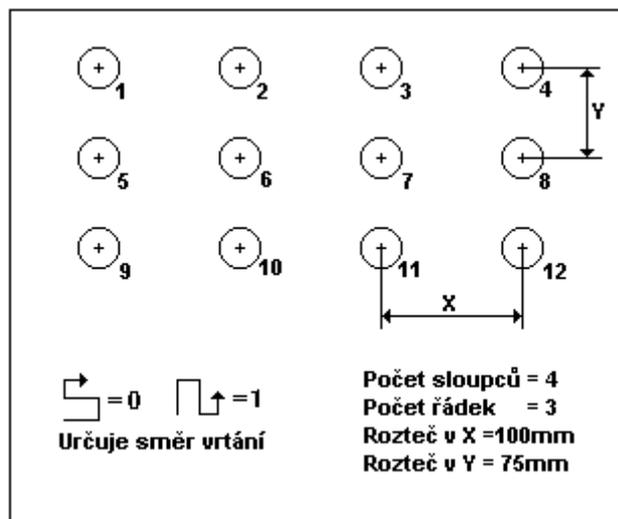
		(включительно). Параметры могут быть заполнены в одинаковом блоке. За функцией 33 не должна быть запрограммирована следующая параметрическая операция.
Перемещение параметров перемещение от op1 до op2 до cil	34	Перемещение параметров от op1 (включительно) до op2 (включительно) до области согласно параметру cil Для op1=99 аннулируется область от cil по op2
Смена параметров смена из op1 по op2 в cil	35	Смена параметров из op1 (включительно) по op2 (включительно) в область согласно параметру cil
Вписанная дуга	36	Функция использует большую область параметров
sin(op1) -> cil	37	Функция sin(x) согласно op1 и запись результата умноженного на *1000000 в cil .
cos(op1) -> cil	38	Функция cos(x) согласно op1 и запись результата умноженного на *1000000 в cil .
arcsin(op1/op2)	39	Функция arccos(x/y) согласно op1/op2 и запись результата в cil .
R*cos((I*B/N)+A)	40	Функция для расчета дыр на окружности
op1+1 -> cil	41	для op2=0 инкрементация величины op1 и запись в cil для op2<>0 больше величины согласно op1 на номер в op2 и запишется в cil
op1-1 -> cil	42	для op2=0 декрементация величины op1 и запись в cil для op2<>0 вычитается величина согласно op1 от номера в op2 и запишется в cil
SAVE	44	Сохранение всех параметров
RESTORE	45	Возобновление параметра от op1 по op2 включительно в cil Для op1=op2=cil=0 возобновление всех 96 параметров
LICHÝ(op1) -> cil (НЕЧЕТНОЕ)	46	Если величина параметра согласно op1 нечетная, запишется в параметр согласно cil величина 73 , в противоположном случае 78 .
SUDÝ(op1) -> cil (ЧЕТНОЕ)	47	Если величина параметра согласно op1 четная, запишется в параметр согласно cil величина 73 , в противоположном случае 78 .
LOGIC(dek) -> dek(cil)	48	Логические операции для декады параметров 8. и 7. декада является кодом для логики = 48 6. декада – это код операции: 1 = SET, 2=CLR, 3=NOT, 4 = OR, 5= AND, 6= XOR, 7=TEST 5. декада определяет порядковый номер декады для логической операции 1,2,...,8 4. и 3. декада определяет 1. операнд для операции OR, AND, XOR и TEST 2. и 1. декада определяют 2. операнд и одновременно cil для операций OR, AND, XOR. Операции SET, CLR и NOT работают только с параметром согласно cil . Для операции TEST в параметре согласно cil будет записана величина 73 или 78 .
READ(TAB) -> cil	49	Перемещение из таблиц в параметры 8. и 7. декада – это код для перемещения = 49 6. декада – это код таблицы: 1=таб. коррекций, 2=таб. сдвига, 3=таб. машинных констант, 4=регистр блока BWRK, 5=регистр блока AWRK 5. декада определяет столбец в отдельных таблицах 1,2... для регистров блока это: 1=функция M, 2=функция G 3=функция P, 4= H, 5= S, 6= T, 7= Q, 8=D 4. и 3. декада определяют источник в таблицах. Для таб. коррекций это 1-99, для таб. сдвига это 53-59, для машинных констант это 0-99 + столбец*100, для регистров блока вводится группа функций M и G

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

11.4 Пример использования арифметики параметров

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

Макроцикл для сверления дыр использует эти вступительные параметры:



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

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

Определяет направление сверления

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

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

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

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

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

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

Пометка:

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

R09=0.081, R12=0.004, R13=0.003

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

В следующем тексте сделаем подробный комментарий к арифметическим операциям.

N1 G79 L9001

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

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

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

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

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

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

СВЕРЛЕНИЕ ПО СТОЛБЦАМ (Y)

N105

.....

.....

.....

N900 G70 „Конец сверления по столбцам

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

" шаг на следующую строку

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

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

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

Количество строк находится в параметре в R19.

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

N1170 G26 R5=47190017 G80

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

N1185 GR17 L1200 R15=0

" шаг на L1200 в случае четного количества строк

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

N1190 G28 R5=-41120012

" Из количества повторов будет рассчитано
количество расстояния столбцов

R6=-03121015

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

R7=08150015

" Изменение знака траектории

Подробным способом будет рассчитан возврат в оси Y.

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

N1200 G27 R5=-03131114 R6=08140014

Рассчитанные траектории записаны в R14 и R15.

N1210 YR14 XR15

" возврат в исходную точку

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

**N1230 G90 G26 R05=45000000
N1300 G70**

" Вернутся назад сохраненные параметры
" конец макроцикла

*