SYD.

SIEMENS

SINUMERIK 8M/8MC/ Sprint 8M

Programming instructions

SINUMERIK 8M/8MC/ Sprint 8M

Programming instructions

Edition 9.84

Siemens Aktiengesellschaft

SINUMERIK - Documentation

Key to editions

Up to the present edition, the editions below have been issued.

In the column "alterations" the chapters are listed which have been altered with respect to the preceeding edition.

Edition	Order Number	Alterations
E.3.80	E321/1737-101	First edition
E.9.80	E321/1753-101	Revised edition
E.3.81	E321/1795-101	Revised edition
E.11.81	E321/1883-101	Revised edition including 8M
E.4.82	E321/1883-101	S.1-10, 3-24, 3-34, 3-38 3-56, 4-2, 5-9, 7-15, 7-17, 8-26, 8-27, 8-28
E.4.83	E321/2028-101	Revised edition
E.9.83	E321/2106-101	Revised edition
E.9.84	E80210-T6-X-A8-7600	S. 0-1, 1-7, 1-8, 1-9, 2-2, 3-60,bis 3-66, 4-4, 6-1, Chapter 7-6 canceld

Table of Contents Page

Table of Contents (continued) Page

Table of Contents (continued) Page

Table of Contents (continued)

0.1 System of the programming instructions and general notes on them

The chapters 0 to 6 describe those functions which the controls 8M, 8MC, and Sprint 8M have in common. Deviations and additional functions are there with Sprint 8M.
These characteristics are summarized in chapter 7/8.
Special applications and notes see chapter 8.

The examples in chapter 0-6 have been programmed for 8M/8MC with notes concerning the deviations with Sprint 8M.

The following assumptions are made in the programs used for these programming instructions.

1. The user datum decimal point input is set.

2. The decimal point is written even when it is automatically generated.

3. Block construction is in accordance to DIN 66024, DIN 66217, ISO R 1056, ISO R 1057, and ISO R 1058.

4. The programming examples are written in ISO code.

5. All geometric values are metric.
   For conversion into inch see chapter 8.

6. The maximum values given are limit values for the control. They can be limited in practice by the machine, interface and input/output devices.

7. These programming instructions are designed for the maximum functional range of the control. Functions to be realized by options may be gathered from the catalogues and technical description.

8. For better understanding preparatory functions are even programmed, if these are commands with reset position.

9. The contents of these programming instructions can be found in the fold-out program key.

10. Functions not included in this manual may be available in the control. However, this does not guarantee that these functions will be available with new equipment or in the case of service. We reserve the right to amend these instructions for technical reasons without prior notice.

11. Functions marked with "*" are not included in the basic model of the control.

1. Program Format

1.1 Perforated Tape Coding

The data on the perforated tape is coded according to strictly defined guidelines. Each hole combination defines a unique character. Two perforated tape codes are permissible.

(DIN66025, ASCII)
EIA RS 244B

The control automatically recognizes the perforated tape format. The coding format is determined on reading the first resp. EOR or LF resp. EOB (setting data)
Individual perforated tapes must be coded in one of the allowable codes. It is not permissible to change codes within the same tape nor is it permissible to splice tapes together using different codes. Failure to observe the aforementioned will cause the control to signal a character parity alarm.

The characters in each code are defined to have even or odd parity:

EIA RS 358B even number of holes
EIA RS 244B odd number of holes

The even/odd criterion is used as a simple program check following the first character read. The block parity monitors for an even number of characters within a block of data. A block with an odd number of characters is made even by writing the characters "HT", "SP", or "DEL". Block parity checking can be selected.

As an additional tape read check, a double tape read is performed by the control. The control reads the program into memory then performs a second read while making a character by

activated with a machine parameter.
If a character mismatch occurs tape read is halted and a read error is displayed on the control operator panel. The word address tape format is defined by DIN66025 (ISO R1056) and is in general agreement with EIA RS 274C.

1.2 Address Characters

All characters are read by the control. However, an executable block is assembled using only legal address characters.

EIA RS 358B (ISO code)

Address Words	A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z
Digits	0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
Reserved Characters	%, (,), +, -, /, :, ., @
Non Printable Characters	HT Tabulator SP Space DEL Delete CR Carriage Return LF Line Feed

INPUT READ Ignored Characters The following characters are neither processed nor stored	OUTPUT TO PRINTER/PUNCH The following characters are generated.
HT	
SP (Except within a comment)	SP (following every word but not within a comment)
DEL	
CR (CR LF sequence is arbitrary)	LFCRCR (CRLF selectable in machine data)

LF is displayed as an *

EIA RS 244B

Address Words	a, b, c, d, e, f, g, h, i ,j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z
Digits	1, 2, 3, 4, 5, 6, 7, 8, 9, 0
Reserved Characters	EOR, (,), +, -, /, ., @
Non Printable Characters	Tab Space Delete CR Carriage Return EOB End of Block

INPUT READ Ignored Characters The following characters are neither processed nor stored	OUTPUT TO PRINTER/PUNCH The following characters are generated.
Tab	
Space (Except within a comment)	Space (following every word but not within a comment)
Delete	
EOB	CR (is generated and output twice following EOB)

NOTE: (,), @ are GN defined character codes, see appendix 8.2.7
CR/EOB is displayed as an *

1.3 Word Address System

(Word address system with Sprint 8M see 7.1)

Explanations:

1st address character	address	
2nd address character	L	absolute, incremental
2nd address character	D	incremental
Sign	+ -	absolute dimension value, signed negative(+ not required)
1st digit	0	leading zeros can be omitted: variable word length
2nd digit	decades	adjust digit sequence
2nd and 3rd digit	decades	adjust digit sequence before and after the comma (coordinate values X,Y,Z,I,J,K in mm)
Sign	*	End of block

A word consists of an address followed vy a signed or unsigned digit sequence.

The word address format and thereby the input format is defined by EIA RS 274-C and DIN 66025.

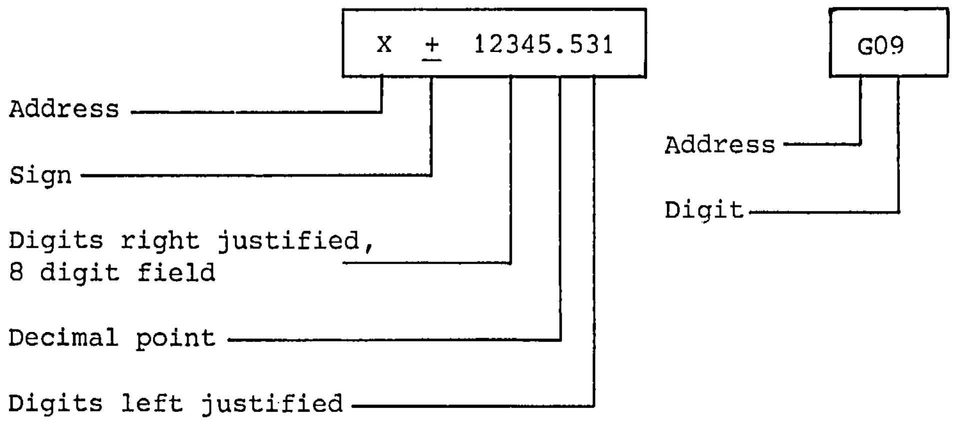
8M: Inch
%04 N04 G02 XL+044 YL+044 ZL+044 UL+044 ID044 JD044 KD044
AL+035 PD044 F05 S04 H06 D03 T04 L5 R2 RL+08 M02 *

Metric
%04 N04 G02 XL+053 YL+053 ZL+053 UL+053 ID053 JD053 KD053
AL+035 PD053 F05 S04 H06 D03 T04 L5 R2 RL+08 M02 *

8MC: Inch
%04 N04 G02 XL+044 YL+044 ZL+044 EL+044 BL+044 CL+044 UL+044
VL+044 WL+044 QL+044 ID044 JD044 L5 PD044 AL+035 F05 S04 T06
H06 D03 R2 RL+08 M02 *

Metric
%04 N04 G02 XL+053 YL+053 ZL+053 EL+053 BL+053 CL+053 UL+053
VL+053 WL+053 QL+053 ID053 JD053 L5 PD053 AL+035 F05 S04 T06
H06 D03 R2 RL+08 M02 *

Example:



Word	Using Decimal Point Programming (User selectable)	Without Decimal Point Programming (User selectable)
0.0001 in. (1 µm)	0.0001 in. (0.001)	1 (1)
0.001 in. (10 µm)	0.001 in. (0.01)	10 (10)
0.01 in. (100 µm)	0.01 in. (0.1)	100 (100)
0.1 in. (1000 µm)	0.1 in. (1. or 1)	1000 (1000)
1.02 in. (10200 µm)	1.02 in. (10.2)	10000 (10200)
10.0 in. (100000 µm)	10.1 in. (100. or 100)	100000 (100000)

Decimal point programming is possible with the following addresses:
X, Y, Z, U, V, W, I, J, K, A, B, C, (D), E, F, R.

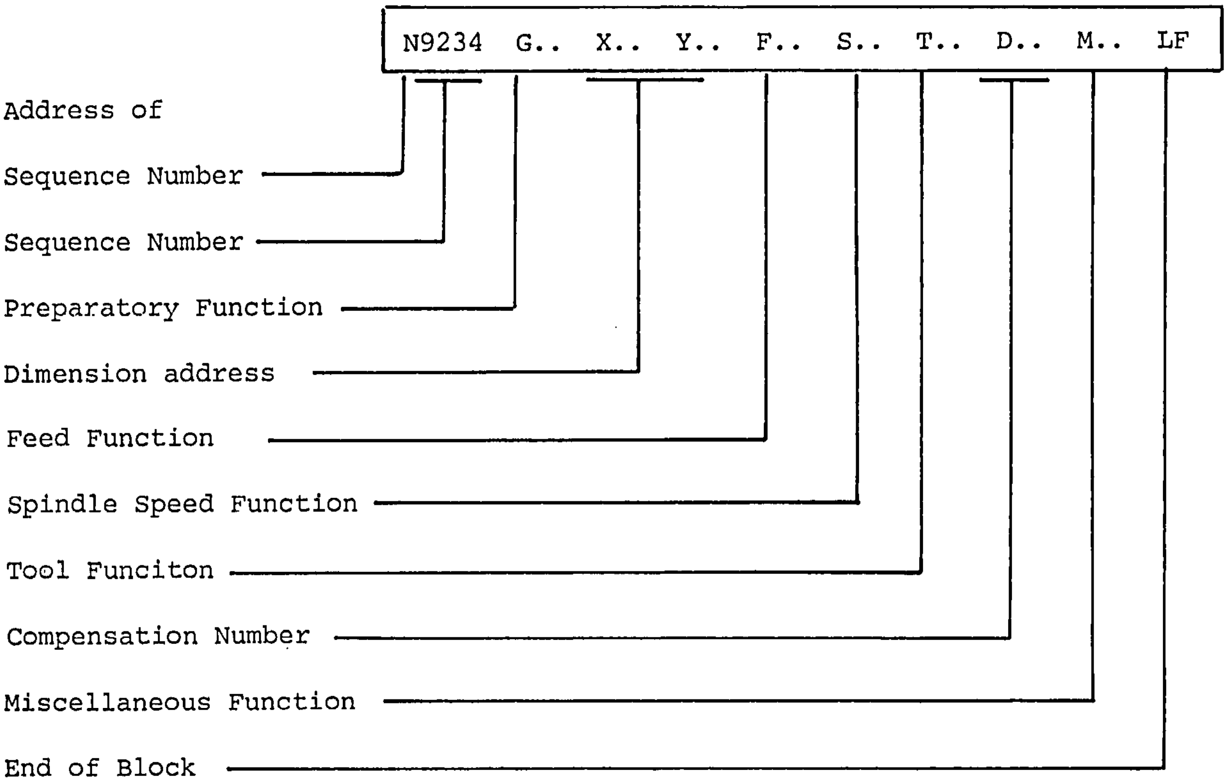
If the input system is modified (decimal point input or input without decim 1 point) it is important to observe that the zero offsets and tool offsets etc. are modified accordingly. There is no automatic updating of these values.

1.4 Variable Block Format

A block consists of several words terminated by the "End of Block" character.

Block length is variable and can have a max. of 120 characters.

An example of a block:



Two types of blocks exist	Main block (:) Subordinate Block (N)
Main block	A main block contains all necessary information required to start the machining sequence at that point.
Subordinate block	Contains all functions that may change from block to block

The block sequence numbers are not necessarily sequentially numbered (N1 to N9999). A numbering sequence can be interrupted arbitrarily, e. g. an edited or inserted block may have a sequence number several orders of magnitude higher than the preceeding sequence number.

At times it is desirable not to execute an entire program, but instead to delete certain program operations. An example would be a part gaging sequence not necessary for every program run. By using the block delete symbol "/" (slash), those program sections not executed every run will be ignored by the control when the skip key (Block delete) has been activated.

The block deleter "/" is placed in front of the block sequence number:

/: Main block deletion

/N Subordinate block deletion

The block preceding the deleted block must agree with the block succeeding the deleted block. If the blocks (preceding and succeeding) do not agree, the program will execute incorrectly when activating and deactivating the block deletion button"skip".

If L999 is programmed in the block preceeding the deleted block, the block deletion button "skip" is active during machining.

Note:

Because of a quick block changeover, several blocks are temporarily stored. When the machine stops because of M00, the following blocks are already stored. The use of "Block delete", however, is only effective with blocks which are not temporarily stored. The temporary storing can be prevented by programming L999 after the M00-block (see also chapter 5.7).

1.5. Leader

The leader is used to differentiate between different tapes. All tape characters are allowed in the leader, with the exception of % if % is used for automatic code recognition or LF if LF is used. The leader is ignored by the control and is not stored.

1.6. Comments

Program blocks can be clarified by using comments. It is possible for the operator to view comments on the display (in the p.p.-picture).

Within a comment all characters except % or LF are legal. A comment between M02/M30 and a further M-function is not allowed. N...M30 (comments) M40 LF is therefore not allowed.
A comment may contain up to a maximum of 29 characters, if more are required then several comments may be programmed consecutively.

Within a comment, there should also be no statement N followed by a number since during block advance the expression in brackets will be read and then N 1234, e.g. would be read as block number.

Example:

```
N20 ...  M00 L999 LF
N25 G26 X 10.25 Y 15.305 (MAX. SAFE ZONE)
(MANUAL CHANGE POSSIBLE)
```

Incorrect	Correct
Y (FLANGE) 100.	X 100. (FLANGE) Z200.
Y 100. (FLANGE) R01	X 100. R01 (FLANGE) Z200.

Address

Dimension

P-parameter

Comment delimeter (

Comment

Comment delimeter)

A comment cannot be placed between a word and its associated parameter or between address and number.

1.7. Part Programm

A part program describes the execution of a work process and contains the part program itself with possible subroutine and/or stored cycle calls.

The program memory may contain a maximum of 199 user programms.

The separation in part programs and subroutines is arbitrary. Stored cycles and user defined cycles provided by the machine tool builder are stored in a protected memory area.

```
%LF
N5 G91 G01 X50. F100. LF
N10        Z100.      LF
N15        X-30.      LF
N20        Z-10.      LF
N25 M30               LF
```

Program start when only program is stored in the program memory.

```
% 1357                LF
```

Start of part program 1357
A maximum of 99 part programs can be simultaneously stored in the program memory. A four digit designator (0-9999) identifies the program. It can be entered or changed subsequently with the help of program edit, as every program is stored under % 0 without a program number.

```
N5 G91 G01 X50. F100. LF
N10        X-30.      LF
N15        Z-10.      LF
N20 M30               LF
```

Preparatory functions, dimensions, feed function direction, etc.

M30 (M02) End of program with reset to program start. In the automatic mode when running directly from tape an M30 prompts a tape rewind.

If the program is enterd from the operator's panel, sequence numbers are automatically generated by the control in intervals of five, after the first block has been entered. By pressing the "CANCEL" key, the control generated sequence number is cleared and a different sequence number can be entered.

1.8. Subroutines

Repetitive patterns and function cycles can be stored as subroutines which can be called arbitrarily by the part program and other subroutines. It is possible to store 199 user programs at the same time in the program memory.

The separation into part programs and subroutines can be made arbitrarily. The numbers L080 to L099 and L900 to L999 can be inhibited.

Subroutine definition:

The definition is designated

- under address L with either 2 or 3 digits and 2 trailing zeros. When entering manually the zeros are automatically generated they must be entered though, from tape.
- optionally either alone without block number or together with other functions in the first block.

The end of the subroutine is defined

- with M17, either alone in an own block or together with other functions - except the L-address - in the last block.

The following definitions are possible:

1. Recommended standard version
```
L12300  N5  G00  X..     LF
N10..
.
.
N..  G00  G90  X..  M17  LF
```

2. Another permissible possibility
```
L12300
N5  G00  X..        LF
N10  ..
.
.
N..  G00  G90  X..  LF
N..  M17            LF
```

3. Smallest subroutine possible
```
L12300  N5  G00  G91  X...  LF
N10  M17  LF
or
L12300  LF
N5  G00  G91  X..  M17  LF
```

4. Path-line machining program with calculation of intermediate points without intermediate stop.
(Machine datum "M17" without 'Auxiliary function output' is set.)
```
L12300  N5  G01  XR..  YR..  LF
N10  M17  LF
```

Presupposition: interface signal "cycle inhibit" set.

Subroutine-call

The subroutine-call is made by a part program with an L-address. Nesting up to three deep (4 levels) is permitted, when the call is made from the part program.

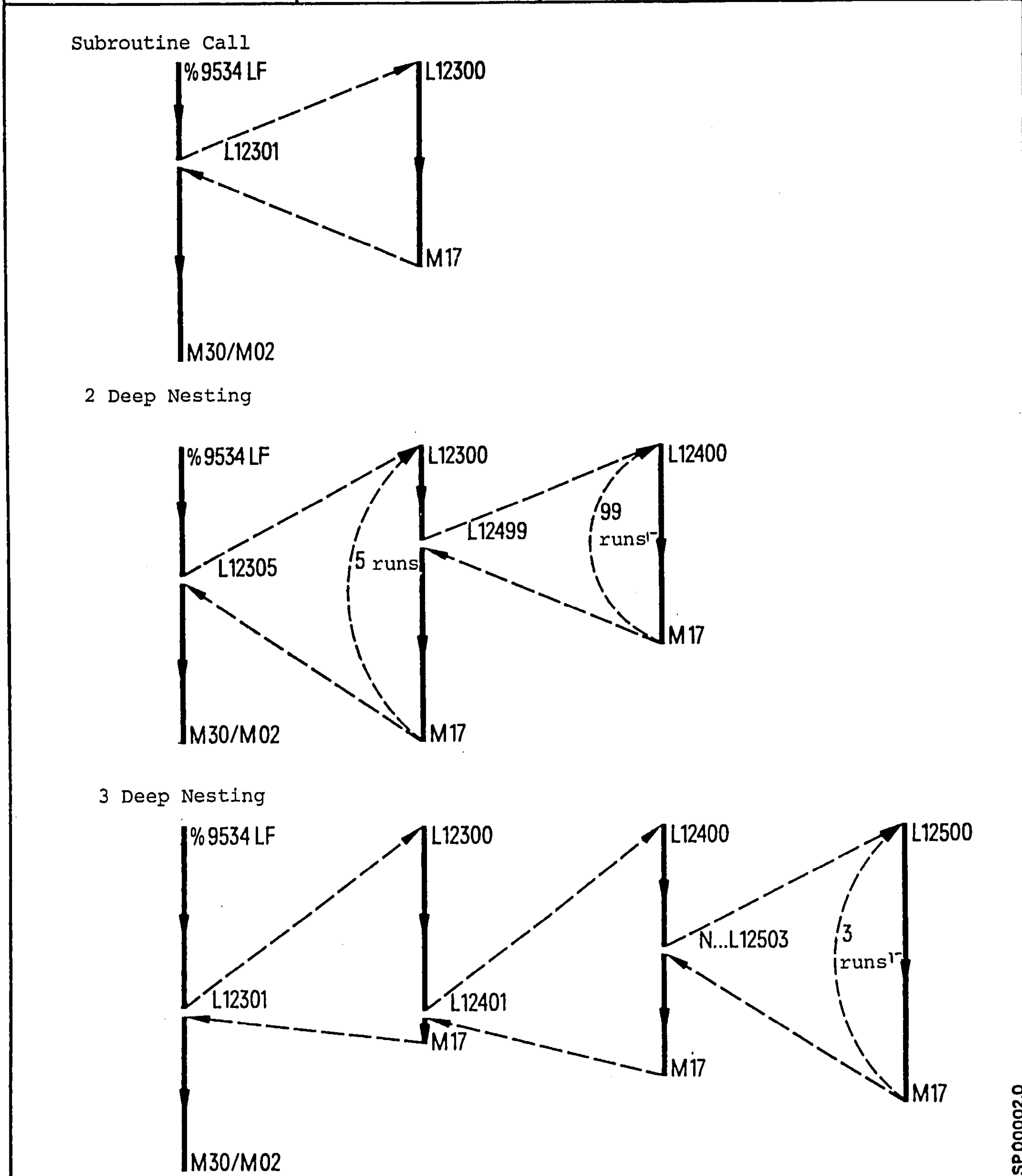
L123 01

- Call with 2 to 5 digits
- 01: Number of complete sweeps has to be set with 2 digits. No definition means only one complete sweep.
- 123: The number of the subroutine has to have 2 to 3 digits (01.. 999)

A subroutine call is not permitted in a block containing M02, M30, or M17. If a subroutine call is made when the CRC (G41/G42) is selected, the CRC is active according to the traverse information programmed in the first and last subroutine block (according to chapter 8.1.6 - "Block without traverse information").

1.9 Subroutine Call, Subroutine Nesting

Part program % 9534 LF	Subroutine L 12300	Subroutine L 12400	Subroutine L 12500



1.10 Perforated Tape Format

Subroutine % {SR} LF | L12300 LF | N1 . . . LF | N2 M17 LF

Leader — Rewind stop and Program start — Subroutine 123 — End of subroutine

L12400 LF | N1 . . . . LF | N2 (Drilling cycle) . . . . LF | N . . M17 LF

Subroutine 124 — (Comment) — End of subroutine

L12500 LF | N . . . . . LF | N . . . . . LF | N . . M17 LF | N . . M02 or M30 LF

Subroutine 125 — End of subroutine — End of subroutine block

Axle Program | % {1234} LF | (Gaging sequence)

Leader — Part program 1234 — (Comment)

N . . . . . LF | N . . . . . LF | N200 M02 or M30 LF

Part program — Part program end

% TO LF | G92 D01 . . . . LF | G92 D02 . . . . LF | M02 or M30 LF

Tool offsets — End of tool offset load

% Z0 LF | G59 N1 . . . . LF | G59 N12 . . . . LF | M02 or M30 LF

zero offsets — End of zero offset load

{ } characters between braces may be ommitted

SR Subroutine

The input sequence for the above tape is arbitrary.

The classification of the memory in part programs and subroutines takes place automatically.

Zero offset and tool offset are inserted in the corresponding memory zones under the code TO (Tool Offset) and ZO (Zero Offset).

1.11 Tape format for program deletion

Using this function it is possible to delete main programs and subroutines in any particular order from the universal input/output interface.

PROGRAM DELETION	- Leader
% CL LF	- Identification (CLEAR)
% 1234 LF	- Delete part program %1234
% 1 % 1200 LF	- Delete part programs %1 to %1200
L10 LF	- Delete subroutine L10
L11 L99 LF	- Delete subroutines L11 to L99
L81 LF	- Delete subroutine L81
M30 or M02 LF	- End of program identification M30 or M02

Example:

% CL LF	% 1 LF	L55 LF	%1 %1200 LF	L11 L99 LF
Program deletion	Delete program %1	Delete subroutine L55	Delete part programs %1 to %1200	Delete subroutines L11 to L99

L81 LF	M30 or M02 LF
Delete subroutine L81	End of program deletion

Attention:

Subroutines L80-L99 and L900-L999 may not be deleted with cycle lock active !

2. Path Information

2.1 Motion Dimension

Rotary Axes

- Are axes set as rotary axes, they must be specified as such with a machine parameter.
- The dimension value for a rotary axis must always be programmed three positions to the right of the decimal point, when decimal point programming is active, even though the rest of the dimension input is in the 10^{-4} inch system.
- Rotary axes can be programmed to ± 256 revolutions. This represents a range of ± 92159.999°.

A dimension word consists of an axes specific address and a dimension value. The dimension value is stored under an axis address.

8M:

X, Y, Z and for the fourth axis addresses A, B, C, E, Q, U, V, and W can be used.

8MC:

X, Y, Z, U, V, W, B, C, E, Q

(Motion dimension with Sprint 8M - see chapter 7.2)

2.2 Mirror Image

Input signals are used to designate which of the primary motion axes (eg. XYZ or any combination thereof) will be mirrored. The primary motion axes are defined by machine parameters. The mirroring of axes results in the sign reversal of the dimension values.(or exchange)

For the Primary Motion Axes the Following Holds True

- Sign reversal of the dimension values (including G92 actual position register store)
- Rotation direction G02 - G03; G03 - G02 Changed from G02-G03 to G03-G02
- Single axis mirrored in the CRC plane ; G41→G42 changed to G42→G41.
- Dual axis mirrored in the CRC plane: G41 → G41, G42 → G42 , G02 → G02, G03 → G03
- Length offset values with slected CRC

The Following Values are not Mirrored:

- Length offset dimensions with selected CRC
- Zero offset dimensions
- DRF - offset
- Preset offset

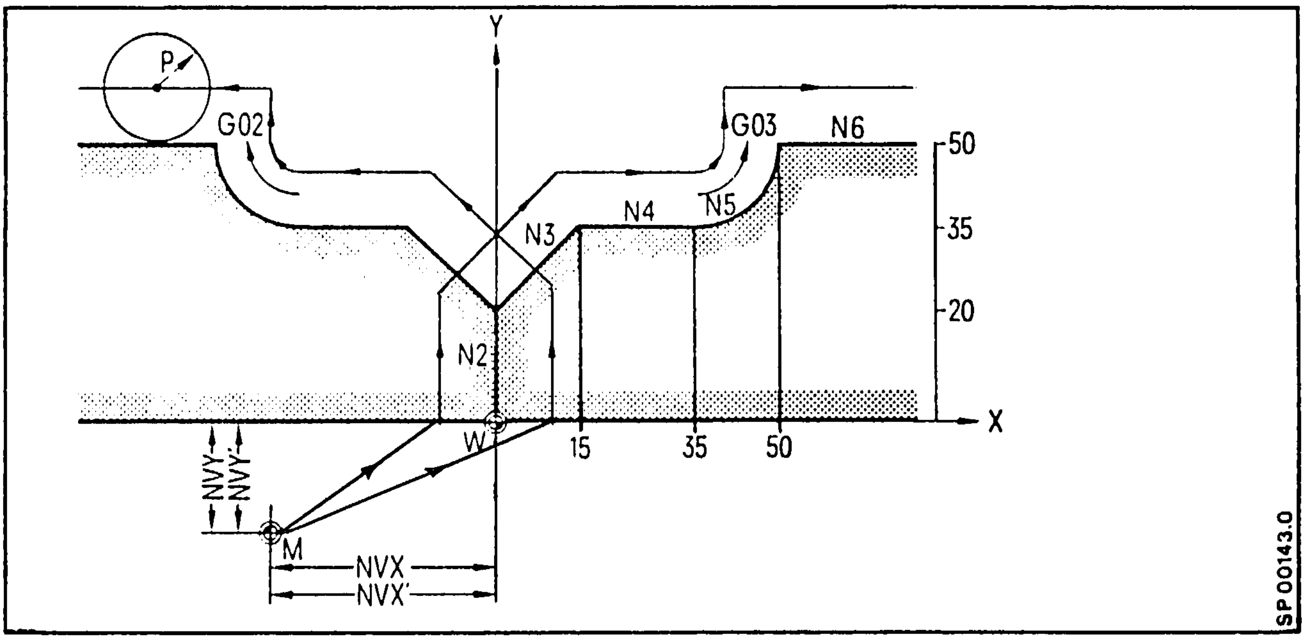
Mirroring the primary motion axis also mirrors the part contour.

Note:

Because of a quick block changeover, up to 11 blocks are temporarily stored. If the control is to react to the "mirror image" input signal (e.g. selected by a M-function) in the following block, then care must be taken that the intermediate memory is cleared. The intermediate storing can be prevented by programming L999 immediately after the block selecting mirror image (see also chapter 5.7).

Example: Mirroring the X Axis

Mirrored Part Programmed Part



M = machine zero point
W = work piece zero point
P = cutter radius
ZO = zero offset

________ work piece contour

--->---->- the cutter center path

```
N1 G00 G90     G64 G41 D01 X0. Y0. M03 S56   LF
N2 G01      Y20.                             LF
N3     X15. Y35.                             LF
N4     X35.                                  LF
N5 G03 X50. Y50. I0. J15.                    LF
N6 G01 X . . .
```

3. Preparatory Functions

The preparatory functions describe the manner in which the machine slide is to move, the method of interpolation, the dimensioning mode, the timed delay of program execution, and the activation of specific operational modes in the control.

The preparatory functions are categorized into groups G1 thru G14 (see the programming key).

A programmed block contains only one preparatory function from each of the 14 groups. When more than one preparatory function of the same group is programmed, the last programmed function is valid, the others are ignored.

On control turn on, reset, or end of program, the control returns to its default state. It is not necessary to program the default preparatory functions.

Modal preparatory functions can only be altered by programming other preparatory functions from the same function group.

3.1 G90/G91 Absolute and Incremental Dimension Programming

Absolute Dimensioning G90

In absolute dimensioning all dimensions are in reference to the part zero dimension. Absolute dimensioning simplifies entry and exit from a program and also makes part geometry program corrections easier.

Incremental Dimensioning G91

An incremental dimension defines the path departure with respect to the present position. Incremental dimensioning is advantageous in subroutine programming.

Note:

A zero offset is always active with absolute and incremental programming.

With incremental programming the settable zero offset must be cleared.

It is suggested that the first program block be programmed using absolute dimensioning.

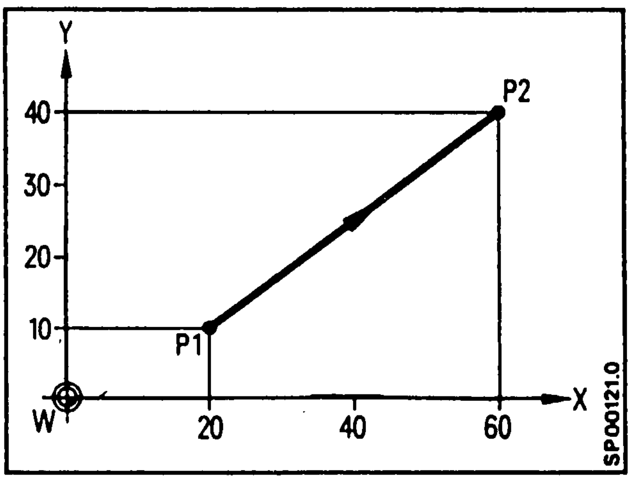
(See also section 8.2.3)

3.2 G00 Rapid Traverse

A block programmed with G00 will traverse in a straight line at the highest possible rate to the programmed position. The control monitors each axis traverse rate so that the maximum allowable rate (machine parameter) is not exceeded.

When programming more than 3 axes, the three axes programmed first determine the traversed path speed. If one of these three axes has a zero movement then alarm 306 results. Basically in order to ensure optimum acceleration the axes with the largest distances to move should be programmed first.

The preparatory function, rapid traverse (G00), automatically causes a controlled velocity decrease (G09) near the programmed endpoint for precise positioning. Programming G00 will not cancel the feed function. The feed function will still be active when programming a G01 following a G00.



W - Part zero point

Absolute Dimensioning

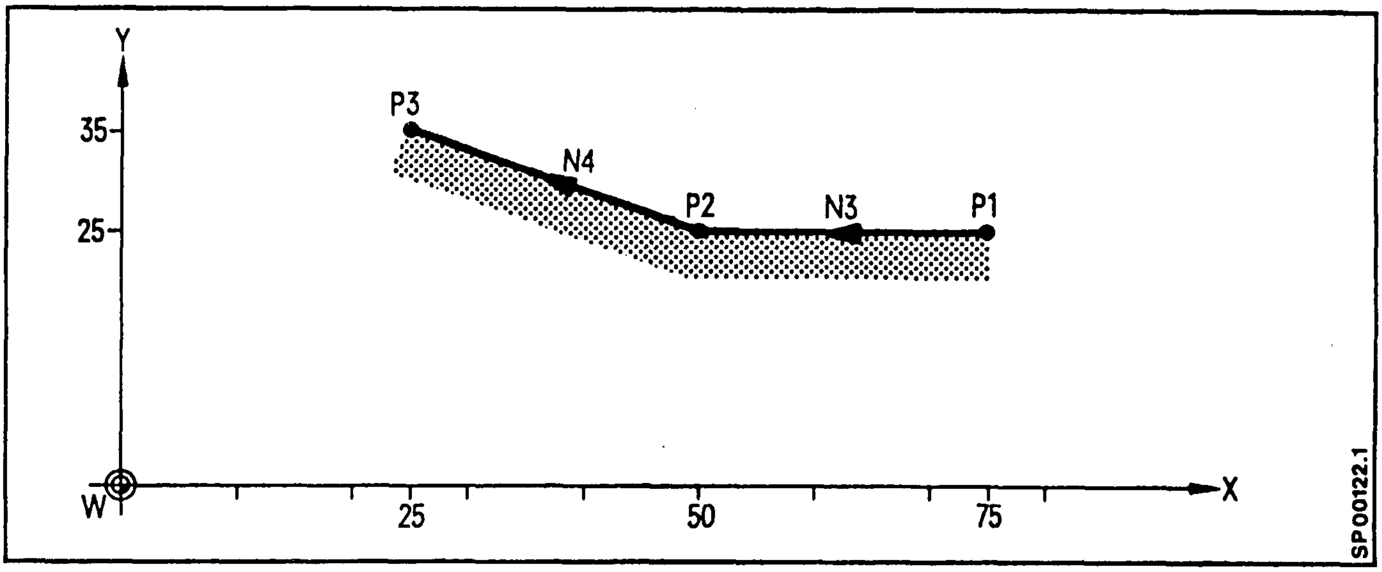
N . . . G00 G90 X60. Y40. LF Tool traverses from P1 to P2

Incremental Dimensioning

N . . . G00 G91 X40. Y30. LF Tool traverses from P1 to P2

3.3 G01 Linear Interpolation (Reset state 1st G-group)

The tool traverses with the stored feed rate in a straight line to the programmed end point. The vectorial velocity is held constant.
If more than 3 axes are programmed then the feedrate is calculated from the first three programmed axes and the path feedrate held to this. In order to maintain optimum acceleration characteristics the axes with the largest distances to move should be programmed first.
A straight line path movement at any angle is possible.
With linear interpolation 4 out of 4 axes (8M/Sprint 8M - presupposition: 3D-Interpolation - ; not 8ME/Sprint 8ME) and 5 out of 10 axes (8MC - presupposition: 3D-Interpolation -; not 8MCE) can be simultaneously traversed. G01 remains modal.



Incremental Dimensioning

```
N3  G91  G94  G01  X-25.  F1000  LF   (P1-P2)   G01 remains active
N4                 X-25.  Y10.   LF   (P2-P3)   until another word
                                                from the first G..
                                                group is called.
                                                G94 F in feed/minute.
```

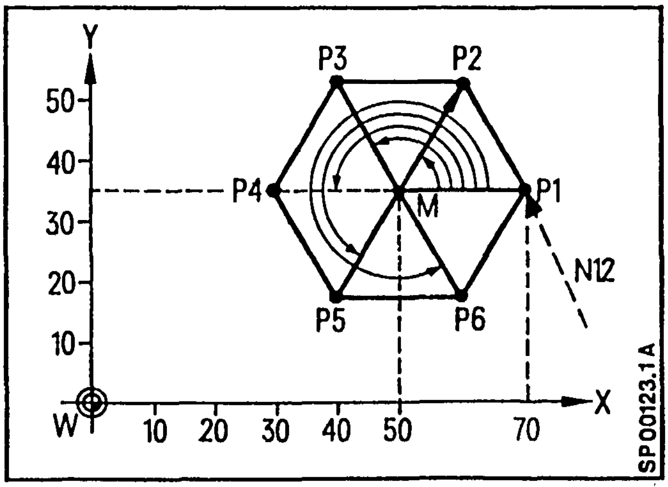
Absolute Dimensioning

```
N3  G90  G94  G01  X50. Y25. F1000  LF  (P1-P2)
N4                 X25. Y35.        LF  (P2-P3)
```

3.4 G10/G11 Polar Coordinate Programming*

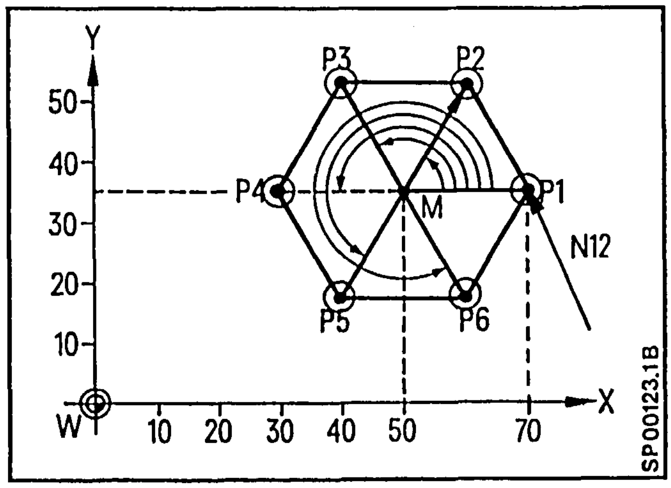
G10 Linear Interpolation Rapid Traverse
G11 Linear Interpolation Feed (F)

Milling of a hexagon head




```
N12 G90 G11 X50.Y35.P20.A0 LF  (P1)
N13                  A60. LF   (P2)
N14                  A120.LF   (P3)
N15                  A180.LF   (P4)
N16                  A240.LF   (P5)
N17                  A300.LF   (P6)
N18                  A  0.LF   (P1)
```

Tracing of boring positions




```
N11 G81                                LF
N12 G90 G10 X50.Y35.P20.A0.            LF
N13     G10 X50. Y35. P20. A60.        LF
N14     G10                       A120.LF
N15     G10                       A180.LF
N16     G10                       A240.LF
N17     G10                       A300.LF
N18     G80                            LF
```

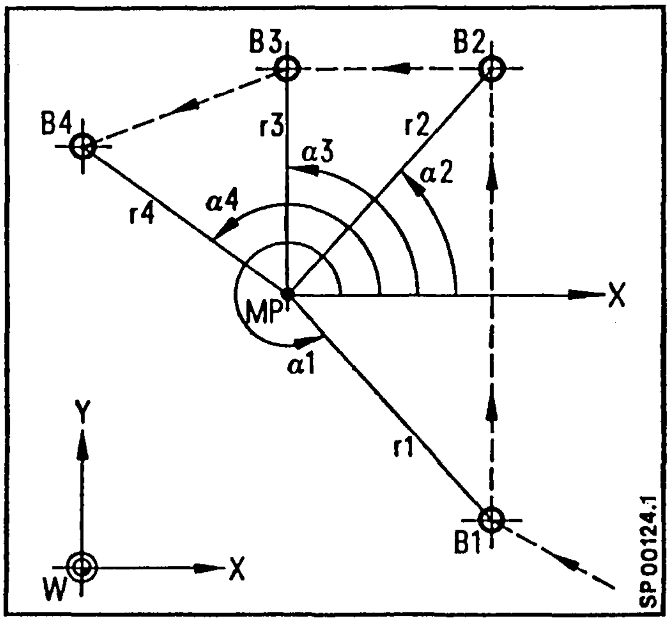
G81: Boring cycle see chapter 7, Cancellation with G80.

Block 13: X.. Y.. Centre point in the polar coordinate system
P.. A.. Position in the polar coordinate system
Radius and angle

The angle refers always to the positive axis which is programmed first (X-axis here).The positive direction of the first programmed axis is equivalent to an angle of 0°,whereas for the second is 90°.

- The angles are defined in absolute and positive decimal degrees. Resolution: 0,00001°
- When fist programming the polar coordinates, both centre point coordinates must be programmed in the datum dimension. It is recommended to program both centre point coordinates.
- The centre point is modal and can be reprogrammed. On "End of Program" (M02/M03), the centre point dimension is cleared.
- The first time polar coordinate programming is used, both centre point coordinates in absolute dimensions must be defined. The incremental departure of the centre point (with G91) is always referenced to the previously programmed centre point.

Using polar coordinate programming for multiple hole drilling where all hole dimensions are with respect to a common center point



<u>Program:</u>

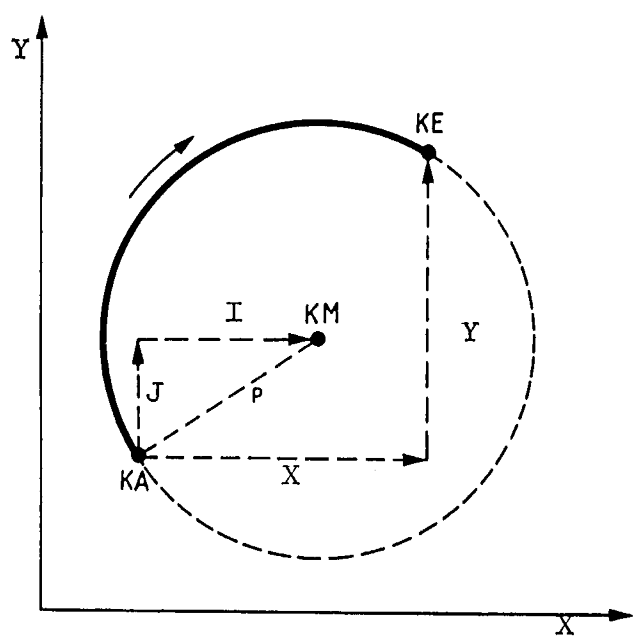
```
N101 G10 G90 G81 X (Center Point) Y (Center Point) A(a1) P(r1) LF
N102 G10      |                   |                A(a2) P(r2) LF
N103 G10      |                   |                A(a3) P(r3) LF
N104 G10      |                   |                A(a4) P(r4) LF
N105         G80                  |                            LF
              |                   |                      |
              |                   | The hole position in polar coordinates
              | Center point of the polar coordinates-system
              |
   Drilling cycle (see Chapter 7)
```

<u>Program Execution:</u>
- move to hole location B1
- preform the desired operation
- move to hole location B2
- preform the desired operation
- etc.
- G10 must be programmed in each block as G81 terminates with rapid.

3.5 G02/G03 Circular interpolation

The interpolation parameters together with axis commands determine the circle or arc. The starting point "CS" is determined by the previous block. The end point "CE" is fixed by the axis values of the plane in which the circular interpolation is programmed. The circle centre point is determined by the interpolation parameters "CC".

a) either through the I, J and K vectors, sign dependant, from a range of 0 to 360^{o}. I in X-direction, J in Y-direction, K in the Z-direction. The sign results from the coordinate direction from the start point to the centre point.



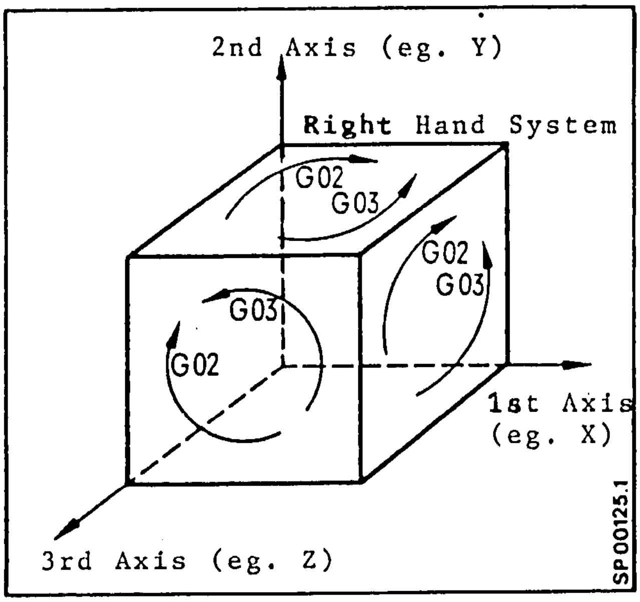
b) or directly through the radius P (option)

+P Angle less than or equal to 180^{o}

-P Angle greater than 180^{o}

Radii should not be programmed when the angle to be traversed is 0^{o} or 360^{o}. In these cases the full circles must be programmed using the interpolation parameters I, J, and K.
Circular interpolation is possible in 2 out of n axes !

The direction in which the arc is traversed is determined by G02 or G03.



Note:

Because of the plane selection (chapter 3.21) follows:
In order to obtain a right hand system in the 3 primary axes, they must be programmed in the following order:

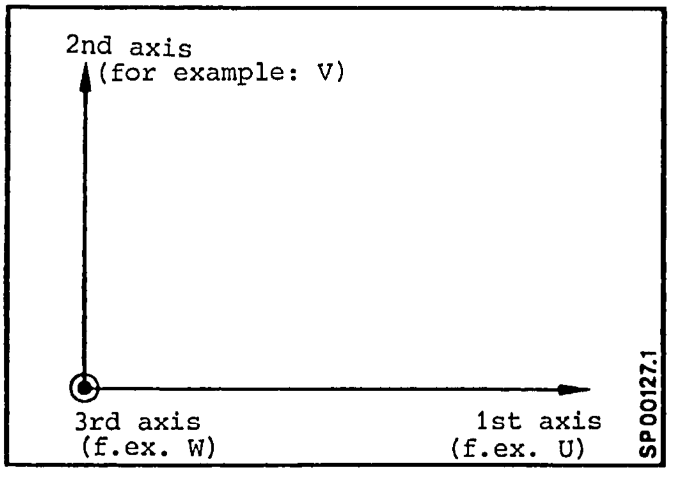
	X...	Y...
	Z...	X...
and	Y...	Z...

3.5.1 Circular Interpolation using Interpolation Parameters

(Circular Interpolation using Interpolation Parameters with Sprint 8M - see chapter 7.3)

The starting point of the circle or arc is determined by the previous block. The end point is given by both end point coordinates.
The circle centre is determined by the interpolation parameters.
When selecting the plane for the circular interpolation the sequence of both programmed coordinates for the end point of the circle is evaluated. In order to obtain a right-handed coordinate system, the end point coordinates of the circle have to be programmed in a fixed sequence.

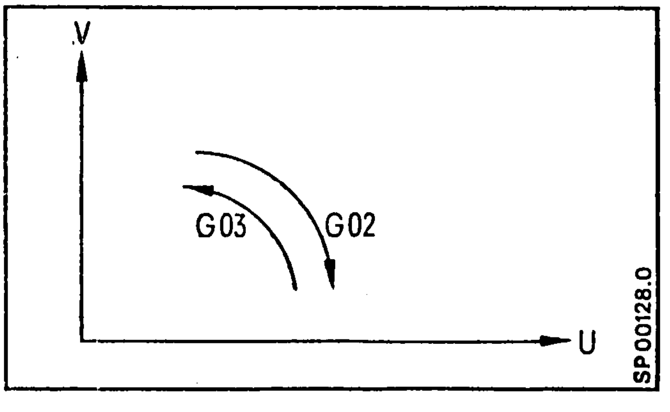
Right-handed coordinate system

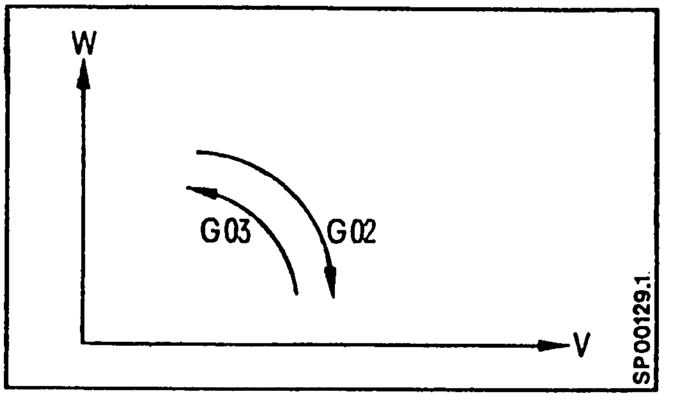


Note:
The axis which is programmed first is always pointing towards the right; the axis which is programmed second is perpendicular to the axis which is programmed first and points upwards in positive direction.
The axis which is programmed third is perpendicular to the first and second axis and points out of the plane that is selected by the first and second axis in positive direction.
(see also Figure SP 00127.1)

Examples:

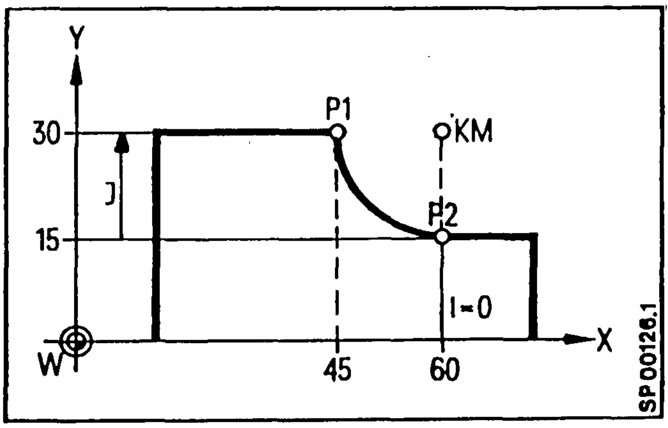
```
G02 U... V... I... I...     G02 V... W... I... I...
G03 U... V... I... I...     G03 V... W... I... I...
    |2nd axis                   |3rd axis in the coordinate system
    |1st axis                   |2nd axis in the coordinate system
```





The interpolation parameters I,J,K are equivalent to one another. I,I,I may be written to the same effect. The assignment of the interpolation parameters to the indivual axes is determined only through the order in which they are written. If one particular value is zero, e.g. I0, it must nevertheless be written in order to ensure the correct assignment.

3.5.2 Example for circular interpolation using interpolation parameters



Input in absolute dimension

N5 G02 G90 X45. Y30. I0. J15. LF

- The tool moves from point 2 to point 1.

N5 G03 X60. Y15. I15. J0. LF

- The tool moves from point 1 to point 2.

Input in incremental dimension

N10 G02 G91 X-15. Y15. I0. J15. LF

- The tool moves from point 2 to point 1.

N10 G03 X15. Y-15. I15. J0. LF

- The tool moves from point 1 to point 2.

3.5.3 Circular Interpolation by specifying the radius (P) *

The starting point of the circle or arc is determinded by the previous block. The end point is given by both of the axis values (e.g. X and Y). The circle centre is defined by the signed radius.

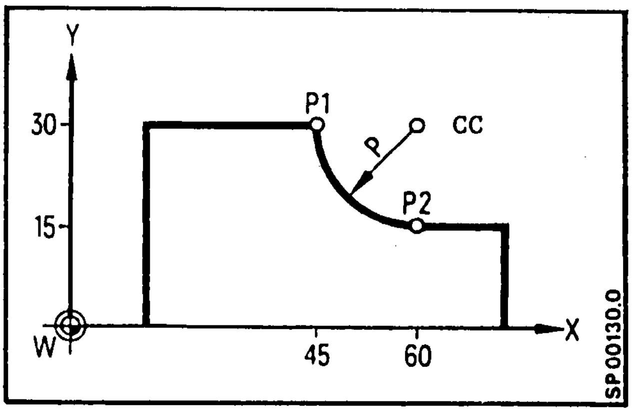
The sign of the radius value is given according to the size of the traversing angle.

smaller, equal	180°	P+
bigger	180°	P-

No radii may be programmed, when the distance between the circle end point and circle start point is less than 10µm. A complete circle must be programmed using the interpolation parameters I, J or K.

3.5.4 Example: Circular Interpolation by specifying the radius

The circle centre point is determined by the signed radius.



N5 G03 G90 X60. Y15. P15. LF

The tool moves from point 1 to point 2.

N10 G02 X45. Y30. P15. LF

The tool moves from point 2 to point 1.

3.5.5 Helical Interpolation *

Presupposition: 3D-Interpolation*
(Helical Interpolation with Sprint 8M - see chapter 7.4)

Helical interpolation is possible between any three perpendicular axes. A block is programmed with one arc path and one linear path. The linear departure must be perpendicular to the plane in which the arc motion is generated. The programmed feed is maintained for the arc motion.

8MC

```
N . . . G91 G02 X200. Y0. I100.  J0 Z200 LF
N . . . G91 G02 X200. Y0. B+100.    Z200 LF
```

- The circular interpolation plane (both motion dimension words must be programmed, 1st before 2nd axis)
- Interpolation parameter or radius
- Linear interpolation path e.g. Z axis

8M

```
N. . . G91 G02 X200. Y0. Z200. J.0 I100       LF
N. . . G91 G02 X200. Y0. Z200.          B+100. LF
```

- The circular interpolation plane (Given by G17, G18 or G19)
- The motion dimension for linear interpolation may be written before or after B word
- Interpolation parameter or radius word

Note:

The helical interpolation is not possible with 8ME and 8MCE.

3.6 G33 Threading

For special programming features with Sprint 8M see section 7.5
With boring and milling machines, threads may be cut by using a boring tool or a facing head.
G33 realises a relationship between the main spindle speed and the feedrate. A spindle encoder generates 1024 pulses per spindle revolution. These pulses are evaluated by the control which in turn influences the feedrate which it delivers to the servo drives. In such a way the spindle speed dictates the feedrate, so that feedrate is no longer applicable. Nevertheless the feedrate previously programmed under address F is stored for subsequent use.

In order to produce a thread in several passes the axis will feed when the zero marker pulse initiates the thread cutting cycle. This ensures that threading commences always with the same workpiece-tool angular displacement. All passes must be carried out at the same feedrate (spindle speed) in order to avoid variance in the following error.

The spindle speed and direction need to be programmed prior to the threading in order to allow the spindle to reach speed. The programmed thread lead should take into account the required acceleration time of the axis drive. Similarly a run out should also be considered to allow for axis deceleration. A sequence of several G33 threading blocks is generally allowable. In order to ensure that all spindle pulses may be evaluated there is a minimum thread length per block "Smin": this is calculated as follows:

$$S_{min}(mm) = 1.7 \cdot 10^{-5} \cdot n\ (rpm) \cdot K\ (mm/rev) \cdot t_A(msec)$$

n = spindle speed, K = thread pitch ,

t_A = duration of auxiliary function output (e.g. 20msec)

- If deceleration is wanted at the end of the block, G09 has to be programmed.
- Thread length plus acceleration and deceleration length are programmed under the corresponding position data whereby the tool width has to be taken into consideration as well.
- The thread lead is specified under addresses I, J, K.
- There is no fixed relationship between threading pitch addresses I,J,K and axes addresses.
- The address pairing for thread cutting:
  the 1st thread lead correlates to the 1st axis value,
  the 2nd thread lead correlates to the 2nd axis value.

- I, J, K parameters are incremental dimensions which specify the lead in feed per revolution. The dimension value is unsigned. The programming resolution for the thread lead is 0.001 mm/rev. (0.0001 in/rev.).
- When thread cutting the feed override, feed hold, spindle speed override, and single block switches are disabled. (from 02-mach. data!)
- Pairing of thread lead and spindle speed - see chapter 8.2.4.

3.6.1 Constant lead tapered threads

(Constant lead tapered threads with Sprint 8M - see chapter 7.5)

For constant lead tapered threads, the thread lead is programmed for the leading axis.

- The leading axis is defined as the axis traversing the longest distance.
  With similar distances the first programmed axis is the leading axis.
- Only if the second programmed axis is leading should two interpolation parameters be programmed. In such cases the value of the first interpolation parameter could be zero.

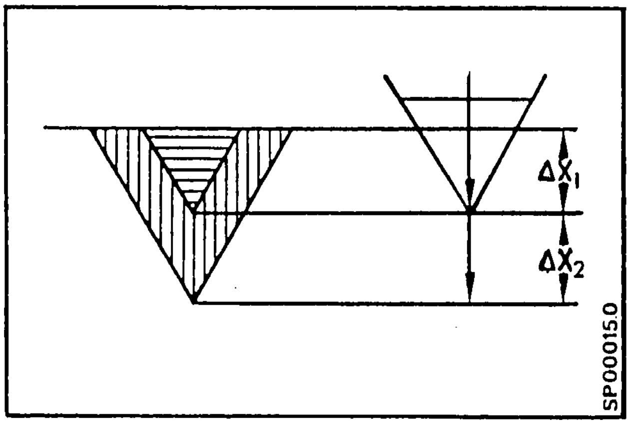
Example: (incremental dimensioning)

```
G33  X20.  Z10.  I0.2            Thread lead  =  0.2  mm/rev.

G33  X10.  Z20.  I0.2          Incorrect programming; both interpolation
G33  X10.  Z20.  K0.2          parameters must be programmed,since the
                               second programmed axis is leading.
G33  X10.  Z10.  J0.2                        =  0.2 mm/rev.
G33  X10.  Z10.  K0.2                        =  0.2 mm/rev.
G33  X10.  Z10.  I0.2  K0.2                  =  0.2 mm/rev.)
                                                            } permissible
G33  Z10.  X20.  I0.2  K0.2                  =  0.2 mm/rev.)
```

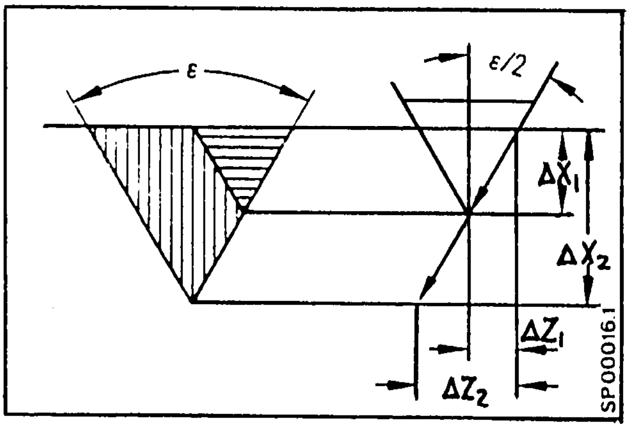
3.6..2 Feed Direction

Two methods can be used to thread cut. The tool can feed perpendicular to the cutting direction or parallel to the cutting direction.



"Perpendicular to the Cutting Direction"

When only one edge of the cutting tool is to cut both axis must feed. The tool is fed in the direction of cut and perpendicular to the cutting direction before the start of the next threading pass.



"Cutter Edge Feed"

$$\Delta Z = \Delta X . \tan \frac{}{2}$$

3.6.3 Variable Lead Thread

The thread lead can be modified by programming several contiguous thread cutting blocks. Within a block, the thread lead is constant. The region of constant thread lead can, if desired, be less than a single revolution. Subsequent thread cutting blocks will execute without waiting for the next zero marker pulse of the pulse encoder.

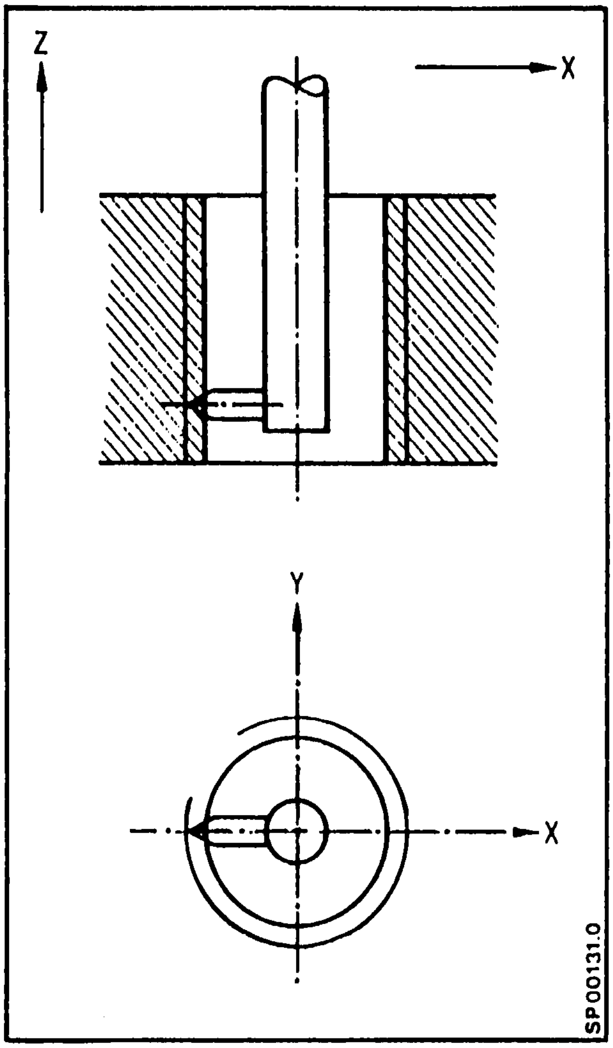
3.6.4 Multiple Thread

A multiple thread is programmed in the same manner as a single thread. After the first thread is cut, the threading start point is displaced by an amount equal to the pitch circle before the thread cut sequence is repeated.

3.6. 5 Thread Cutting With a Boring Bar

With the work piece stationary, a thread can be cut by simultaneously rotating and feeding the boring tool. It is necessary to program the bar to retract to the start point:

Before the bar is retracted, the spindle must be stopped in an oriented position (M19S). The bar is moved out of the cut and with the stopped spindle is retracted to the start position.



Example: Threading a Blind Hole with a Boring Bar

```
N20 G90 G00 X100. Y . .       S45 M03      LF
N25                     Z200.              LF
N30 G33                 Z120. K10.         LF
N35                           M19 S0.      LF
N40 G00     X105.                          LF
N45                     Z200.     M00      LF
N50         X100.                 M03      LF
N55 G04                 F2                 LF
N60 G33                 Z120. K10          LF
```

Blocks 20, 25: The boring bar is centered over the drilled hole. The spindle is turned on.

Block 30: The first threading cut is made. The thread end position (eg. in absolute dimensions) is programmed under address Z. The thread lead is programmed under address K.

Block 35: The spindle is brought to an oriented stop.

Block 40: The boring bar is moved out of the cut in the X direction.

Block 45: The boring bar is moved out of the hole in the Z direction. It is possible with a programmed stop (M00) to feed the boring bar (eg. manual feed) and take a second cut.

Block 50: The boring bar is centered over the drilled hole at the same time the spindle is turned on.

Block 55: In the event the positioning time in block 50 is shorter than the time it takes the spindle to accelerate to the correct speed, a dwell time of sufficient length must be programmed in block 55. This insures that the spindle has reached the desired speed before beginning the next threading pass.

Block 60: A second threading cut begins.

3.7 Feed acceleration ramp time for thread cutting

For threading it is possible to define a damping time and therefore a feed acceleration ramp during which time the feed axis accelerates to the required feedrate prior to synchonisation with the already rotating spindle. This value, programmed through G92 T... , in effect averages the actual spindle speed over this period. The ramp up time of the drive should be matched to this lead in distance. The smaller the available lead in distance the smaller the ramp up time needs to be. For parts with a greater lead in distance available it is reccomended to program an appropriately longer ramp up time in order to protect the machine from stress due to rapid acceleration.

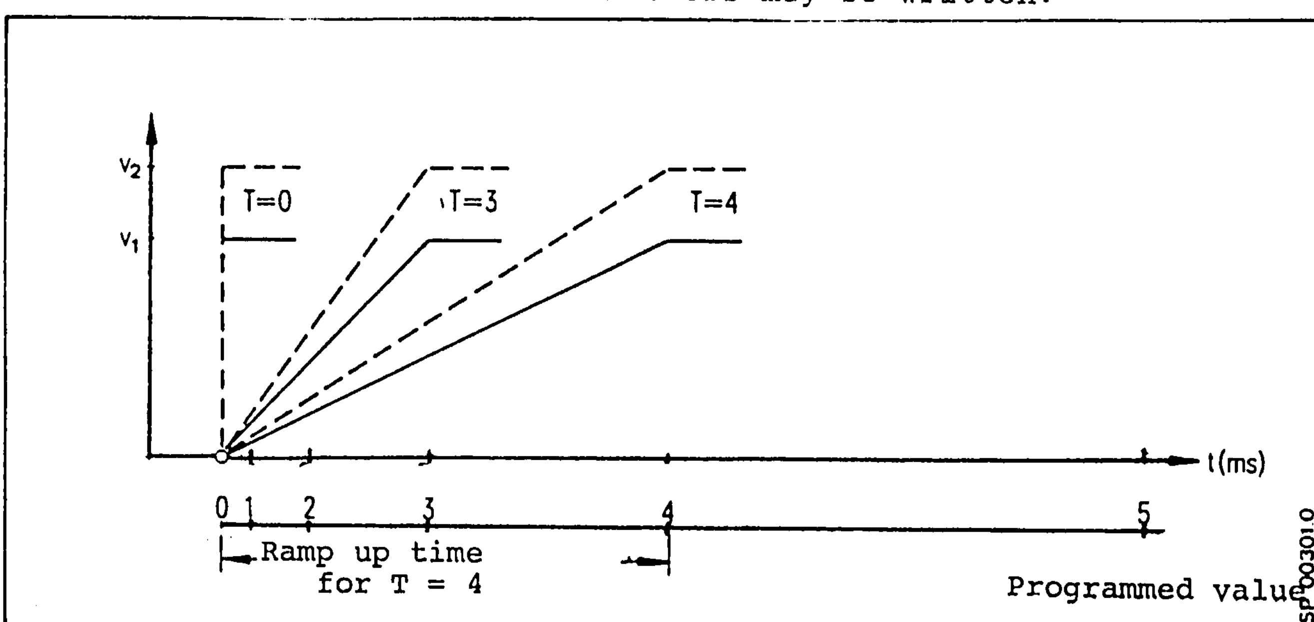
The ramp up time is programmed in a self contained block or may alternatively be input by the operator: N.. G92 T. LF

One of six values may be chosen:

Programmed value with G92 T LF	0	1	2	3	4	5	
Damping time/ ramp up time to the threading feedrate(msec)		10	30	70	150	310	8MC ≤ 5 axes
		20	60	140	300	620	8MC > 5 axes
	0	16	48	112	240	496	8M

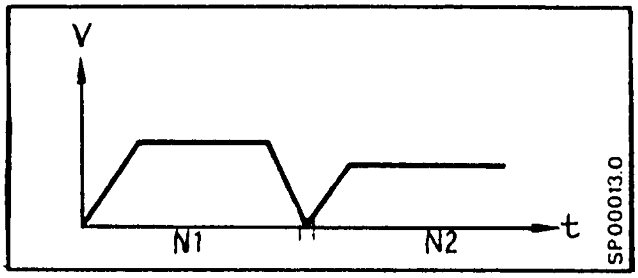
For normal operation T = 3 is reccomended.

In a G92 T.. block no other characters may be written.

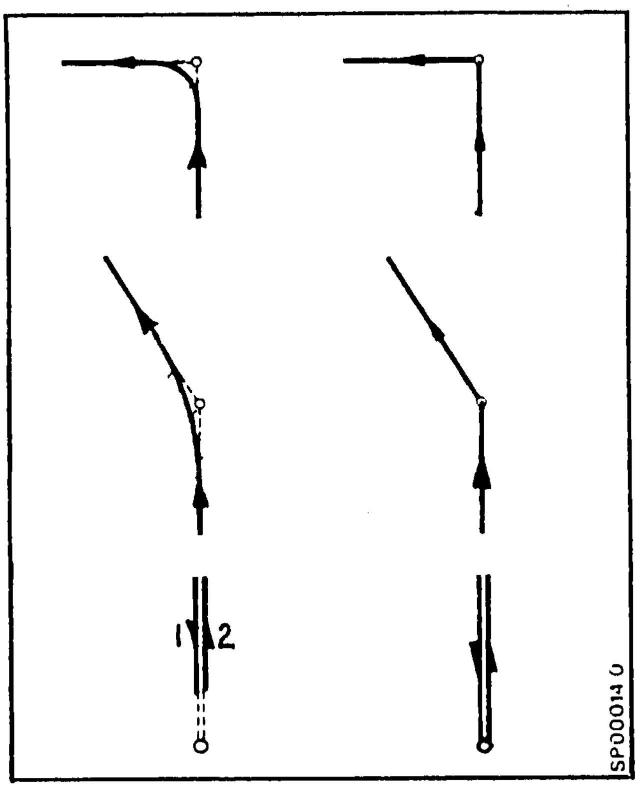


3.8 G09 Deceleration G60 Exact Positioning (Reset state 10th G-group)

With the preparatory functions G09/G60 it is possible to position exactly to a target position (within the "in position band tolerance"). The feed velocity is reduced to zero. The following error is worked to zero.



The preparatory functions G09/G60 are used, for example, to machine sharp corners, for plunge cutting, or when reversing direction. Blocks with G00 need not be programmed with G09. A G09 is automatically preformed with G00. G09 is not modal. G60 is modal and is cancelled with G64 (contour machining). The example shows direction reversal with and without G09/G60.



3.9 G63 Tapping with a Floating Tap Holder

The preparatory function G63 is programmed when tapping drilled holes with a floating holder. The feed axis and spindle rotation are not synchronized.

Spindle speed is programmed under address S with the appropriate feed function programmed under address F. The floating tap must take up length variations resulting from the difference between the tap lead and the lead deviations due to feed rate and spindle speed fluctuations. Sufficient length compensation must be provided on reaching the programmed position to allow for overshoot due to spindle speed run down.

G63 inhibits the feed rate override switch and dependent on the interface design will shut the spindle down when "feed hold" is signalled. The spindle override switch is inhibited. G63 may only be used with linear interpolation G01. A G60 will cancel G63.

3.10 G64 Contour Machining

To prevent dwell marks, the preparatory function G64 is programmed. G64 assures smooth path transitions between contiguous blocks containing path movements, however, a tangential direction change will result in a rounded corner.

3.11 Milling of cylindrical contours G92 P: *

(Function is not possible with Sprint 8M)

	Input System	
Unit Circle Diameter for	Metric	115 mm
Unit Circle Diameter for	Inch (0.0001)	11.5 in.
Unit Circle Diameter for	Inch (0.00001)	1.15 in.

The unit circle diameter is defined by the equation

$$360° = \pi \cdot \text{diameter}$$

$$d = \text{unit circle diameter} = \frac{360°}{\pi} \text{ in mm (inch)}$$

N . . G92 P . . . C LF

Normalized ratio address (P)

Normalized diameter = $\frac{\text{part diameter}}{\text{unit circle diameter}}$ (P . . .)

Rotary axis (eg. C) associated with the normalized diameter (C)

The normalized diameter is modal but can be redefined in subsequent blocks (Resolution: 0.00001). The value is reset with M02/M30.

An axis whose normalized diameter is not equal to one cannot be used to interpolate with more than 2 axes. eg. linear interpolation with more than 2 axes is possible only after the normalized diameter is set equal to one.

Apart from the names of the axes, no other signs may be written in a block with G92 P.. .

With cylindrical interpolation it is possible to machine cylindrical contours by coordinating the motion of a rotary axis with a linear axis while the rotary axis diameter is held constant. Straight contour paths as well as arc contour paths using intersectional cutter compensation can be programmed.

The rotary axis angle is dimensioned in degrees:

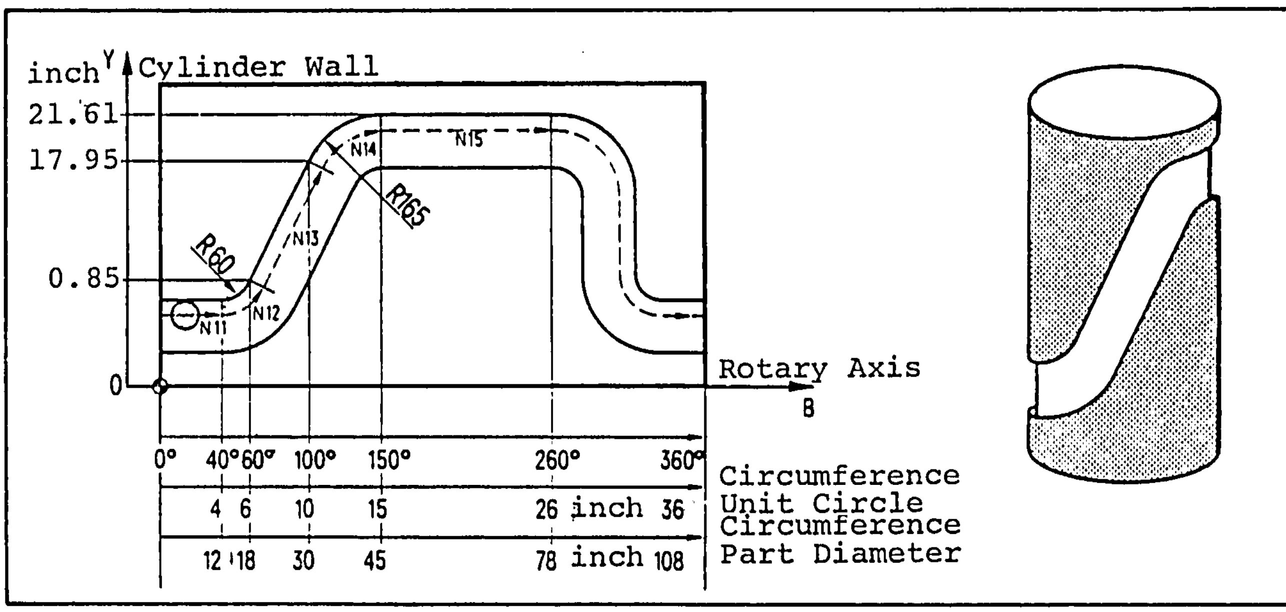
The circumferential dimension is calculated by the control using the previously programmed normalized diameter.

The normalized diameter is defined as

$$P = \frac{\text{part contour diameter}}{\text{unit circle diameter}}$$

and programmed with G92P . . .

The programmed feed function is maintained at the contour surface.




```
N10  G92  P3     B                  LF   Cylindrical interpolation
N11  G01  G42 B40.  Y200            LF   mode is selected
N12  G03  B60    Y0.85   P+60.      LF
N13  G01  B100.  Y17.95             LF
N14  G02  B150.  Y21.61   P+165     LF
N15  G01  B260.                     LF
N26  G92  P1     B                  LF   Cylindrical interpolation
                                         mode is deselected
```

3.12 G04 Dwell Time

The dwell duration time is programmed under address F, but can be programmed under address X.

for F address programming: Range 1 msec - 99999 msec
for X address programming: Range 1 msec - 99999999 msec

A block programmed with G04 may not contain other functions.

eg.

N . . G04 F11.5 LF

Dwell time 11.5 sec
always an unsigned number

When necessary, several contiguous blocks containing dwell functions may be programmed.

Dwell times are programmed when a tool is to cut free of the part and may be used for speed change and machine switching functions. G04 is not modal.

3.13 G70/G71 Input System

G70: Input dimensioning system is in inch

G71: Input dimensioning system is in metric

The default mode is defined by a machine parameter. It is not permissible to change the input system in a running program. A change may be programmed in the first program block. The dimensional field width for individual systems is shown in section 8.2. The display format is always with respect to the currently selected dimensioning system (see operator manual pg 2-4).

When a change is made from G70 to G71 or G71 to G70, the operator or programmer must insure that all relative user data (see operations manual pg. 4-17 and 4-18) is set correctly for the desired input systems.

3.14 G25/G26 Programmable Safe Zone

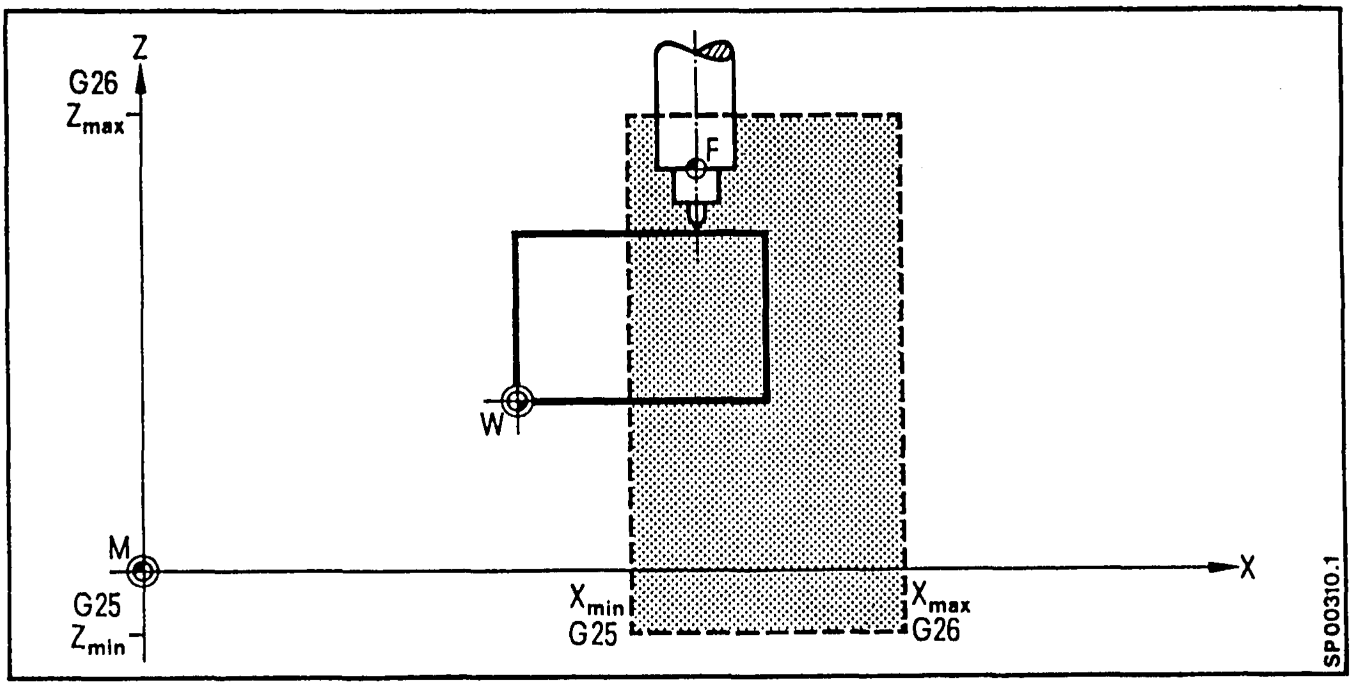
A programmable safe zone protects the machine from programming and operator error. When the safe zone envelope is reached, the feed is cancelled (program stop and alarm) and the following error is worked to zero.

The programmable safe zone works only in automatic mode and is treated by the NC as if a software limit switch had been actuated. The safe zone is defined with respect to the machine zero point. A block programmed with G25/G26 may not contain other functions.

N . . . G26 X . . Z . . LF

G25 Lower safe zone limit
G26 Upper safe zone limit

Coordinate dimension for the limit
(1 to n axes)

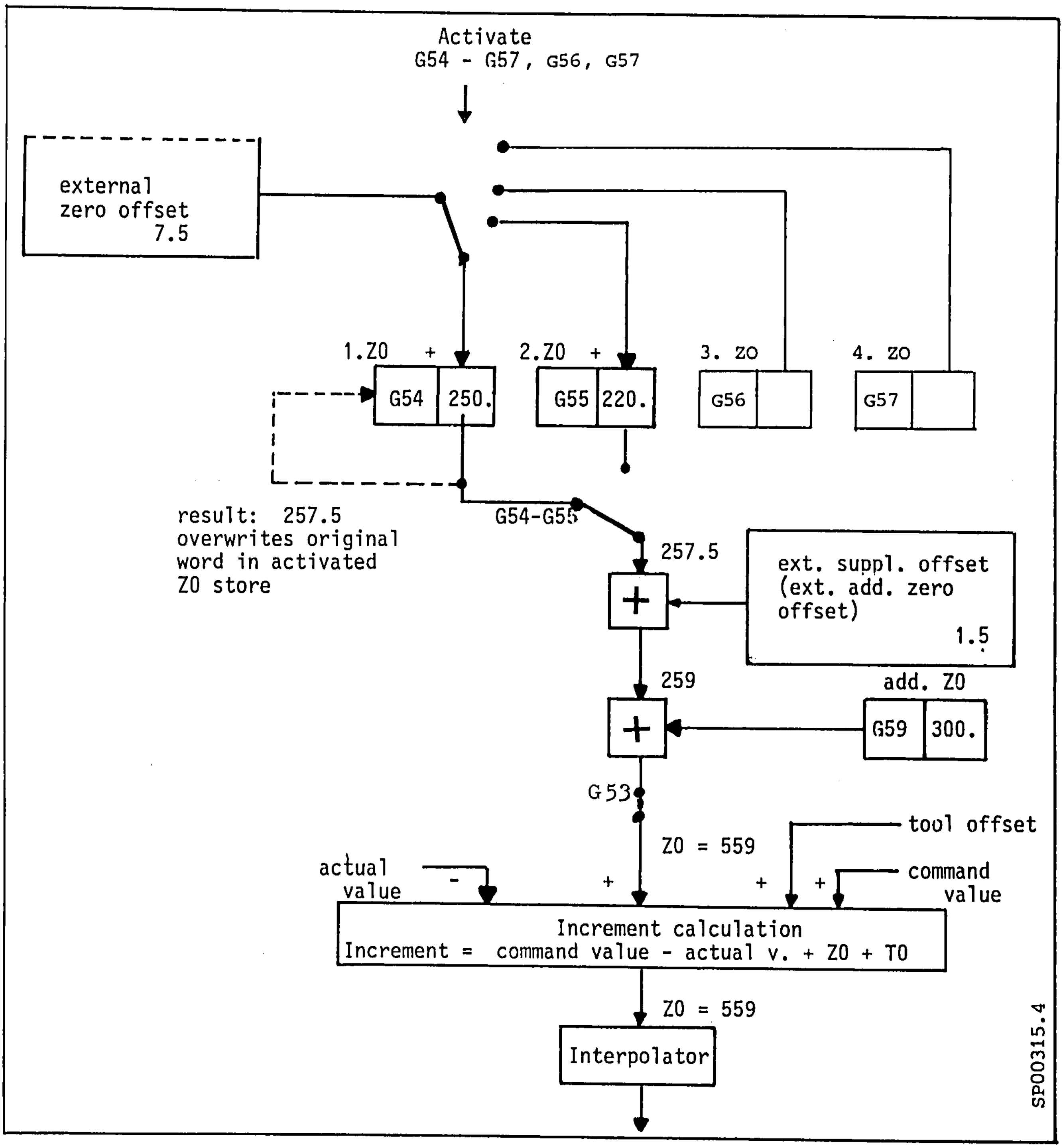


The point F (tool centre point) is allowed to traverse within the dotted zone. As soon as the tool leaves the set working zone or happens to be outside of this zone at the program start, an alarm is indicated and all machines are set still.

3.15 Zero point offsets (ZO)

ZO = set. ZO (G54-57) + add. ZO (G59) + ext. ZO (PC) + ext. suppl. ZO

The zero point offset is the difference between the workpiece zero point (to which the measurements are related) and the machine zero point.



Attention: With CRC selected the zero offset must not be modified.

3.15.1 G54/G55/G56/G57 Settable Zero Point Offset

(G54 is the reset state of the eighth G-group)

(Settable zero point offset with Sprint 8M - see chapter 7.6)

Values for the zero point offset for each axis can be entered into the control manually, via the operator's panel or using tape. Absolute data blocks (G90) are used to calculate the final block point, when the associated axis is programmed. With incremental data blocks (G91) any change in zero point offsets is taken into account.

Example:

Change from G54 to G55 in an incremental data block. The resulting difference between ZO (G55) and ZO (G54) is included in the calculation (see block increment calculation, chapter 8.2).

Four or twelve * adjustable zero point offsets per axis can be selected.

When a zero point offset (e.g. G54) is included in the calculation, the external zero point offset originating in the interface control for the corresponding axis is also taken into account (additive ZO plus supplementary offset).

Selecting a Settable Zero Offset

A zero offset is called with G54 thru G57 or with interface generated signals for groups 1 thru 3 *.

ZO per axis	8M/8MC 12*		
Group	1	2	3
Input	N1 - N4	N5 - N8	N9 - N12
ZO 1 ZO 2 ZO 3 ZO 4	G54 G55 G56 G57	G54 G55 G56 G57	G54 G55 G56 G57

Activation of the interface signals e.g. through an M-function. For clarity it is necessary to clear the buffer with L999 (see interface description) and also cancel the cutter radius compensation.

```
N10 G40 X....      LF
N15 M..            LF  Selection of the zero offset group 1 to 3
                       with an M-function
N20 L999 ......LF      Clear the buffer (see section 5.7)
N25 G56   ......LF     Zero offset is called up
```

Loading a Settable Zero Offset

From tape

```
% ZOLF   (ZERO OFFSET)
G59 N1   X . . .  Y . . .  Z . . .  LF  maximum 5 axes per
                                        block programmable.
G59 N      . . .    . . .    . . .  LF  When more than 5 axes,
                                        must be loaded a second
                                        block must be used.

G59 N12 X . . .  Y . . .  Z . . .   LF
M02 (M30)                           LF  M02/M30 in a separate
                                        block.
```

Loading a Settable Zero Offset (G54-G57, Group 1 - 3) in the program

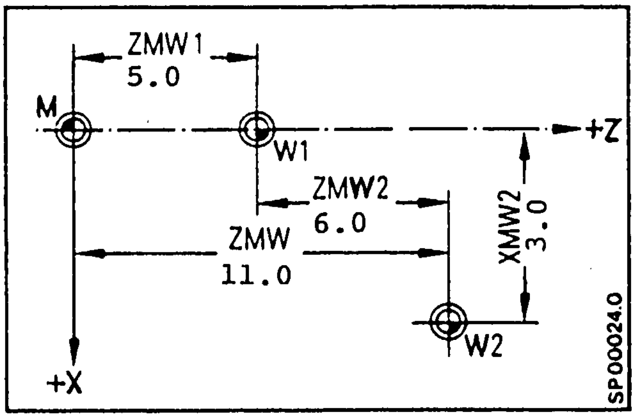
```
N100 G59
N110 G59 N1 X... Y... Z... LF    max. 5 axes per block programmable
N120 G59 N2 .... .... .... LF    When more than 5 axes must be loaded
.
.
N220 G59 N12 X.. Y... Z... LF
N230
```

N1 - N12 designates the zero offset and group division (1 thru 3) The zero offset input is done by the operator. (See programming instruction).

N110 - N220 designate the block number of the blocks that are used for loading the settable zero offset.

3.15.2 G59 Programmable Additive Zero Offset

An additional zero offset can be programmed with G59 under addresses X, Y, Z etc. The programmed value is added to the settable zero offsets.



Settable Zero Offset:

Input Value: $XMW_1 = 0$
$ZMW_1 = 5.0$

Programmed Additive Zero Offset:

Input Value: $XMW_2 = 3.0$
$ZMW_2 = 6.0$

Resultant Zero Offset XMW = 3.0
ZMW = 11.0

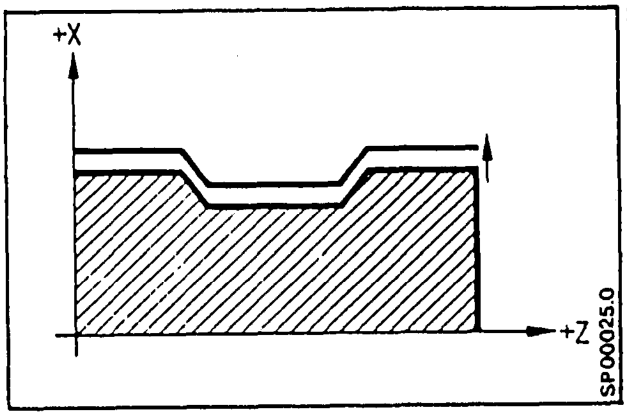
No other information may be programmed in the G59 block.

Example

The contour is programmed in absolute dimensions. To allow for finishing stock, the entire contour can be displaced in X with a programmable (additive) zero offset.

Selected with N . . G59 X . . . LF

Cancelled with N . . G59 X0. LF



Programmable additive zero offset e.g. in X

With M02/M30 or on a program exit, the programmable zero offset is automatically cleared since a new program start will reload the offset value.

3.15.3 G53 Cancelling the zero offset

G53 suppresses blockwise the coordinate displacement achieved by

- settable ZO (G54 - G57)
- programmable additive ZO (59)
- external ZO
- external additive ZO

The tool offset and setting of the actual value store using G92 must be cancelled in separate blocks.

Further, any DRF-offset present will remain active.

Attention with CRC selected:

If only one axis of the CRC plane is programmed in the G53 block then G53 is active also for axis that is not programmed. In such cases CRC should be cancelled previously through G40. In the next block after G53 all the zero offsets are again active.

In the block following G53 all zero offsets are again active.

Example: referred to machine zero point

N1232 G40 D00 ...	- cancellation of tool offset
N1233 G92	- cancellation of any G92 offsets
N1234 G53 X.. Y..	- cancellation of all ZO's and traverse to position in machine system

If the actual value store has to be reset after reference to the machine zero point, a G53 must be written in this block. This ensures that the ZO is ignored.

Example: setting actual value store after reference to machine zero point as for previous example plus:

N1235 G53 G92 X... Y... - setting actual value store

The sequence G53 G92 has to be kept.

3.15.4 G53 Cancelling the ZO from software stand C2

G53 has two different effects which are selectable through machine parameter:

Machine data N424 bit 2 = 1
Reference to machine zero point

Blockwise suppression of:
- -settable ZO (G54-G57)
- -programmable additive ZO (G59)
- -external ZO
- -external additive ZO
- -PRESET shift
- -G92 shift
- -DRF shift (handwheel offset)

Remain active:
- -selected tool offset

Machine data N424 bit 2 = 0
Reference to control zero point

Blockwise suppression of:
- -settable ZO (G54-G55)
- -programmable additive ZO (G59)
- -external ZO
- -external additive ZO

Remain active:
- -selected tool offset
- -G92 shift
- -PRESET shift
- -DRF shift (handwheel offset)

* After programming G53 the DRF shift reamins inactive until reset or end of program.

The positon display refers always to the control zero point.

Example:

Reference to machine zero point

N1232 G40 D00 X... Deselect TO
N123 G53 X... Y..
Cancel all ZO and move to position in machine position system.

Reference to control zero point

N1232 G40 D00 X... Deselect TO
N1234 G53 X.. Y..
Cancel ZO
Move to point in control position system.

ditto. with cancellation of G92 shift
N1232 G40 D00 X.. Deselect TO
N1233 G92 cancel G92 shift
N1234 G53 X... Y...
Move to point in control position system (PRESET/DRF)

The position actual value G53 X... is active in diameter with machine data "diameter programming" set.
It is meaningful to cancel the CRC as with programming G53 X... or G53 Y... the offsets are taken out in both axes.

For G92 Set actual value stores, the following applies:

Set actual stores with reference to machine zero point

N1232 G40 D00
N1234 G53 X... Y...
N1235 G53 G92 X...Y...

Set actual value stores with reference to control zero point

N1232 G40 D00
N1233 G92
N1234 G53 X...Y...
N1235 G53 G92 X...Y...

The sequence G53 G92 is important.

3.16 Position Register Preload G92

The function G92 should only be used for special applications. For normal applications it is recommended to use the settable zero point offset G54/57, the programmable zero point offset G59 and the tool preset D... (separately adjustable from tool wear). (See section 3.24).
No additional character may be written in a block with G92 X... Y...
Exception: Setting of actual value stores after referring to the machine zero (3.15.3): G53 G92 X... Y...
Without G92 the control zero point (S) and machine zero point (M) coincide. The control zero point is the reference point for all internal control calculations. Using G92 X... Y... the control zero point can be displaced with reference to the machine zero point. This function is particularly advantageous, when no program interrupt and restart within the program is anticipated, e. g. machining of batch components with short program cycle times.

Resetting all G92 offsets:

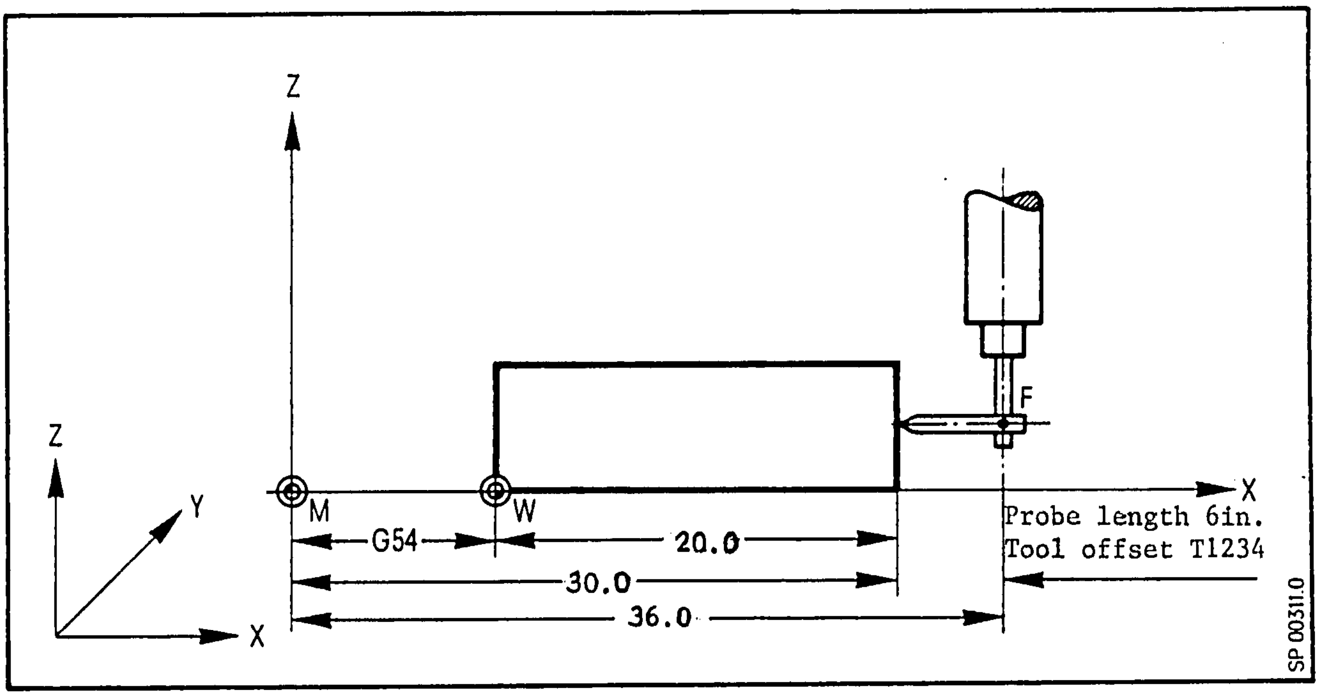
If G92 is programmed alone, i. e. without X and Y address, all summated G92 offsets for each axis are reset.

Example: N... G92 LF

Note: Using constant cutting speed, the spindle speed is derived from the machine actual value which corresponds to "zero" in the turning axis, and not from the actual value reset by G92 X...

Example:

The position of the shoulder on each milled component of a series varies more in the longitudinal axis due to automatic chucking than the available machining offset. To prevent the operator from having to continually adjust the zero point offset, a guage is moved in until it touches the shoulder and the operative block is interrupted. Using G92 and taking guage length into account, this position referred to the workpiece is set as a tool offset and the workpiece length set as the X-position. Only then does the actual machining program start (all dimensions in mm).



Programming:

```
N...                D05  LF
N...  G54  X-9999.       LF         Block interrupt via guage
N...  G92  X200.         LF         Calculation of the actual
                                    position with register
                                    preload

N...      (MACHINING PROGRAM)
```

The control loads the actual position referred to the machine zero by the following calculation:

Calculation of ZO and/or tool offsets via machine parameter	Example (see fig.)	Example without ZO with TO	Example without ZO without TO
G54 zero offset	10.0	0.0	0.0
+ X preload position	20.0	20.0	20.0
+ TO length compensation of the feeler guage	6.0	6.0	0.0
Actual position X	36.0	26.0	20.1

The actual position register is loaded with G92 and is reset to the original position at the end of program (M02/M30).

3.17 G94/G95/G96/G97 Feed F; M36, M37

The programmed feed rate when using cutter radius compensation is maintained on the contour surface. If more than 3 axes are programmed per block, then the first three determine the path speed.
With a rotary axis the feed function is programmed under address F as an angular velocity in degrees/minute. The feed can be programmed in feed/minute instead of degrees/minute, however, the angular velocity and the part radius must be used to calculate circumferential velocity. For the unit circle diameter

$$Do = (1mm)\frac{(360)}{\pi} \quad ; \quad Do = (1in)\frac{(360)}{11.5in}$$

The resultant vectorial tangential velocity at 1°/min equals 1 in/min (1 mm/min).

If a rotating axis is only moving and the stationary tool tip contacts the part surface at a diameter equal to D, then the surface velocity of the tool tip relative to the part surface equals:

$$V_{tool} \text{ inch/min (or mm/min)} = \frac{D}{Do} \cdot V_{programmed} \frac{\text{degrees}}{\text{minutes}}$$

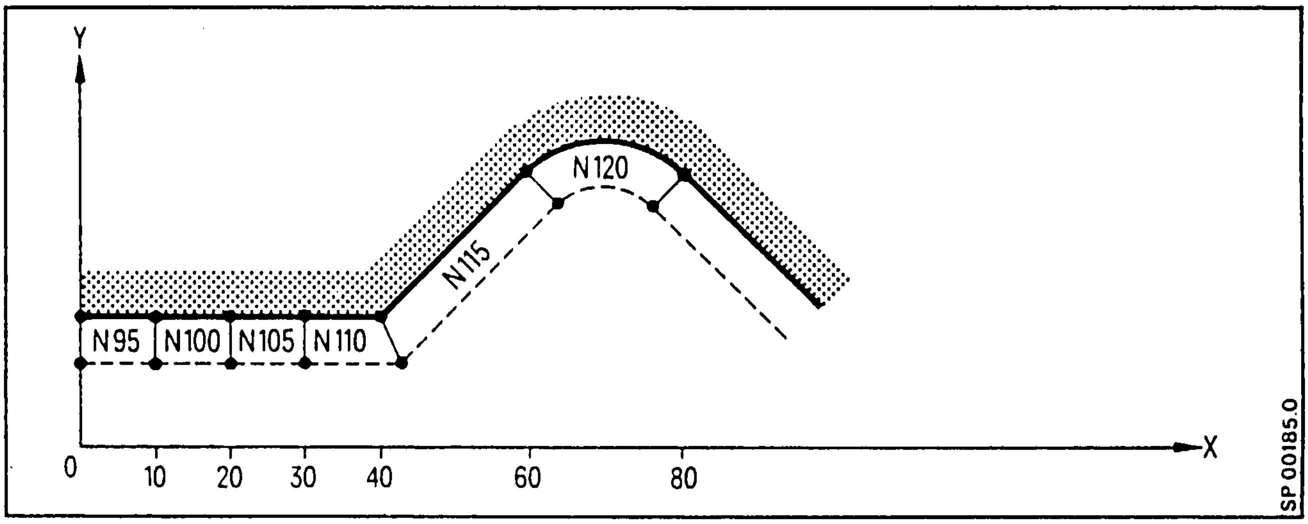
$$V_{programmed} \frac{\text{degrees}}{\text{minutes}} = \frac{Do}{D} \cdot V_{tool} \text{ in inch/min (or mm/min)}$$

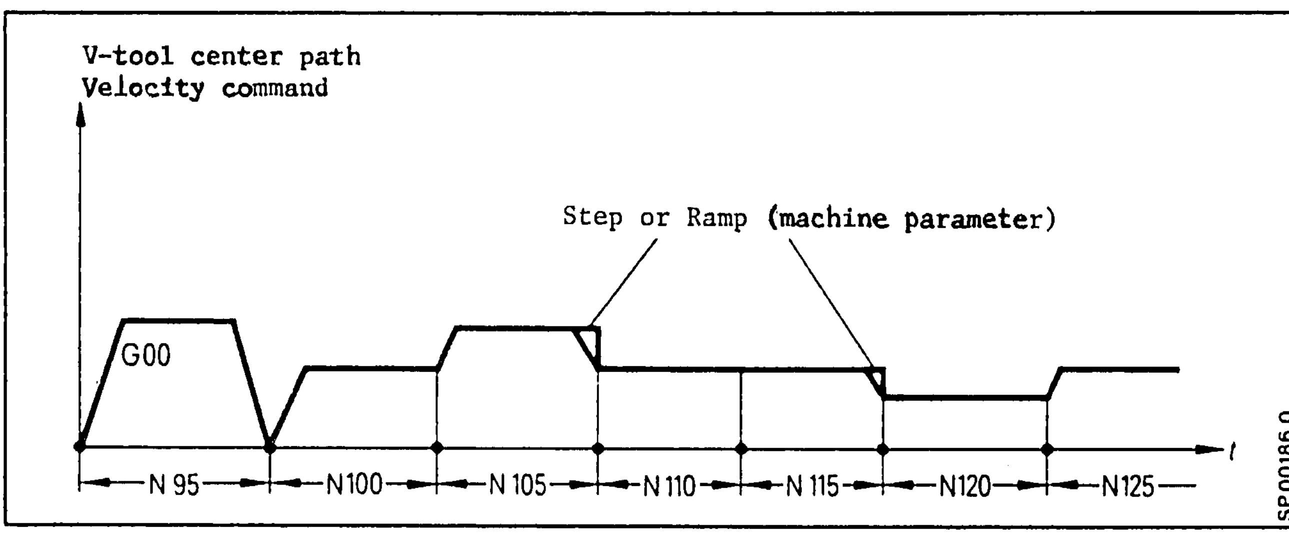
The feed rate override switch located on the operator panel can modify the programmed feed from 1% to 120%. The 100% setting corresponds to the programmed value.

The feedrate programmed under "F" has several meanings dependant upon both G- and M-functions. For this reason refer to the program key on pages 8-29 to 8-34.

Velocity Transitions

```
.
.
N95    G91 G42   G00  X10.
N100             G01  X10.      F2000
N105                  X10.      F3000
N110                  X10.      F2000
N115                  X20.      Y20.
N120             G02  X . . .   Y . . .   I . . .   J . . .
N125             G01  X . . .   Y . . .
```

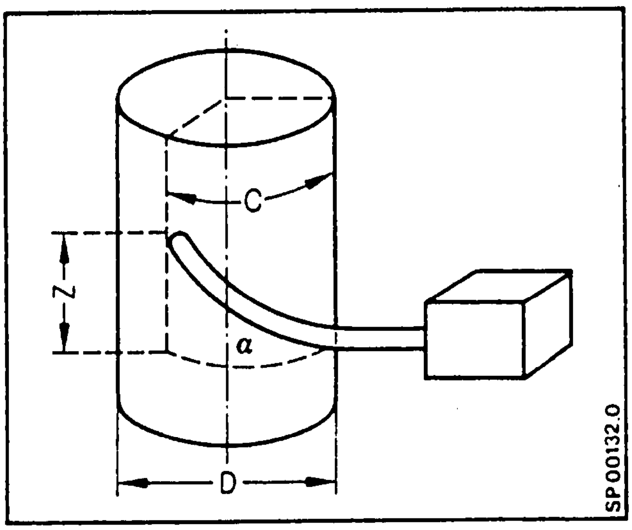




In block N120 a velocity transition or change occurs with respect to the tool center path in relation to the two radii (cutter radius, contour radius).

The following holds true when simultaneously moving a linear and rotary axis:

Whenever the distance between the tool tip contact point and rotary axis remains constant, the magnitude of the surface tangential velocity will also be constant. A constant path velocity also results when linearly interpolating a rotary and linear axis in a path parallel to the axis of rotation (helical cutting on a cylinder). The resultant path velocity of the tool tip relative to the cylinder surface is a function of the programmed velocity, the cylinder diameter on the slope of the helix.



$$V_{tool} = V_{programmed} \sqrt{1 + \frac{D^2 - Do^2}{Do^2} \cos^2 \alpha}$$

$V_{programmed}$ = programmed path velocity in deg./min.
D = helix diameter in inches
Do = unit circle diameter = 11.5 in
α = arc tan $\frac{Z}{C}$ (slope angle of the helix)

Z = programmed departure (in/mm)
C = programmed angle in degrees

If the distance between tool tip, work surface, and the rotary axis is not held constant (eg. spiral in a plane), then the path velocity will not be constant. The path velocity will continuely change as a function of the variable machining diameter.

A constant path velocity can be simulated by splitting the programmed block into several contiguous blocks in which the feed function is changed to approximate the desired velocity. A subroutine program using parameter chaining is a useful technique for velocity approximation.

When interpolating helically, the programmed feed is maintained on the arc path.

The programmed feed can be down rated 1:100 by programming M37. M36 will restore the feed to it's programmed value (default setting).

N5 . . . G94 M36 F10 LF

G94 Feed F in (feed/min) — G94

G95*Feed F in (feed/rev)

G96*Feed F in (feed/rev)
and under address S constant
surface speed in (m/min, in/min)

Programming M36 is only logical if M37 was previously programmed — M36

Feed in in/min, mm/min (or in/rev, mm/rev) — F10

The format for feed per rev and spindle speed programming is shown in section 8.2.

3.18 G96 S.. Constant surface speed (V=constant)*

Typical application: facing attachment
Dependant on the programmed surface speed, the control derives the appropriate spindle speed as a function of the part diameter.

N5 G01 G96 W.. S.. F.. LF

constant surface speed
in m/min

The relationship and interdependancy of the part diameter, spindle speed and the feedrate motion enable an optimum matching of the program to the machine, the material and the tool. The zero point of the W-axis is normally the turning centre. If this is not the case then this difference may be reflected in the zero offset (G54 to G57, G59). For the calculation of the spindle speed with constant surface speed the following variables are taken into account in the control:

- Machine position
- Tool offset
- Zero offset in the W-direction
- Positional shift through G92 W...
- Preset shift

A DRF shift is not taken into account.The displayed position is referred to the radius. In the block in which G96 is selected, the W-axis should be programmed alongside.
The G97 function freezes the constant surf ce speed and the last calculated spindle is stored. G97 is selected in order to avoid undesirable speed fluctuations in intermediate blocks in the W-direction without machining. The constant surface speed is reactivated through programming G96.

Gear changing : With constant surface speed programming, machining is executed at one particular gear range. A gear change is possible at all times at appropriate places in the program.

3.19 G92 S . . . Programmable Maximum Spindle Speed with G96

It may become necessary (e.g. with constant surface speed G96) to limit the spindle speed to a constant maximum value. Prior to the block in which a spindle speed limitation is required, a block is programmed with the limiting value under address S in rpm. The preparatory function G92 S . . . can be reprogrammed throughout the program.
No other characters may be programmed within the G92 block.

N . . G92 S300 LF

No other commands within the same block → G92

Spindle speed limited to 300 rpm → S300

Neither G94 nor G95 will cancel G92 S, it remains in effect throughout the program. An automatic cancellation through G94 or G95 is in preparation.

G92 S . . . is cancelled by programming a new G92 S . . . corresponding to the selected gear range. G92 S 0 reduces the spindle speed to zero.

3.20 G26 S . . . Actual Spindle Speed Monitor

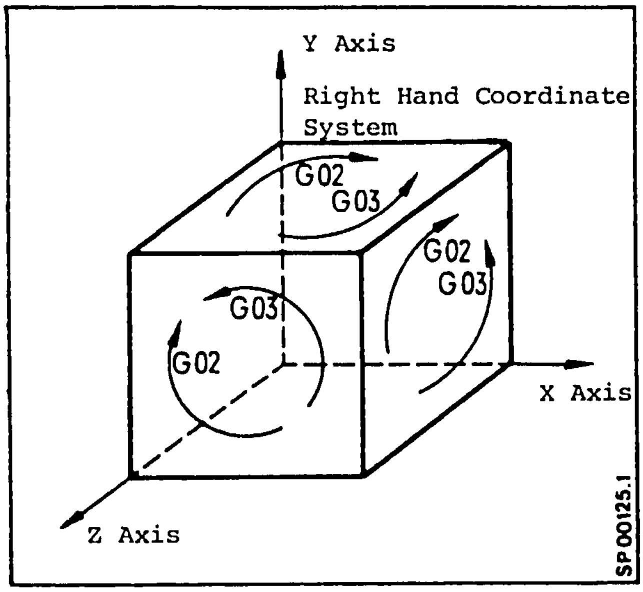
The speed monitor G26 S . . . serves as a tool or chuck dependent maximum safe speed limit. It is independent of the G94 - G97 function. The function is primarily intended to protect the operator.

With constant surface speed G92 S . . . is also in effect.

To input G26 S see operator's manual.

3.21 Machining Plane Selection

(Machining Plane Selection with Sprint 8M - see chapter 7.7)



The plane in which the cutter radius compensation is preformed and also the plane in which circular interpolation is preformed is implicit from the programmed axis words following G02/G03 or following G41/G42 D . . . . To select the plane, two axis words are necessary even though one dimension may be zero (also see section 3.5).

No more than 2 axes may be programmed.

The order of the axes programming is evaluated by the plane selection (also see section 3.5.)

In order to obtain a right hand coordinate system, the axis must be programmed in the following order:

X... Y...
Z... X...
and Y... Z...

3.22 G40/G41/G42 Intersectional (look ahead) Cutter Radius Compensation

(Intersectional (look ahead) Cutter Compensation with Sprint 8M
- see Chapter 7.8)

G40 Cutter compensation off
G41 Tool to the left of the part
G42 Tool to the right of the part

When mirror imaging is used and the sign is considered, the traversed path is as follows:

Both axis are mirrored or Neither axis is morrored			one axis is mirrored	
Sign for the radius compensation value of the cutter				
	+	-	+	-
G41	left	right	right	left
G42	right	left	left	right

G40, G41 and G42 may be programmed in blocks without axis moves. The compensation of the cutter radius is active in the plane of the two programmed axes. Length compensation may be selected for any axis with G43/G44.

Selecting and cancelling the intersectional (look ahead) cutter Radius compensation

The selection is only possible, when G00 or G01 are active. Two axes have to be programmed when selecting. With this the plane is selected once. Afterwards max. 5 axes can be programmed. The intersectional cutter compensation is only effective in the selected plane.

N10 G01 G41 D07 X... Y... LF	At the end of this block, the compensated path is reached. The plane is fixed through X and Y.
N25 D00 X... LF	Cancellation of the CRC

or

N25 G40 X... LF	Cancellation of the CRC and length compensation in X

With G40, the compensations G41/G42 are cancelled. However, at least one axis' motion must be programmed in order to restore the tool to its uncompensated path.
Length and radius compensation can both be cancelled, when D00 and the respective axis are programmed.
Exception: without any previous selection, G41, G42 D00 may only be programmed with both axes.

Switching from G41 to G42

```
N10 G01 G41 D12 X... Y... LF
N15     G42     X...      LF
N20             X... Y... LF
```

Calling a different tool offset function

The G-functions (G41/G42) must be reprogrammed.

```
N10 G01 G41 D12 X... Y... LF
N15     G41 D10 X...      LF
N20             X... Y... LF
```

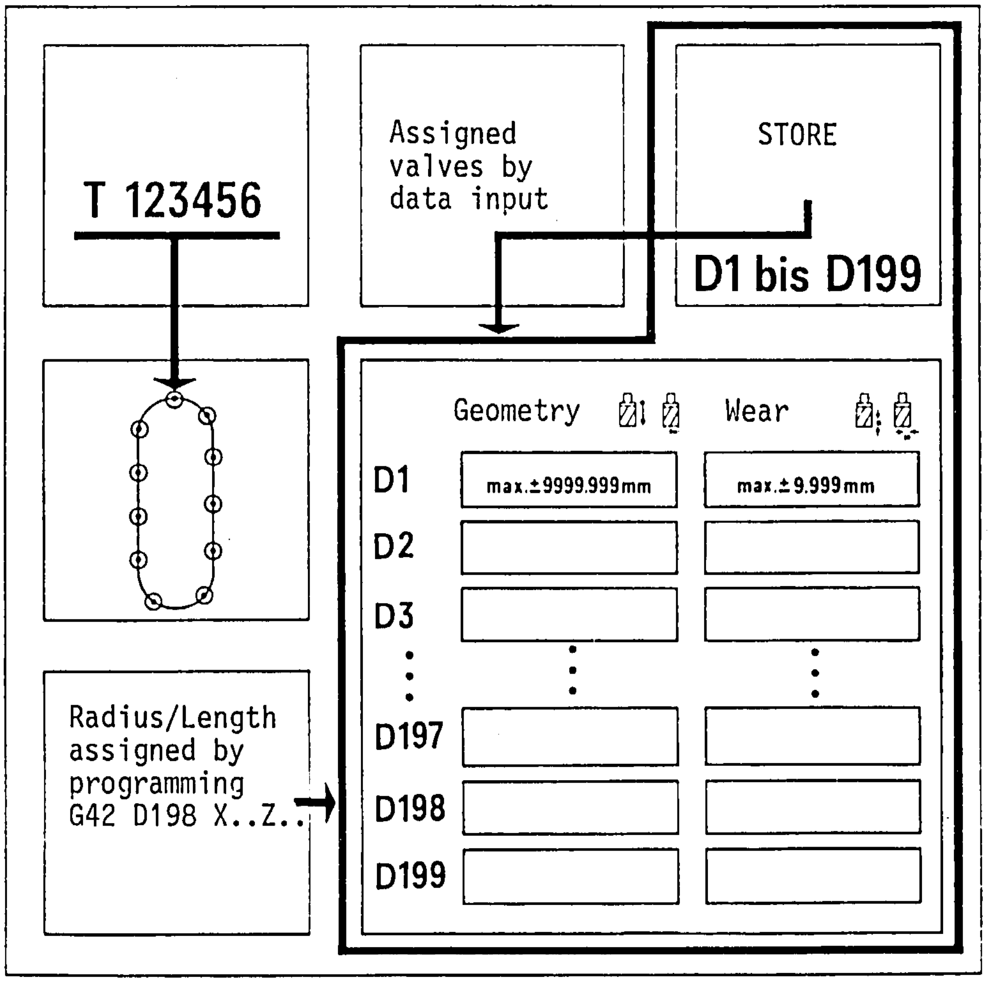
3.23 Tool Offset

(Tool offset with Sprint 8M - see chapter 7.9)

The tool data are stored under a tool offset number.

Wear compensation	Length	± 9.999 mm
	Radius	± 9.999 mm
Tool geometry	Length	± 9999.999 mm
	Radius	± 999.999 mm

A total of 199*offsets is available.



Under the tool offset number the length or radius dimensions are stored. The wear compensation values are input via the operator's panel. They are stored according to the offset number (designator).

Tool offset call and input (geometry)

A tool offset is called via a two digit designator
D01 ... D199 (length or radius)

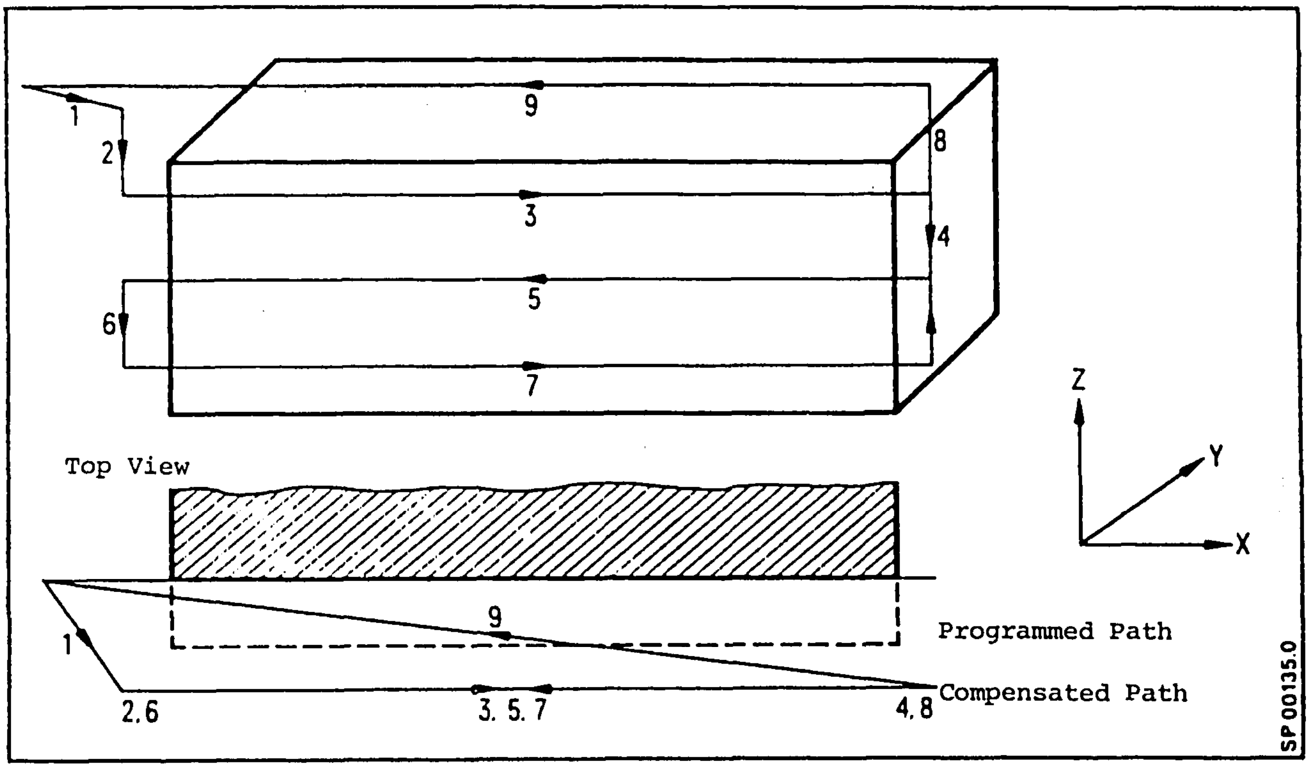
- Input via tape

```
% TO LF
G92 D01 D... LF
.
.
G92 D199 D... LF
M02 or M30 LF
```

- Input via program

```
N11 ... LF
G92 D01 D... LF
.
.
G92 D199 D... LF
N12 ... LF
```

Example: Straight Milling



N1	G00	G42	D13	G91	X15.	Y0.		LF	Intersectional cutter radius compensation and plane selection
N2	G01	G43	D17	F100			Z-15.	LF	Advance to the 1st milling depth
N3					X105.			LF	Approach and mill
N4							Z-15.	LF	Advance to the 2nd milling depth
N5	G41				X-105.			LF	Approach and mill
N6							Z-15.	LF	Advance to the 3rd milling depth
N7	G42				X105.			LF	Approach and mill
N8							Z45.	LF	Pull tool out of the work
N9	G40				X-120.			LF	Cancel all compensations and offsets.

3.24 G40/G43/G44 Tool Length Offset and Axis Parallel Cutter Radius Compensation

(Function not possible with Sprint 8M)

G40	Cancel tool offset	Axes are parallel arcs may be contained within the contour from software stand 01
G43	Positive tool length offset	
G44	Negative tool length offset	

With the help of a tool length offset, the difference between actual tool dimension and the programmed assumed dimension can be compensated . The preparatory functions G43 and G44 inform the control in which direction the offset must be made.

If an offset in an axis is desired, the preparatory function and the D tool offset word must precede the dimension word in the program block.

N5 G43	D17	X...		Y...	LF	G43 D17 refer to the X-axis, the offset is in effect for succeeding blocks.
N10 G44	D18			Y...	LF	G44 D18 refer to Y, the X value determined by G43 D17 remains in effect.
N15 G44	X...		G43	Y...	LF	G44 D17 offsets the X dimension G43 D18 offsets the Y dimension
N20 D15	X...		D00	Y...	LF	G44 D15 offsets the X dimension The offset is cancelled in Y
N25 G40					LF	All offsets are cancelled (the compensation distance is not traversed)
N30 G40	X...			Y...	LF	All offsets are cancelled and the offset distance together with the programmed dimension is traversed. The end position is the uncompensated programmed position.

The preparatory functions and the offset value are modal and are effective for the axis in which the offset was programmed. The offset can be modified by a new G43/G44 and a new tool offset word when properly formatted.

Sign Convention

A positive dimension is input, when the actual dimension of the tool is greater than the programmed value has taken into account. A minus dimension is input, when the actual dimension of the tool is smaller than the programmed value has taken into account.

For example:

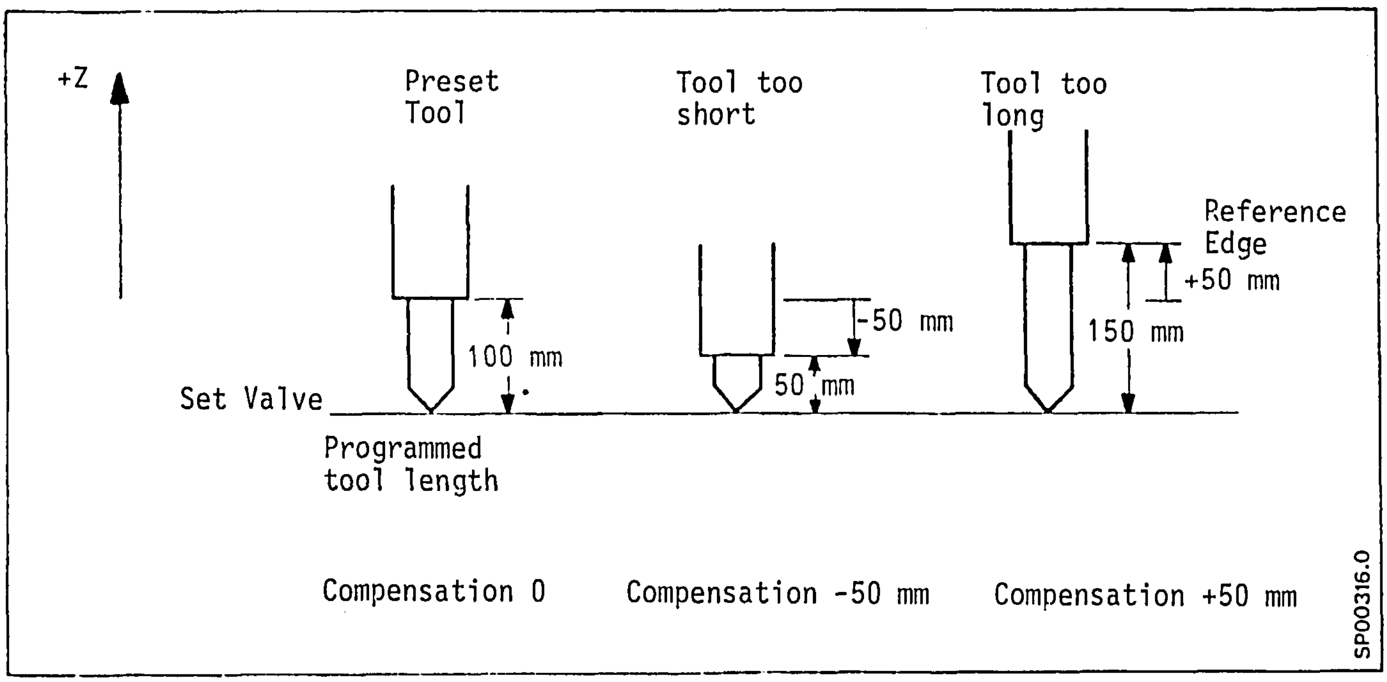
The actual drill length is langer than the programmed drill length	+ offset
The actual cutter has a smaller radius than the programmed radius	- offset

G43 positive tool length offset

The offset called by the D word is calculated with its sign to the associated axis.

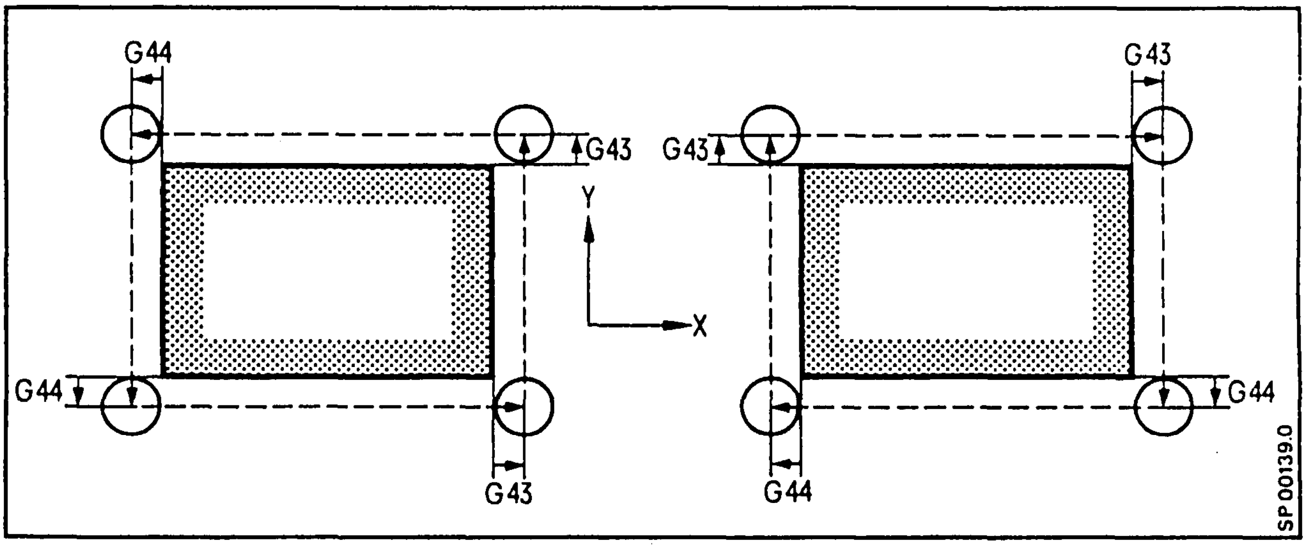
G44 negative tool length offset

The offset called by the D word is calculated with its sign to the associated axis.



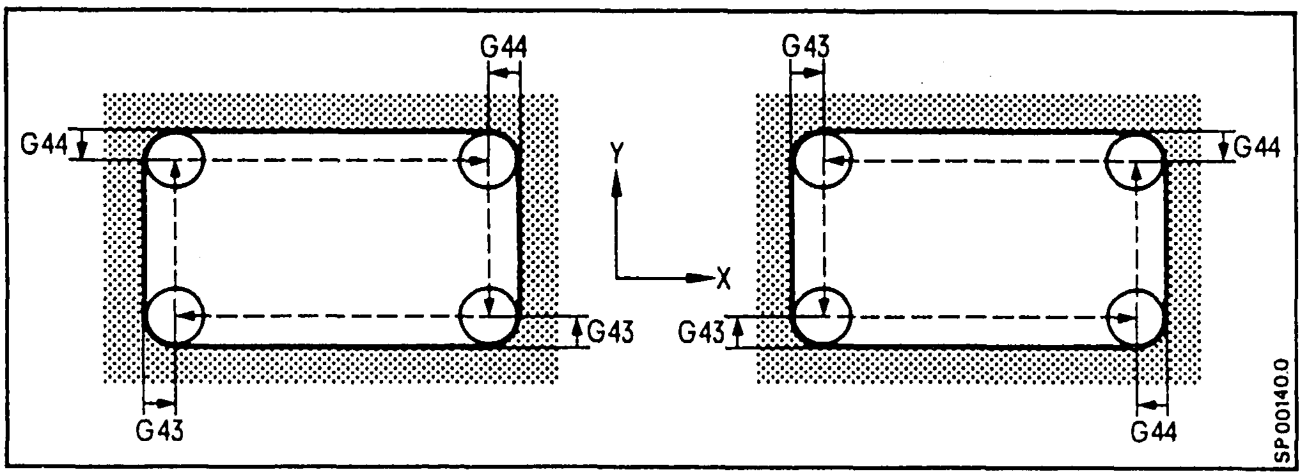
External Machining Operation

Positive axial motion G43
Negative axial motion G44



Internal Machining Operation

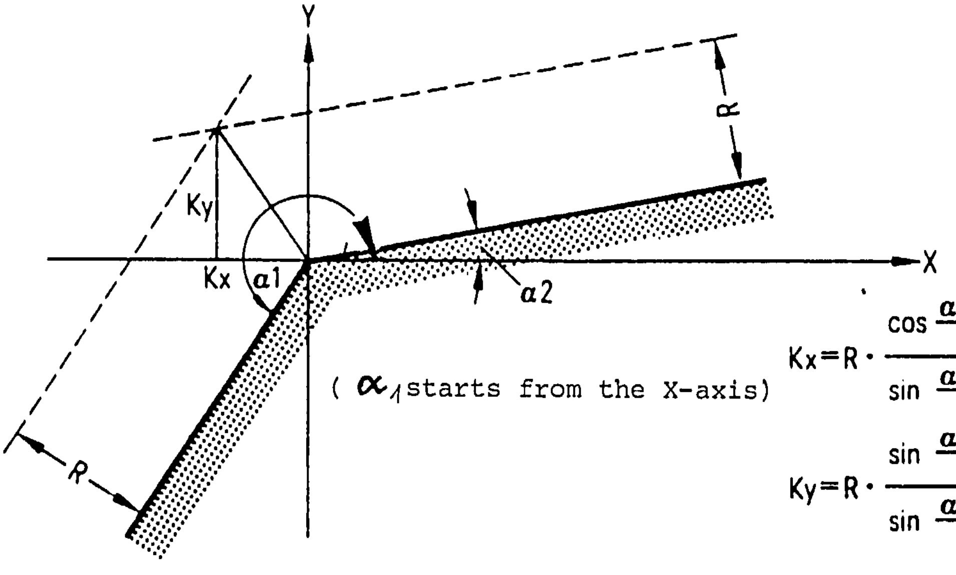
Positive axial motion G44
Negative axial motion G43



3.25 G43/G44 Tool Length Compensation on an Incline

(Function not possible with Sprint 8M)

In addition to parallel axis milling with offsets an inclined plane can also be milled using tool length offsets. The operator, however, must calculate the axial offset for each axis associated with the incline. The values are calculated using the trigonometric relations that define a line and slope.



(α_1 starts from the X-axis)

$$K_x = R \cdot \frac{\cos\frac{a1 + a2}{2}}{\sin\frac{a1 - a2}{2}}$$

$$K_y = R \cdot \frac{\sin\frac{a1 + a2}{2}}{\sin\frac{a1 - a2}{2}}$$

SP00141.0

The above equations are valid when the cutter is located to the left of the part. If the cutter is situated to the right of the part, then the signs of the equations must be reversed.

Parts which contain several inclines of varying slope will have offsets that vary accordingly. For these operations intersectional cutter radius compensation (G41/G42) is a more advantageous programming technique.

See example on page 3-58 , blocks N2 and N3.

3.25.1 G43/G44 Tool length compensation with arcs with tangential transitions

In addition to the inclined compensation, G43/G44 may also be used to compensate for full arc quadrants, multiples thereof (from one quadrant to another) and arc sections (with tangential block transitions). Compensation may be deselected/selected by the use of an intermediate block (without movement on the programmed contour); see the example on the next page; blocks N4,N5,N6 and N7.(Blocks N4 and N8 may also be inclined).

Attention:
The compensations must only be defined for the path information and not for the interpolation parameters. The parameters are automatically compensated for from the path information compensation. With blocks with G43/G44 without G41/G42 cutter radius compensation, the G43/G44 acts as a simple cutter radius compensation by adding the length compensation to the interpolation parameters in the case of arcs. Furthermore the length compensation is taken into account with mirror image.

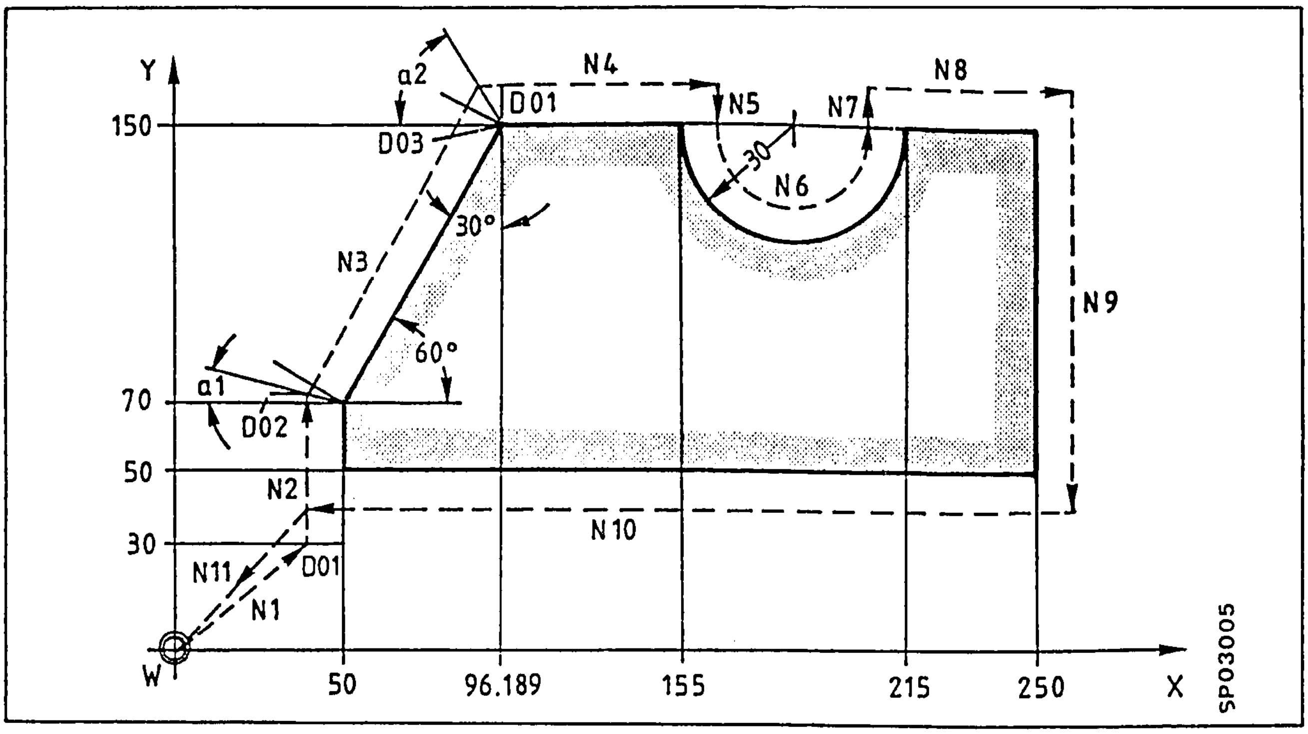
There are clearly two resulting cases:

1. The centre point of a programmed arc is not shifted by the length compensation -
   - Program length compensation with G43/G44 do not program cutter radius compensation

2. The centre point of a programmed arc is shifted by the length compensation (if CRC and LC are programmed) -
   - Program length compensation with G43/G44 program cutter radius compensation with G41/G42

Attention: This function does not work in conjunction with "Cylindrical interpolation function(B73)".

An example using axis parallel tool offsets



——————— part contour

- - - - - - - - offset cutter centre path

Programming:

```
N 1 G00 G64 G90 G44 D01 X 50000            Y 30000                 LF
N 2 G01                              G43 D02 Y 70000 F200   S56 M03 LF
N 3                 D03 X 96189          D01 Y150000                 LF
N 4             G43 D01 X155000                                      LF
N 5                                      D00 Y150000                 LF
N 6 G03         G44     X215000              Y150000 I+30000 J0      LF
N 7 G01                                  D01 Y150000                 LF
N 8             G43     X250000                                      LF
N 9                                  G44     Y 50000                 LF
N10             G44     X 50000                                      LF
N11 G00     G40         X0                   Y0                      LF
```

Description:

The part contour is programmed. The cutter radius is 10.000 mm.
The operator must enter the following offsets:

D01 : 10 000

D02 : 2 678 (10 • tan α_1; α_1 = 15° = half angle)

D03 : 5 774 (10 • cot α_2; α_2 = 60° = half angle)

3.26 Cutter Radius and Tool Length Offsets Used Together In A Program

For Sprint 8M see section 7.8.1

The tool offsets stored under D01 thru D199 are used in conjunction with G43 and G44. The length offset (LO) is programmed individually for each axis and is independent of the plane.

It is possible to use CRC and LO Together. The first two axes programmed select the CRC plane.

N10 G41 D10 X.. Y.. G43 D11 Z.. LF   X-Y is the CRC plane
Length offset is active in Z

Where logical CRC and LO may be programmed in the same block e.g. a facing head or with an angular milling head.

N12 G41 D10 G43 D11 X.. Y.. Z.. LF

X-Y is the CRC plane

Length offset is active in X

3.27. G36/G47 Coordinate transformation "TRANSMIT"

With 8M/Sprint 8M from SW02, from SW03 also with 8MC, (activation through machine data).

For milling of turned parts on rotary tables when the desired contour should be obtained by interpolation of the rotary axis with one linear axis (application for special purpose machines). The coordinate transformation function enables programming a fictional cartesian coordinate system whilst the machine motion is in reality in polar coordinates.

The fictional cartesian coordinate system is constructed from the first axis, X and the corresponding rotary axis which is defined in machine parameter through machine data 465 bits 0-3. Thereafter the rotary axis is labelled the "C" axis and the fictional axis, "Cf".

Selection and cancelling the transformation is through G-functions in the program.

G36 Coordinate transformation cancelled
(reset state)

Programming is as normal in the polar coordinate system (the machine coordinate system).
Rotary axis C in degrees, speed in degrees/min.

G37 Coordinate transformation selected

Programming is in the fictional cartesian coordinate system. C is in mm, the speed is given in mm/min.

The peculiarities concerning programming resulting from the turning/milling operation by software switching 8T/Sprint 8T after Sprint 8M are detailed in the printed matter "SINUMERIK 8T/Sprint 8T, turning/milling operation".

Programming with Sprint 8M

With G17/G18/G19 the missing axis of the current fictional plane is modified if only one of the two axes forming the fictional plane is programmed in a block. (examples a and b).

If one of the fictional axis pair is programmed along with another axis not belonging to this pair,then the control generates alarm 504 (example c).

If axes must be programmed together with the fictional plane axes together in a block,then both the fictional pair must be programmed (example d).

Example: G37 active, X-Cf = fictional plane = G17 - plane

a) G17 X.. LF = Cf is modified
b) G17 Cf.. LF = X is modified
c) G17 X.. Z..LF= alarm 504
d) G17 X.. Cf.. Z.. LF = no axis modification - no alarm
e) G17 Z.. LF = " " " " "

Programming with 8M

Both axes of the fictional plane must always be programmed.
Example: G37 active, X-Cf = fictional plane

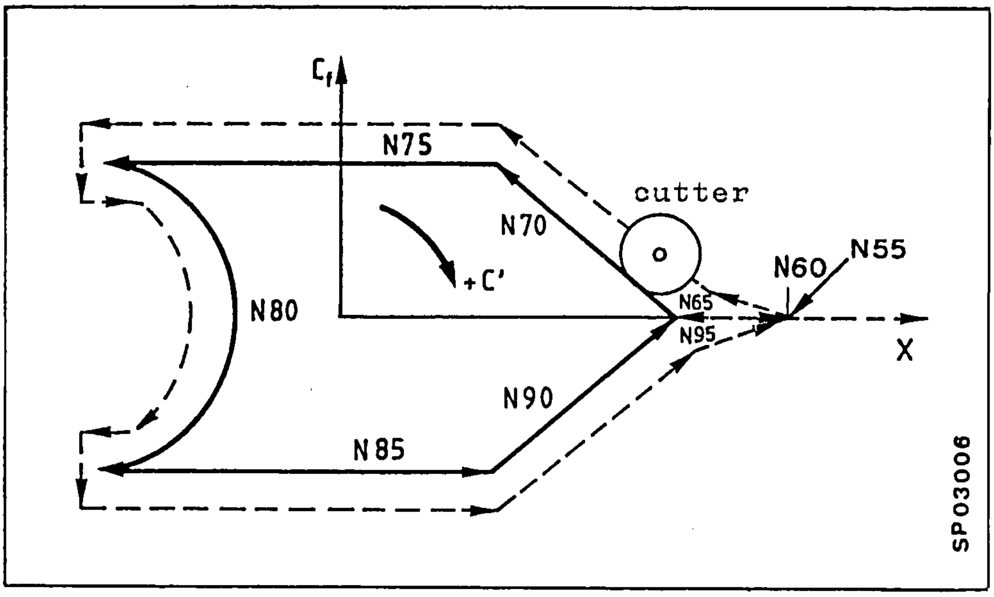
a) X.. LF = not allowed
b) Cf..LF = not allowed
c) X..Z.. LF = not allowed
d) X..Cf.. LF = allowed
e) X..Cf..Z.. LF = allowed
f) Z.. LF = allowed

Notes:

- The workpiece zero point lies in the middle of the facing axis.
- Rapid traverse movement must be programmed using G01 or G11 with the appropriate F value.
- Any shift in Cf direction may not be compensated for in the control.
- When changing from G36 to G37, the actual value of the C axis is set to zero and the actual value of the X axis is set to the machine actual value independently of existing offsets (zero offset, preset, G92). The zero offsets are accounted for in cartesian coordinate values.
- The continuous path velocity is programmed and kept in the X-CF coordinate system. In a circle around the middle of the facing axis only those X-CF velocities are permitted which result in permissible C axes rotation speeds. If this limit is exceeded, the movement is stopped.
- At selected cutter radius compensation G41/G42, the transformation may not be switched on or off.
  (Change of G36/G37)
- Block advance via G37 blocks is not permitted.
- Within a contouring cycle train, the transformation may not be switched off or on. (Change of G36/G37)
- Conversion of inch/metric in the X-CF system is not permitted. All dimensions must be programmed in metric.
- The accuracies achievable on the part when using the C axis are dependent upon the instantaneous working radius.
  (Control in degree)

The peculiarities that apply to turning/milling through the 8T/Sprint 8T to Sprint 8M software conversion are more fully detailed in: "SINUMERIK 8T/Sprint 8T, turning/milling operation".

Program example: Milling of a "face contour" with TRANSMIT



%1234	
N5 X.. C.. Z..	
*	
*	real coordinate system
N50 X.. C.. Z..	
N55 G0 x 120 C0 Z100 D50	approach start point select length compensation
N60 G37 G01 F200 Z90	Transformation block with G37, programming switches over to fictional coordinate system
N65 G42 X90 C0	Selection of the cutter radius compensation (C is equivalent to the fictional axis Cf)
N70 X40 C40	
N75 X-60	
N80 G02 C-40 J-40	Half circle
N85 G01 X40	
N90 X90 C0	
N95 G40 X120	Deselect the cutter radius compensation
N100 Z100	Retract in Z
N105 G36	Switch off transformation
N110 X.. C.. Z..	
*	real coordinate system
*	

4. Miscellaneous and Auxillary Functions M, T, S, H

The miscellaneous and auxillary functions are output when the program block is executed. A maximum of three M, one S, one T, and one H function may be programmed in one block. The functions are output to the interface in the following sequence:

- All functions are output simultaneously except when a 2nd and or 3rd M function is programmed.
- 2nd M-function is output
- 3rd M-function is output

A machine parameter is used to define whether the function is output before or while the programmed axis is in motion. See the machine tool builders manual.

If the functions are output while the axis is in motion, the following will hold true:

If a new function is to be in effect while an axis is in motion, then the function must be programmed in the preceeding block.

4.1 S-Function

The S-function is selected to specify:

Spindle speed as a coded number
Spindle speed in rpm or 0.1 rpm
(defined by machine parameter)
constant surface speed in: in/min (m/min)
or
0.1 in/min (0.1 m/min)
(defined by machine parameter)

For further definitions refer to the program key; pages 8-29 to 8-34.

the same input format used for feedrate programming is also used for constant surface speed programming.

4.2 Auxillary Function H

A switch function or auxillary function of the machine not under NC control can be programmed as an H-function word. One auxillary function (H-function) can be programmed per block under address H. The H address has a maximum field width of six digits. For the H-function definition, see the machine tool builders manual.

4.3 Tool Function T

(Tool Function with Sprint 8M - see chapter 7.10)

The tool function (T-function) designates the tool necessary for a machining operation.

T 123456

Address for the tool function

Tool designation (position)
6 decades

4.4 Miscellaneous Function M

M00 Programmed Stop (unconditional)

M00 enables an executing program to stop. An operation may be performed and when completed, program execution can commence by pressing the "cycle start" key. Stored information is not affected. The miscellaneous function M00 functions in all automatic modes. Whether or not the spindle is stopped depends on the machine tool builder and is specified in the machine programming manual.

M00 is effective in blocks programmed with or without axis dimension words. M0 and M are recognized as an M00.

M01 Optional Stop (conditional)

M01 functions similar to M00, however, the optional stop key must be activated in order to enable an M01. M00 and M01 function in the same manner as the "single" block mode.

M1 is recognized as an M01.

M02 End of program

M02 is programmed in the last program block. An M02 will reset the control to the first program block. The control will revert to its default state (see the program key). M02 may be programmed alone or together with other functions in a block. M2 is recognized as an M02.

M17 End of Subroutine

M17 may be programmed alone or together in a block. M17 signals a subroutine return to the calling program. A subroutine call and M17 may not be programmed in the same block.

M30 End of program with rewind

M30 acts like M02, except that in automatic mode from tape reader it initiates tape rewind to rewind stop "%" (only with reader with reels).

M03, M04, M05, M19 Main spindle control

If the NC is equipped with analogue spindle speed output (option), certain M words are used for spindle control:

M03 Direction of spindle rotation clockwise
M04 Direction of spindle rotation counter-clockwise
M05 Spindle stop
M19 Oriented spindle stop (only with encoder).

Using M19 S it is possible to stop the spindle in a pre-defined position. The angle is programmed using S in degrees (distance from the marker pulse in the M03 direction). The angle programmed using address S is modal. When M19 is programmed without S the stored value becomes effective for the angle. A block containing M19 is only finished when the signal "Spindle Stop" is received from the interface. M3, M4, M5 may also be written. M19 or M19 S... must be programmed in its own in a separate block. The spindle positioning occurs in parallel to axis movement, independent to block boundaries, even from a stop state (SW02). From SW03, M19 is possible from the stop state after switching on without previous spindle rotation.

M36, M37 Decreasing the feedrate

The feedrate programmed under F in mm/min or mm/rev can be reduced by the ratio 1:100 using a further function.

- M36 Feedrate remains as programmed under F
- M37 Feedrate is reduced by a ratio 1:100

Unassigned miscellaneous functions

All miscellaneous functions except M00, M01, M02, M03, M04, M05, M17, M19, M30, M36, and M37 are unassigned. Exact information regarding the application of the individual functions is given in the program key specific to the machine. A partial definition of this function is given in DIN 66025.

5. Parameter

Parameters R00 to R49 may be assigned to all addresses with the only exceptions of N and @ , throughout the part programs and subroutines.
A parameter is set equal to a numerical value in the part program or in the subroutine. The R-parameter dimension takes on the characteristics of the address under which it is programmed. A maximum of 10 parameters are allowed in a program block.

Example:

```
L51000          LF        Parameters R01, R05 and R49 are
N1  Y-R49  SR05 LF        used in subroutine.
N2  X300.  -R01 LF
.
.
N50 M17         LF

%  4081         LF
N1 . . .        LF
.
.
N37  R01  10.  R49-20.05  R05  500  LF

N38  L51002  LF           Subroutine L510 is called, it will
                          run twice
                          R01  =  10.
                          R05  =  500
                          R49  =  -20.05
```

5.1 Parameter Definition

During parameter definition, individual R-parameters are set equal to signed numerical values. The parameters are assigned in part programs or subroutines. Up to 10 parameter definitions may be programmed in one block. A total number of 120 characters may thereby not be exceeded.

Program Statement	Operation	Result
R01 1.078	R01 +1.078	R01 = +1.078
R02 9.534	R02 +9.534	R02 = +9.534
R03 -55.51	R03 -55.51	R03 = -55.51

5.2 Assigning Parameters in a Program

Direct Assignment (except address N). An address is given the value defined by the R-parameter.

Program Statement	Operation	Result
FR01	FR01	F = +1.078
YR03	YR03	Y = -55.51
Y-R03	XR03	X = +55.51

Arithmetic Assignment

To the numerical value of an address, the control performs a signed addition or subtraction with the parameter value.

Program Statement	Operation	Result
X 20.78 -R01	X = 20.78 -10.78	X = 10
Z 44.9 -R03	Z = 44.9 -(-55.61)	Z = 100.41
F 10.1 R02	F = +10.1 +9.534	F = 19.634

The sequence address, numerical value, parameter must be maintained.

An unsigned parameter or number is assumed positive. (+)

5.3 Parametric Operations

Function	Program Statement	Operation	Result is stored in
Addition	R01 R02	R01 + R02	R01
Subtraction	R01 - R02	R01 - R02	R01
Multiplication	R01 . R02	R01 . R02	R01
Division	R01/R02	R01 : R02	R01
Square Root	@ 10 R01	$\sqrt{R01}$	R01
Sine	@ 15 R01	Sin (R01)	R01
Parameter Definition and Addition	R01 10 R02	R01 = 10 R01 + R02	R01
Parameter Definition and Subtraction	R01-10-R02	R01 = -10 R01 - R02	R01
Arctan *	@ 18 R01	$\arctan\frac{R01}{R02}$	R01

Only R-parameters may be multiplied or divided with one another; i.e. a parameter and a number may not be multiplied or divided together. The decimal point defines the operation as multiplication. The block skip character "/" defines the operation as division. The sequence determines the order in which the expression is evaluated.

The argument of the sine is an angle whose value is limited to $\pm$ 360°. The control calculating time is approx. 10 m/sec per operation. Only one operation per block may be programmed.

Range: (1×10^{-8}) to $(2^{27} - 1)$ = 134 217 728

Display: floating decimal point (+.8) to +8.).

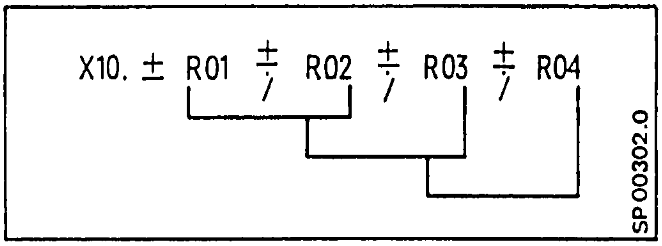
(key zero offset from N100)

For examples, see "Freely programmable cycles"

* Only from software stand 02

5.4 Parameter Chaining

Through parameter chaining a parameter value is altered continually as it loops through a section of program or a subroutine. A calculation is preformed whenever chaining parameters are encountered in a running program.
The last parameter of the chain remains unchanged.
A maximum of 4 parameters may be chained.



Conventions for Evaluating an Expression

A new parameter is calculated from the chaining of two parameters and the sign between them.

An example of 2 parameter chaining

R01 + R02	$R01_{new}$	=	R01 + R02
-R01 + R02	$R01_{new}$	=	R01 + R02
R01 - R02	$R01_{new}$	=	R01 - R02
-R01 - R02	$R01_{new}$	=	R01 - R02
R01 . R02	$R01_{new}$	=	R01 . R02
R01 / R02	$R01_{new}$	=	R01 / R02

An example of 4 parameter chaining

-R01 + R02.R03-R04

$$R01_{new} = R01 + R02$$
$$R02_{new} = R02 \times R03$$
$$R03_{new} = R03 - R04$$
$$R04_{new} = R04$$

The parameters as well as the parameter value may be a signed number.

Example:

%9534LF

```
N1 L0105 R01-10.R02 81. R03 3. LF
N6 L0206 R04-1. R05  4. R06 -1. LF
.
N100 M30 LF
```

A subroutine call to loop 5 times and 6 times respectively. Parameters are defined prior to subroutine entry as:
R01 = -10. R02 = 81. R03 = 3.
R04 = -1. R05 = 4. R06 = -1.

```
L00100
N5 X 1000. -R01 +R02/R03 LF
N10 M17 LF
L00200
N1 Y100. +R04. R05 + R06 LF
N20 M17 LF
```

Parameter used in called subroutines

The following numerical values are taken by the motion axes and the parameters.

L00100

Subroutine repetition /pass	Operation Address and Parameter Definition	for 1st pass	X	-R01 +	R02 /	R03
			1000.	-10.	81.	3.
after1	value for	2nd pass	1010.	71	27.	3.
after2	value for	3rd pass	929.	98.	9.	3.
after3	value for	4th pass	902.	107.	3.	3.
after4	value for	5th pass	893.	110.	1.	3.
after5	value for	6th pass	890.	111.	0.333	3.

L00200

Subroutine repetition /pass	Operation Address and Parameter Definition	for 1st pass	Y +	R04 .	R05 +	R06
			100.	-1.	4.	-1.
after1	value for	2nd pass	99.	-4.	3.	-1.
after2	value for	3rd pass	96.	-12.	2.	-1.
after3	value for	4th pass	88.	-24.	1.	-1.
after4	value for	5th pass	76.	-24.	0.	-1.
after5	value for	6th pass	76.	0.	-1.	-1.
after6	value for	7th pass		0.	-2.	-1.

At the end of a program, the parameters take on the values defined to them by the last performed parameter manipulation. This value remains stored until the parameter is redefined or a parameter manipulation is done resulting in a new value.

5.5 R-Parameter Assignment Under Address "L"

An R-Parameter can be used to define a subroutine number under address L, a looping value under address L or both.

Example:

S.R. Call	S.R. Number	Looping Index Value
N13 L123	123	1
L12 . . . R010 N13 L123 R01	123	0
L12 . . . R0199 N13 L123 R01	123	99
N12 . . . R01 150 N13 L123 R01	124	50
N12 . . . R01 12365 * N13 LR01	123	65
N12 R01 1236 * N13 LR01	12	36
N12 R01 1312.36 N13 LR01	13	12

* Note: In this case the R-Parameter value must be 4 or 5 digits. (LR01)

That is a number of repetitions of 1 must still be programmed.

5.6 An Example of a Subroutine Using Parameters

Example: A Rectangle

The following subroutine illustrates the machining of a rectangle whose sides vary dimensionally. The rectangle sides are assumed parallel to the machine axes.

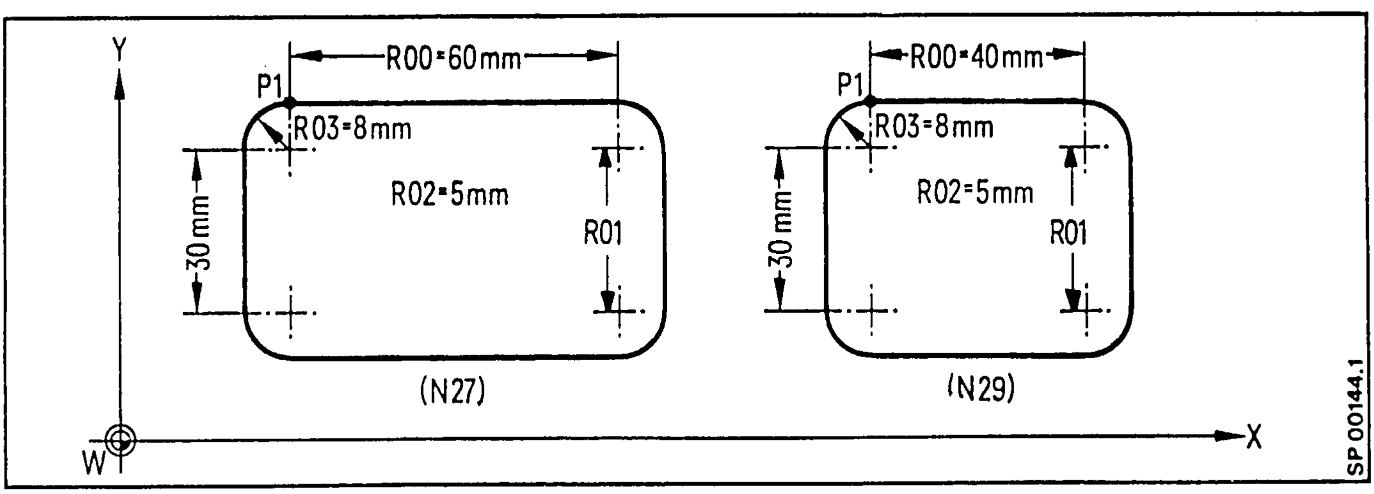
Subroutine

```
L4600                                        LF
N5   G01  G91  Z-R02                         LF
N10            X R00                         LF
N15  G02       X R03  Y-R03  I0   J-R03      LF
N20  G01              Y-R01                  LF
N25  G02       X-R03  Y-R03  I-R03  J0       LF
N30  G01       X-R00                         LF
N35  G02       X-R03  Y R03  I0   J R03      LF
N40  G01              Y R01                  LF
N45  G02       X R03  Y R03  I R03  J0       LF
N50  G01       Z R02                         LF
N55  M17                                     LF
```

Subroutine call:

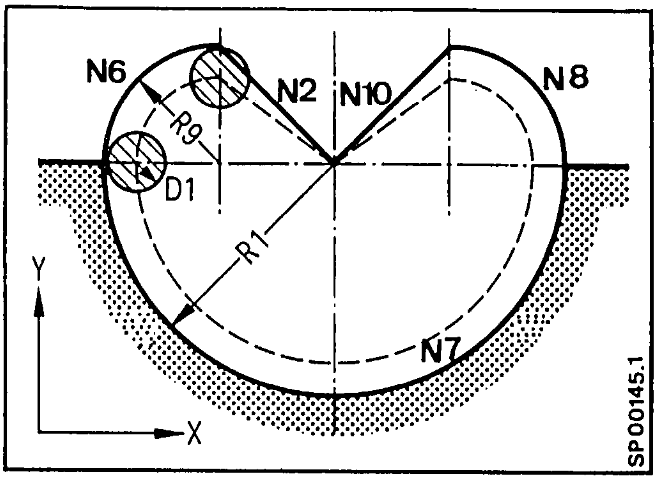
```
N26  G90  X . . .  Y . . .                LF
N27  L46  R0060  R0130.  R025  R038       LF     1st subroutine entry
N28  G90  X . . .  Y . . .                LF
N29  L46  R0040.                          LF     2nd subroutine entry
```

R02 = tool advance in the Z axis direction
P1 = rectangle start and end point of the subroutine



Example: Machining an Internal Semi-Circle Surface

The following subroutine illustrates a stock removal and finishing operation for a semi-circle. The contour radius and the cutter entry radius are written as parameter variables. Each time the subroutine is run, the actual part dimension can be compared to the programmed dimension by enabling the optional stop button (M01). The resulting difference can be stored as an additive tool wear offset.



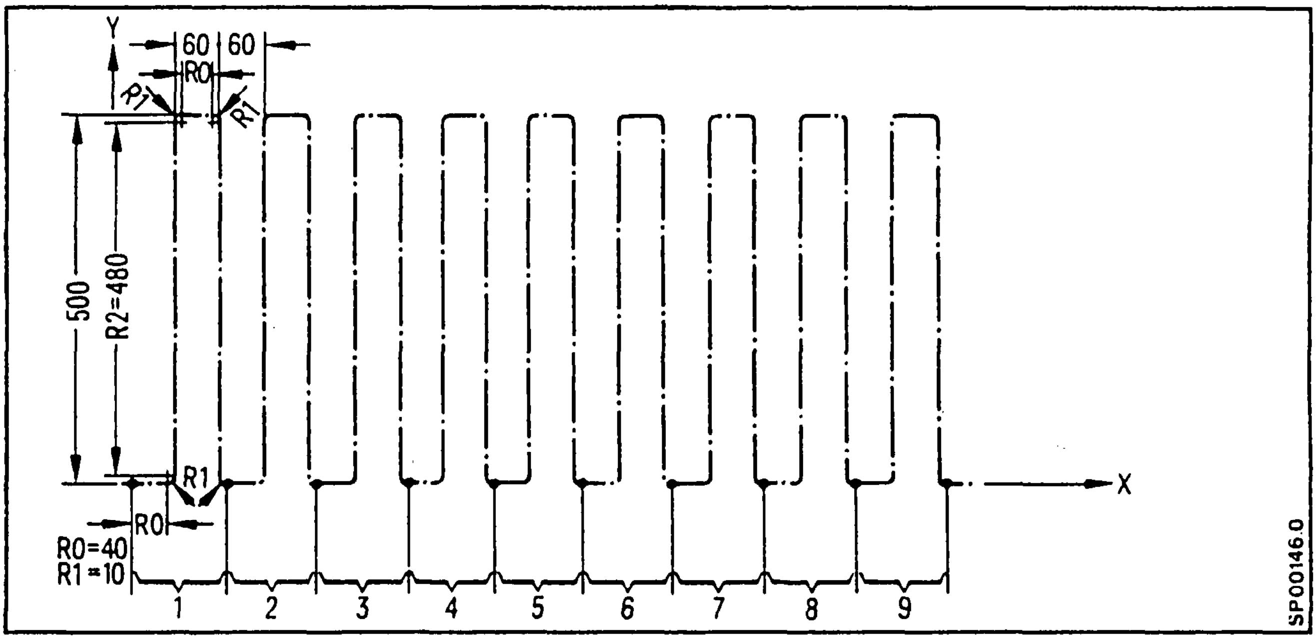

```
%SP
L00100
N1 R02 0 R01
N2 R02 R01
N3 R01 -R09
N4 G00 G64 G91 G41 D R08 X -R01 Y R09
N5 R01 R09
N6 G03 X -R09 Y -R09 P R09
N7 X R02 YO P R01
N8 X-R09 Y R09 P R09
N9 R01 -R09
N10 G00 G40 X -R01 Y -R09
N11 R01 R09
N12 M17
M02
```

Subroutine call

```
% 5873                                          LF
N1 G00 G90 X... Y...                            LF (semi-circle centre)
N2 L0101 R01 30. R08 1. R09 15 F1000 S1000 M03  LF
N3 ...                                          LF
.
```

Example: Straight Milling

The path transitions are programmed with radii to avoid a reduction in the feed rate. In this manner dwell marks are avoided during a path direction change.



L34 is Called by another Program:

```
•
N15 L3409 R00 40. R01 10. R02 480. F200 LF
•
```

Subroutine:

```
L3400
N1 G01 G64 G91 XR00                   LF
N2 G03          XR01 YR01 I0. JR01    LF
N3 G01               YR02             LF
N4 G02          XR01 YR01 IR01 J0.    LF
N5 G01          XR00                  LF
N6 G02          XR01 Y-R01 I0 J-R01   LF
N7 G01               Y-R02            LF
N8 G03          XR01 Y-R01 IR01 J0.   LF
N9 M17                                LF
```

5.7 Buffer store empty, L999; for intended influences on the program (further read in of NC-blocks is inhibited)

A series of influences through the control or from the interface control (parallel interface or PC) are registered in the active store of the NC indirectly via buffer stores. Associated with these influences are:

- mirror image
- external zero offset group (8M/8MC)
- external R parameter input
- external additive ZO
- external ZO
- synchronous machining (8MC)
- external tool offset
- block delete (switch on the operator's panel)
- text in clear for the user after programmed stop M00

These influences may be activated (e.g. using M-functions).

If these functions which are actuated in the active program are to be effective in the block following their selection, the block buffer store must be emptied. Alternatively the selected control signal only becomes active several blocks later.

In each program the buffer store can be emptied by a single call-up of the subroutine L999. The subroutine L999 must be defined as follows:

```
L999  00  LF
@31       LF
M17       LF
```

The control registers the status "buffer store empty" in the interface control and the selected control signal or the required external data input can be enabled.

First example:

Activation of external tool offset, e.g. after a measurement of the tool.

N15 M...	Read in activation of external TO
N20 L999	Empty buffer store. Pefore block N15 is not carried out, no further calculations are made.
N25 ...	The new TO is calculated

Second example:

Text in clear for the user after M00
N.. M00 L999 LF
(operational instruction) readable in the p.p.-figure
N..

Attention: @ 31 must be alone in its own block

6. Canned Cycles

6.1 Boring Cycles G81-G89

A boring cycle (working cycle) defines a series of machine and motion events (acc. to DIN 66025) necessary to drill, bore, tap, or preform some other task.
The boring cycles G81 thru G89 are stored in the control as subroutines L81 to L89.

Canned Cycle No.	Canned Cycle Subroutine	Traverse Rate into the Part After Positioning to the Reference Plane	At Hole Bottom Dwell	At Hole Bottom Spindle	Retract to the Reference Plane	User Example
0	L8000	-	-	-	-	Cancels L81-L89
1	L8100	in feed	-	-	in rapid	Drilling, centering
2	L8200	in feed	yes	-	in rapid	Drilling, counter sinking
3	L8300	in feed, start-up	-	-	in rapid	Deep hole drilling
4	L8400	feed per revolution	-	reversal	in feed	Tapping
5	L8500	with rapid traverse	-	-	with rapid traverse	Boring 1
6	L8600	Spindle on, in feed	-	stop	in rapid	Boring 2
7	L8700	Spindle on, in feed	-	stop	manual retract	Boring 3
8	L8800	Spindle on, in feed	yes	stop	manual retract	Boring 4
9	L8900	in feed	yes	-	in feed	Boring 5

The user may deviate from a standard fixed cycle and redefine it to suit his specific machine or tooling requirements. The parameters R00 thru R11 are used by the subroutine to define the variable values necessary to correctly execute a fixed cycle (e.g. reference plane coordinates, the hole depth, feed rate, dwell time, etc.) Proir to a subroutine call, all the necessary parameters must be defined.

A fixed cycle call is initiated with G80 to G89. G81 to G89 are modal fixed cycles that are cancelled with G80. A fixed cycle can be called with L81-L89, however, L81 to L89 are not modal. A G81 to G89 fixed cycle is executed at the end of every positioning move L81-L89 is performed only once, in the block in which it is programmed. At the end of a fixed cycle the tool is positioned to exit.

The cycles G81 to G89 end all in the same way with the preparatory functions G00, G60, and G90. When continueing the program G-functions that are different from these have to be programmed anew.

The Following R-Parameters are Used in Cycles L81-L89

R00 Dwell time at the start point (deburr hole)
R01 First depth advance (incremental) stored as an unsigned dimension
R02 Reference plane (absolute)
R03 Final depth (absolute)
R04 Dwell time length at hole bottom (break chips)
R05 Depth advance modifier stored as an unsigned dimension
R06 Reverse spindle rotation direction
R07 Return to the original spindle rotation direction used in the calling program (after R06 or M05)

R09 Depth advance or thread lead modifier. SR number and SR run for hole positions
R10 Retract position
R11 Drilling axis (Axis numbers from 1 to 10 selectable, e.g. X = 1, Y = 2, Z = 3, 10th axis = 10)

The cutter must be positioned to the correct location in the plane prior to the subroutine call. The appropriate feed, spindle speed, and rotation direction must be programmed in the calling routine. The fixed cycles are programmed for absolute dimensioning. After a return from a fixed cycle care must be exercied to insure that the correct dimensioning mode is again programmed.

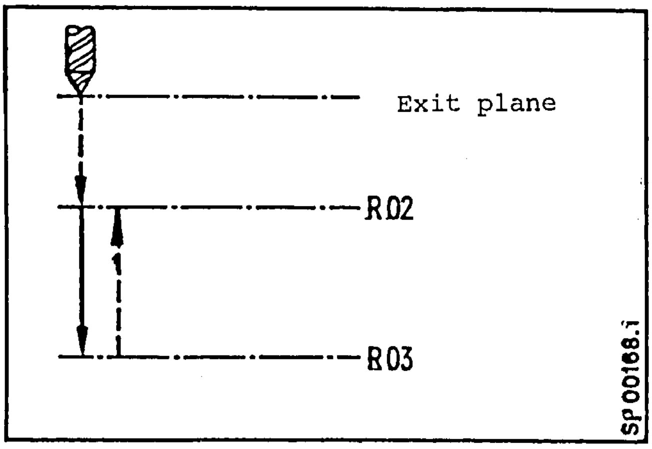
Subroutine L80: (cancels G81-G89)

G80 is an internal control function call.
No parameter definitions are required.

Subroutine L81: (Drilling, Centering)

The following parameters must be defined:

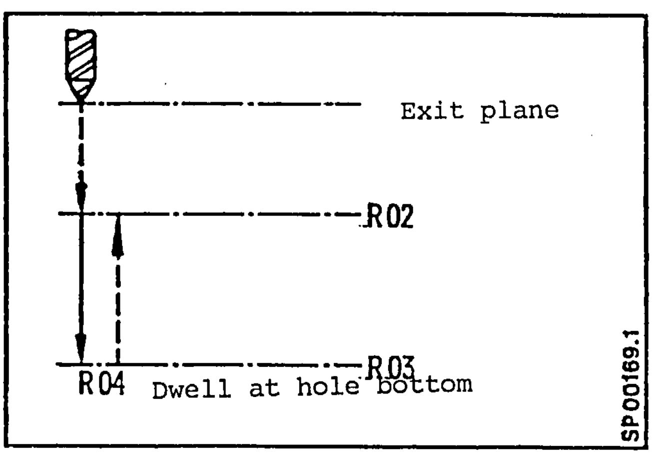
R02 Reference plane (retract position)
R03 Final hole depth
R11 Drilling axis



Subroutine L82: (Drilling, Counter Sinking)

The following parameters must be defined:

R02 Reference plane (retract plane)
R03 Final hole depth
R04 Dwell time
R11 Drilling axis

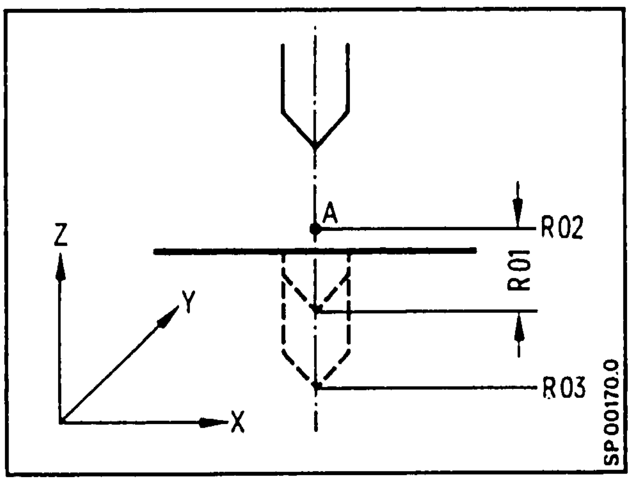


--------	Rapid traverse
————	Feedrate

Subroutine L83 (Deep Hole Drilling)

The following R-parameters must be defined prior to calling canned cycle L83.

R00 = Dwell is preformed at the start position. (To deburr hole.)
R01 = First depth advance (incremental) stored as an unsigned dimension.
R02 = Reference plane = retract plane (absolute) "A"
R03 = Final hole depth (absolute)
R04 = Dwell time length (break chips)
R05 = Incremental depth advance modifier stored as an unsigned dimension.
R11 = Drilling axis



R03 Final Hole Depth: The incremental depth diminishes with each successive drill amount till the final hole depth R03 is reached. If the incremental depth advance modifier exceeds the actual drill advance, succeeding drill advances will be held constant. At the end of the drilling cycle the drill is brought to point A.

If the remaining depth is greater than R05 and less than 2 times R05, it is divided into 2 drilling strokes.

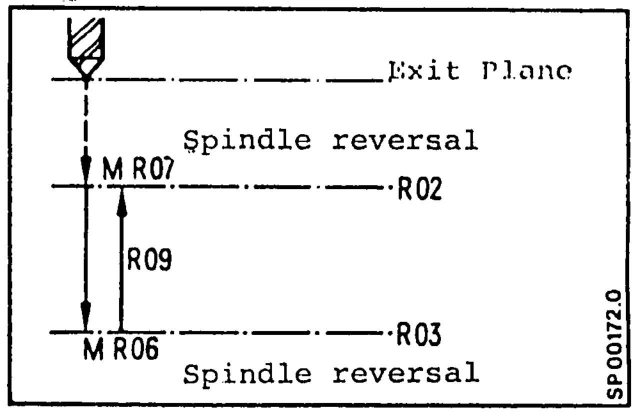
$$R05 < a < 2\ R05$$

a = remaining depth

Subroutine L84: (Tapping with Spindle Encoder)

The following R-parameters must be defined

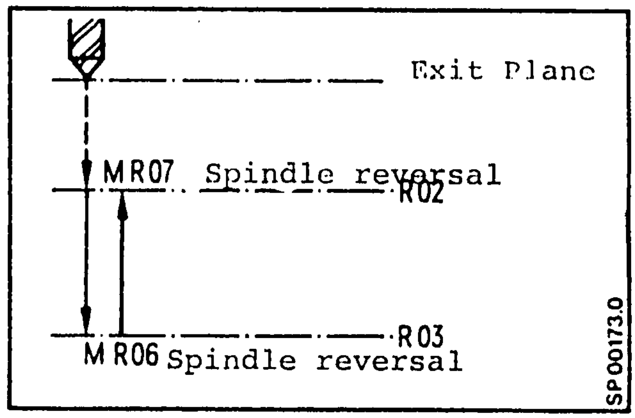
R02 Reference plane (retract position)
R03 Final depth
R06 Spindle rotation reversal
R07 Original spindle rotation direction.
R09 Thread lead dimension
R11 Drilling axis



Subroutine L84: (Tapping without Spindle Encoder)

The following R-parameters must be defined

R02 Reference plane (retract position)
R03 Final depth
R06 Spindle rotation reversal
R07 Original spindle rotation direction.
R11 Drilling axis

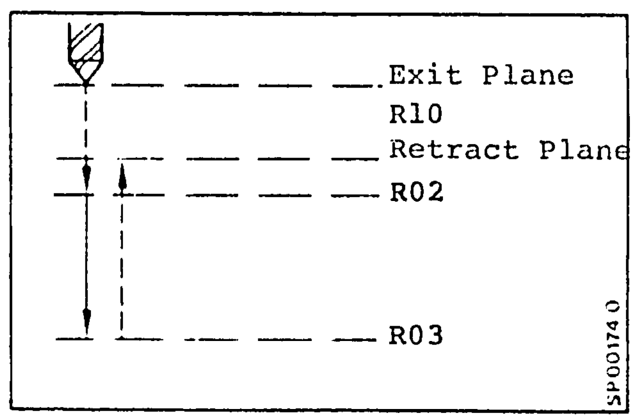


Subroutine L85: (Boring 1)

The following R-parameters must be defined.

R02 Reference plane
R03 Final depth
R10 Retract plane
R11 Drilling axis

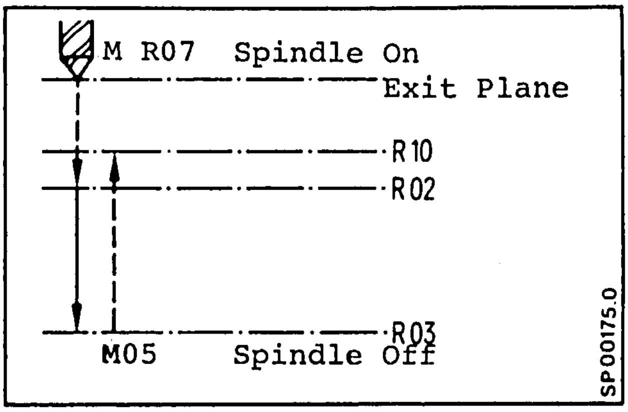
---- Rapid traverse
____ Feed rate



Subroutine L86: (Boring 2)

The following R-parameters must be defined:

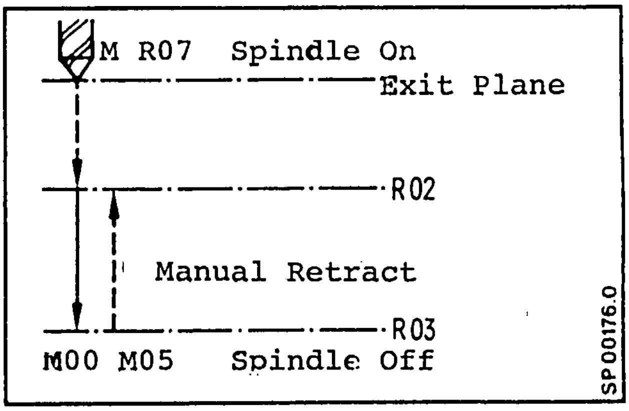
R02 Reference plane
R03 Final depth
R07 Spindle on (after M05)
R10 Retract plane
R11 Boring axis



Subroutine L87: (Boring 3)

The following R-parameters must be defined:

R02 Reference plane (retract position)
R03 Final depth
R07 Spindle on (after M05)
R11 Boring axis



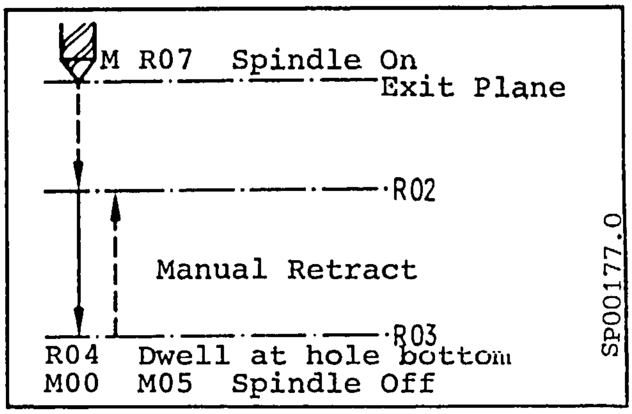
Subroutine L88: (Boring 4)

The following R-parameters must be defined:

R02 Reference plane (retract position)
R03 Final depth
R04 Dwell time length
R07 Spindle on (after M05)
R11 Boring axis

Fixed cycle L88 (G88) is similar to L87 (G87), however, a dwell is preformed at the bottom of the hole.

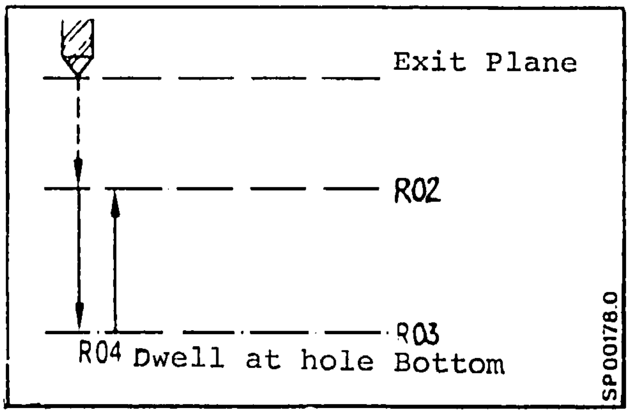
Rapid traverse — — — —
Feed rate ————————



Subroutine L89: (Boring 5)

The following R-parameters must be defined:

R02 Reference plane (retract position)
R03 Final depth
R04 Dwell time length
R11 Boring axis



Rapid traverse — — — —
Feed rate ————————

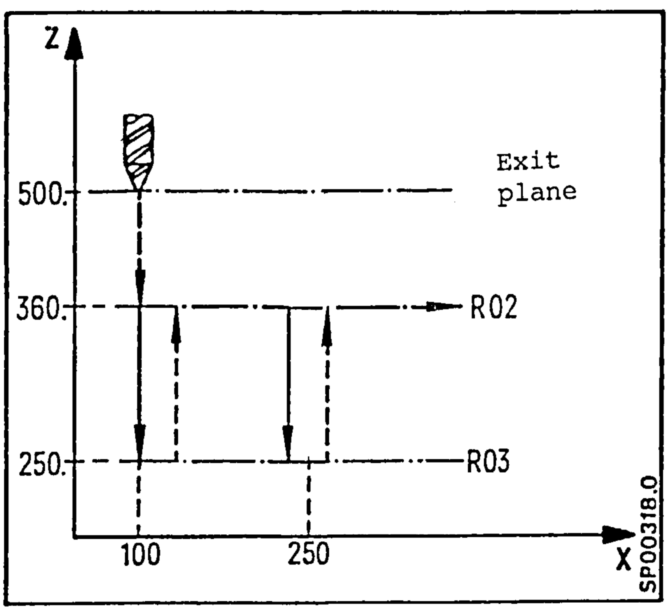
6.2 Examples of Limitations in Cycle Call-up

The drill cycle for every hole to be machined is called up only after the drill position has been reached.

The preparatory functions G81 through G89 can call up the subroutines L8100 through L8900 for a cycle run. At every drill position a called up drilling cycle is activated; and is active only in the called up subroutine plane. This modal drilling is cancelled with G80. If the cycles G81 - G89 are run with any remarks written in the program these remarks have to be given in blocks with departure data. If one of these remarks is written alone between 2 LF characters a drill cycle will also be executed.

Call-up G81 (drilling, centering)

```
N8101  G90  S48  M03  F460                  LF   - Spindle ON
N8102  G00  D01  Z500.                      LF   - Activate tool offset
N8103  X100.  Y150.                         LF   - 1st drill position
N8104  G81  R02 360.  R03 250.  R11  03     LF   - Call-up cycle
N8105  X250.  Y300.                         LF   - 2nd drill position
  .                                                and automatic
  .                                                G81 call-up
N8110  G80            Z500.                 LF   - Cancelling G81 and
                                                   returning to Exit plane
```



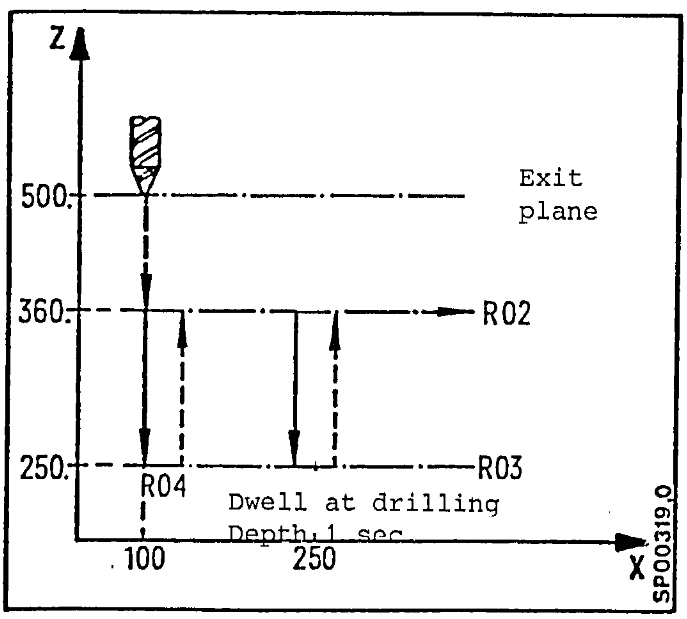
Call-up with L81

```
N8101  G90  S48  M03  F460.                    LF
N8102  G00  D01  Z500.                         LF
N8103  X100.  Y150                             LF
N8104  L81  R02 360.  R03 250.  R11 03         LF    - Call-up drill cycle
                                                       1st hole
N8105  X250.  Y300.                            LF
N8106  L81  R02 . . .                                - Call-up drill cycle
.                                                      2nd hole
.
N81..  Z500                                    LF
```

As opposed to the call-up with G81, here the drill cycle must be called up anew at every new drill position.

Call-up G82 (drilling, counter sinking)

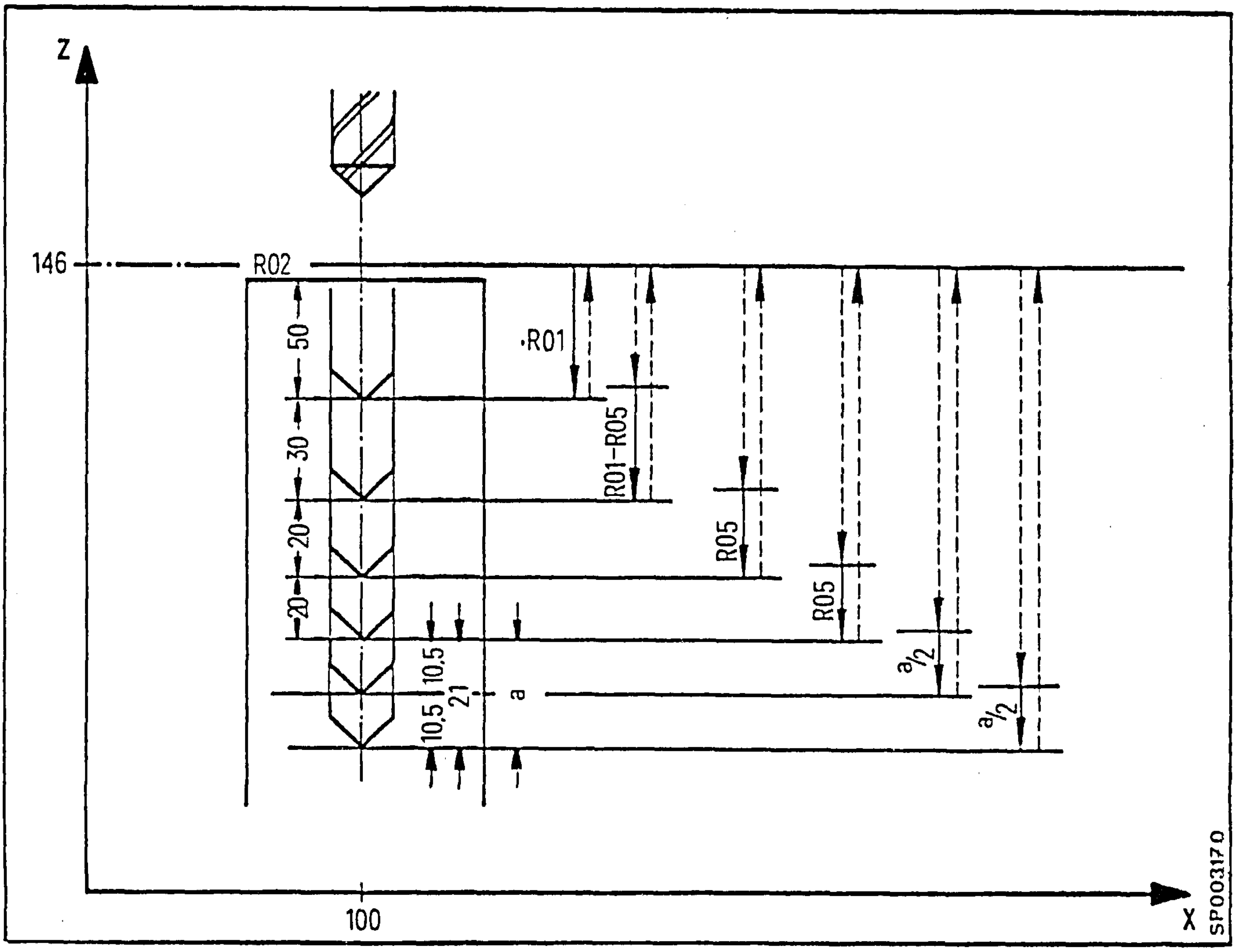
```
N8210  ...  M03  F460                                  LF
N8202  G00  D01  Z500.                                 LF
N8203  X100.  Y150.                                    LF
N8204  G82  R02 360.  R03 250. R04 1. R11 03           LF
N8205  X250.  Y300.
.
.
N82  ..  G80  Z500.                                    LF
```



Call-up G83 (deep hole drilling)

First drilling depth	50	mm	R01 50.
Reference level = retract level	146	mm	R02 146.
Final drilling depth	5	mm	R03 5.
Dwell at starting point	5	s	R00 5.
Dwell at drilling depth	1	s	R04 1.
Degression value	20	mm	R05 20.
Drilling axis (here Z)			R11 03.

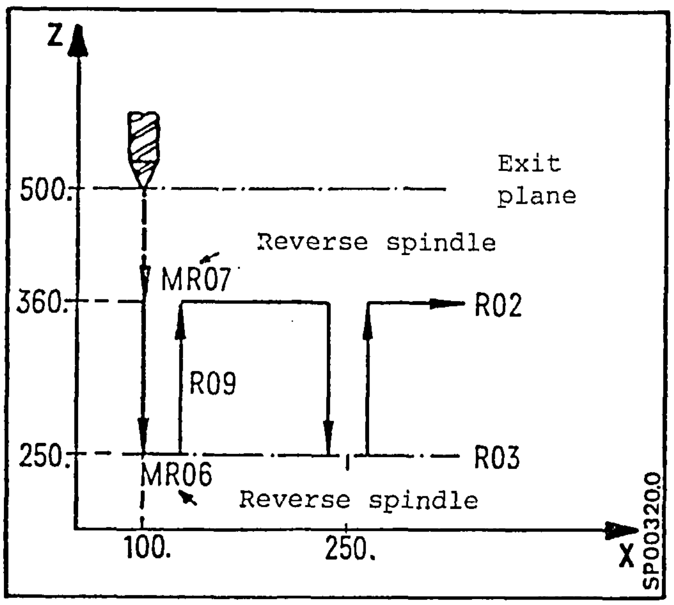
```
N8301  ...  S48  M03  F460.                                       LF
N8302  G00  D01  Z500.                                            LF
N8303  X100.  Y150.                                               LF
N8304  G83  R01 50.  R02 146.  R03 5.  R00 5  R04 1.  R20.  R11 03  LF
N8305  X250.  Y300.                                               LF
.
.
N83..  G80  Z500.                                                 LF
```



At the rapid traverse advance repective to the new drilling depth, a safety distance of 1 mm will be kept (taking care of the chips still remaining in the hole).

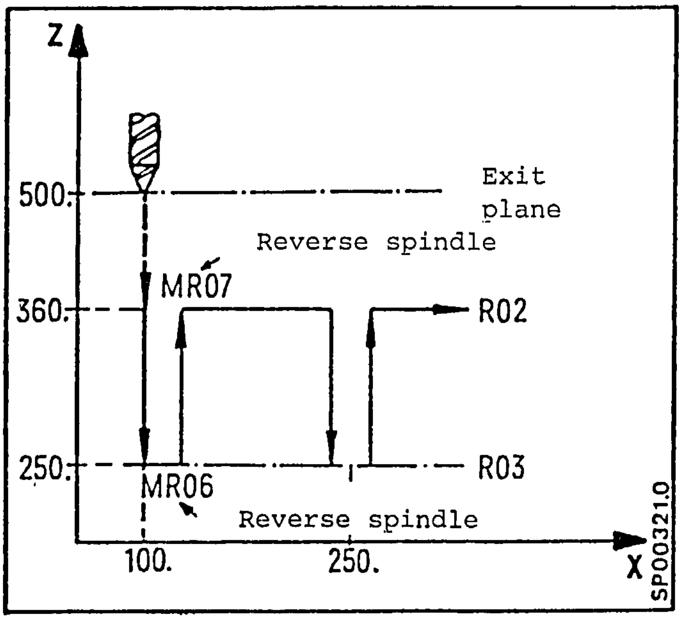
Call-up G84 (Tapping for machines with spindle encoder)

```
N8401   ...   S48   M03   F460.                                              LF
N8402   G00   D01   Z500.                                                    LF
N8403   X100.   Y150.                                                        LF
N8404   G84   R02 360.   R03 250.   R06 04   R07 03   R09 5.   R11 03        LF
N8405   X250.   Y300.                                                        LF
.
.
N84..   G80   Z500.                                                          LF
```



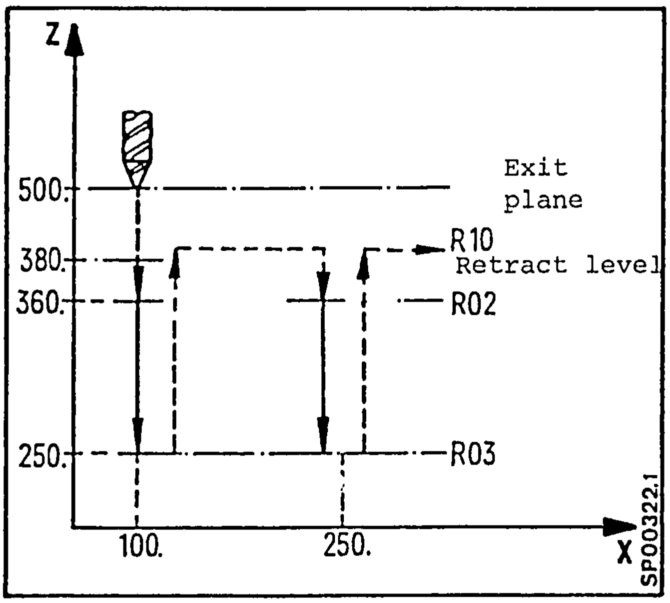
Call-up G84 (Tapping for machines without spindle encoder)

```
N8401   ...  S48  M03  F460                               LF
N8402   G00  D01  Z500.                                   LF
N8403   X100.  Y150.                                      LF
N8404   G84  R02 360.  R03 250.  R06 04  R07 03  R11 03   LF
N8405   X250.  Y300.                                      LF
.
.
N84..   G80  Z500.                                        LF
```



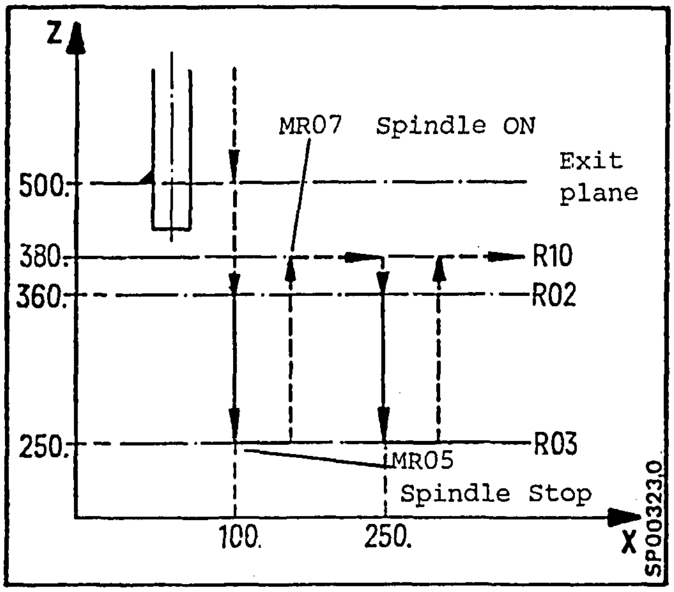
Call-up G85 (Boring 1)

```
N8501   ...   S48   M03   F460.                       LF
N8502   G00   D01   Z500.                             LF
N8503   X100.  Y150.                                  LF
N8504   G85   R02 360   R03 250.   R10 380   R11 03   LF
N8505   X250.  Y300.                                  LF
.
.
N85..  680   Z500                                     LF
```



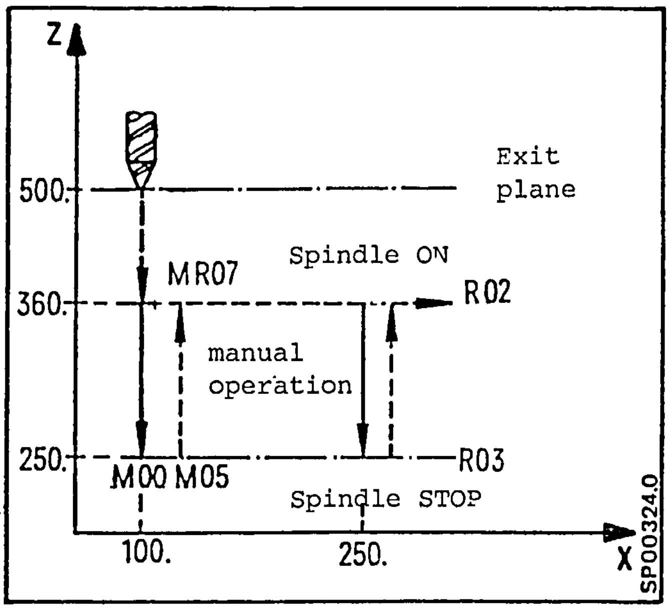
Call-up G86 (Boring 2)

```
N8601   ...   S48   M03   F460.                                  LF
N8602   G00   D01   Z500.                                        LF
N8603   X100.   Y150.                                            LF
N8604   G86   R02 360.   R03 250.   R07 03   R10 380.   R11 03   LF
N8605   X250.   Y300.                                            LF
.
.
N86..   G80   Z500.                                              LF
```



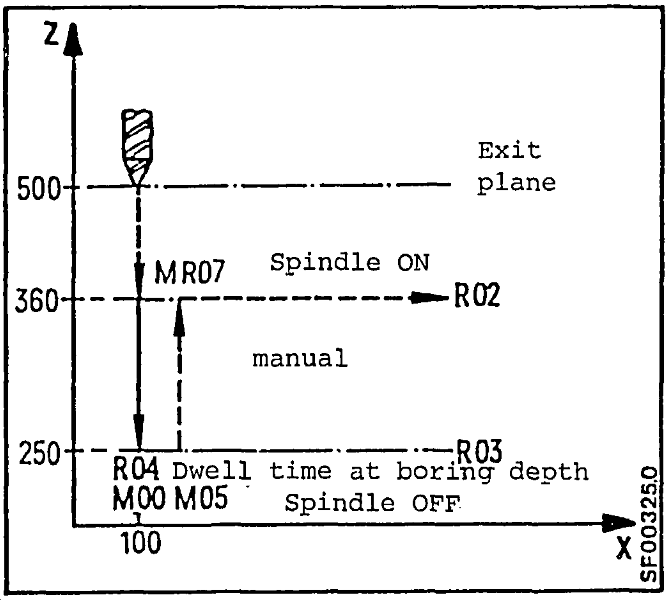
Call-up G87 (Boring 3)

```
N8701   ...   S48   M03   F460.                        LF
N8702   G00   D01   Z500.                              LF
N8703   X100.  Y150.                                   LF
N8704   G87   R02 360.   R03 250.   R07 03   R11 03    LF
N8705   X250.  Y300.                                   LF
.
.
N87..   G80   Z500.                                    LF
```



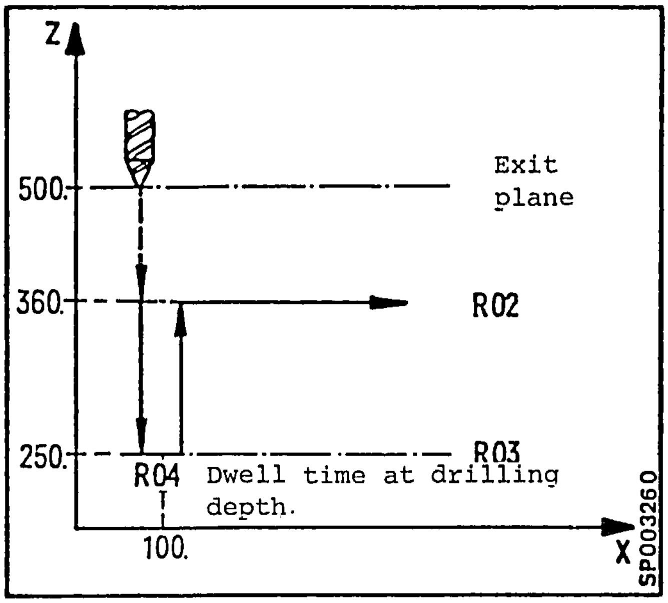
Call-up G88 (Boring 4)

```
N8801   ...   S48   M03   F460.                                  LF
N8802   G00   D01   Z500.                                        LF
N8803   X100.   Y150.                                            LF
N8804   G88   R02 360.   R03 250.   R04 1.   R07 03   R11 03     LF
N8805   X250.   Y300.                                            LF
.
.
N88..   G80  Z500.                                               LF
```



Call-up G89 (Boring 5)

```
N8901   ...  S48  M03  F460.                     LF
N8902   G00  D01  Z500.                          LF
N8903   X100.  Y150.                             LF
N8904   G89  R02 360.  R03 250.  R04 1.  R11 03  LF
N8905   X250.  Y300.                             LF
.
.
N89..   G80  Z500.                               LF
```



6.3 Subroutine Pattern L900, switchable axis in X, Y, Z, 4 *

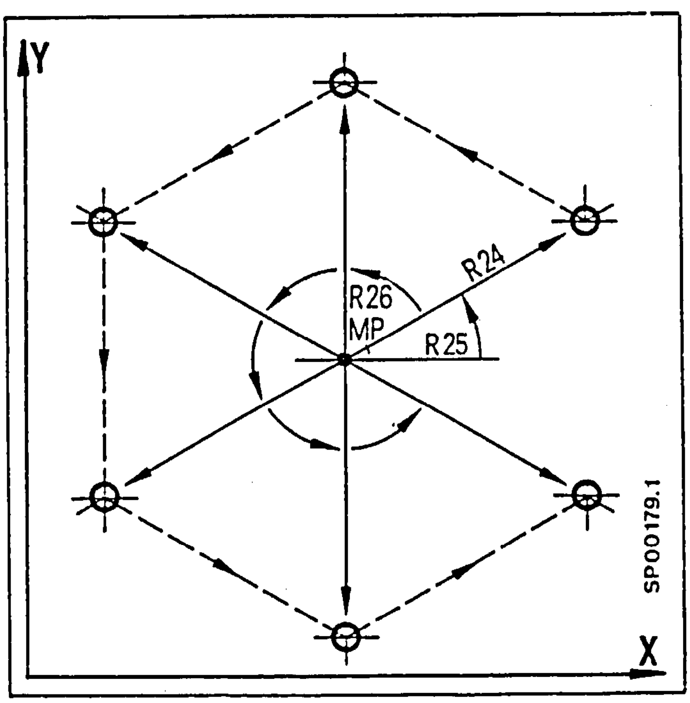
Note:

The milling and boring patterns L900 - L905 are programmed in absolute dimensioning. The radius is programmed with address the angle is programmed with address A.
The boring and milling patterns L900 - L905 will all together be finished with the G-functions G00, G60, G90 as well as with the cancelled cutter radius compensation G40. The tool length compensation remains selected.
Before the call of the cycles the cutter radius compensation must be cancelled, the length offsets must however be active. According to requirement either the axis switchable cycles L900,L902,L904 or the Z-advance cycles L905,L901,L903 may be used.

When programming, only the subroutine "drilling pattern" is called up, and the following parameters are to be supplied with their respective values.

R11	Drilling axis (X = 1, Y = 2, Z = 3, 4=4)
R22, R23	MP - centre point of the hole pattern, given in reference to the part's zero point
R24	Radius
R25	Initial angle (in reference to the horizontal axis)
R26	Incremental angle If incremental angle programmed is 0, the number of holes will be divided from 360°.
R27	Number of holes
R28	The number for the desired drill cycle (81-89)

Example: XY-plane
Drilling axis Z is selected through R11 03



*(Presupposition: programming of radius and polar coordinates)

Subroutine call-up:

.
N1900 L90001 R11... R22... R23... R24... R25... R26...
R27... R28... ... LF
N1901 ...
The necessary parameters for the drill cycle must also be defined, e.g. in the preceeding block of the call-up for L900.

6.4 Subroutine, boring pattern L905, boring axis Z *

While the subroutine L900 is axis switchable in X, Y, Z, the Z axis is obligatory for the boring pattern.
The parameter R11 is therefore inapplicable.

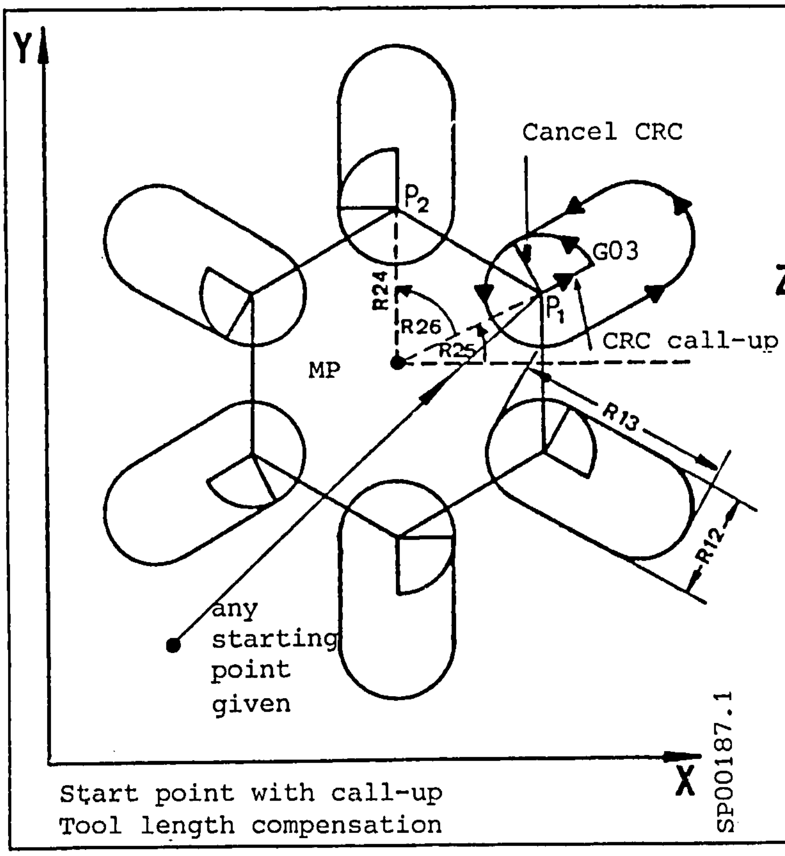
Subroutine call:

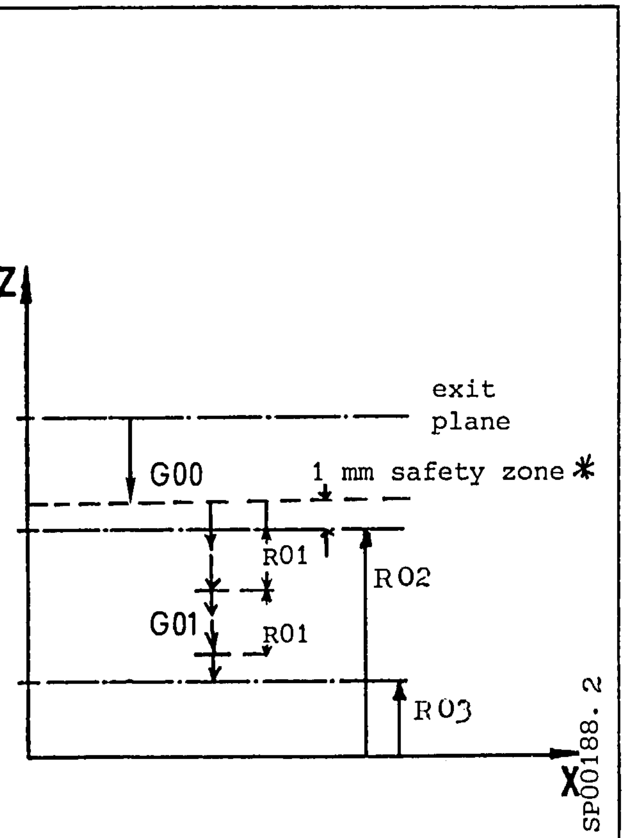
.
.
N10 R22.. R23.. R24.. R25.. R26.. R27.. R28.. L90501 LF

6.5 Milling Pattern "Groove" L901, machining axis Z *

When programming, only the subroutine "groove" is called up, and the following parameters should be supplied with their respective values. Subroutine L901 functions only in the XY-plane.

R01	First depth advance (inremental, without sign)
R02	Reference level (absolute)
R03	Depth of groove (absolute)
R22, R23	MP - centre point of the groove pattern, in reference to the part's zero point
R24	Radius (distance MP to groove edge)
R25	Initial angle (in reference to the horizontal axis)
R26	Incremental angle If the incremental angle programmed is 0, the number of grooves will be devided from 360°
R27	Number of grooves
R12, R13	"Groove" parameters; R12 groove width, R13 groove length (both incremental)
R14 (8MC only)	Tool offset no. of the milling tool (radius)





* taken into account in the SR

Before calling up the milling pattern "groove", the tool length compensation has to be selected with D.. (≠ 0).

```
.
.
N10  R01  R02..  R03..  R22..  R23..                 LF
N25       R24..  R25..  R26..  R27..  R12..  R13..  LF
N20          D05     L90101                          LF
.
.
.
```

6.6 Subroutine Milling Pattern "Groove" L902 * 1)

While subroutine L901 can only be used in the XY-plane, the milling pattern L902 can be applied dynamically to other planes. The boring axis has to be defined in a parameter additional to the parameters defined for L901.

R11 advance axis (X = 1, Y = 2, Z = 3)

Subroutine call-up:

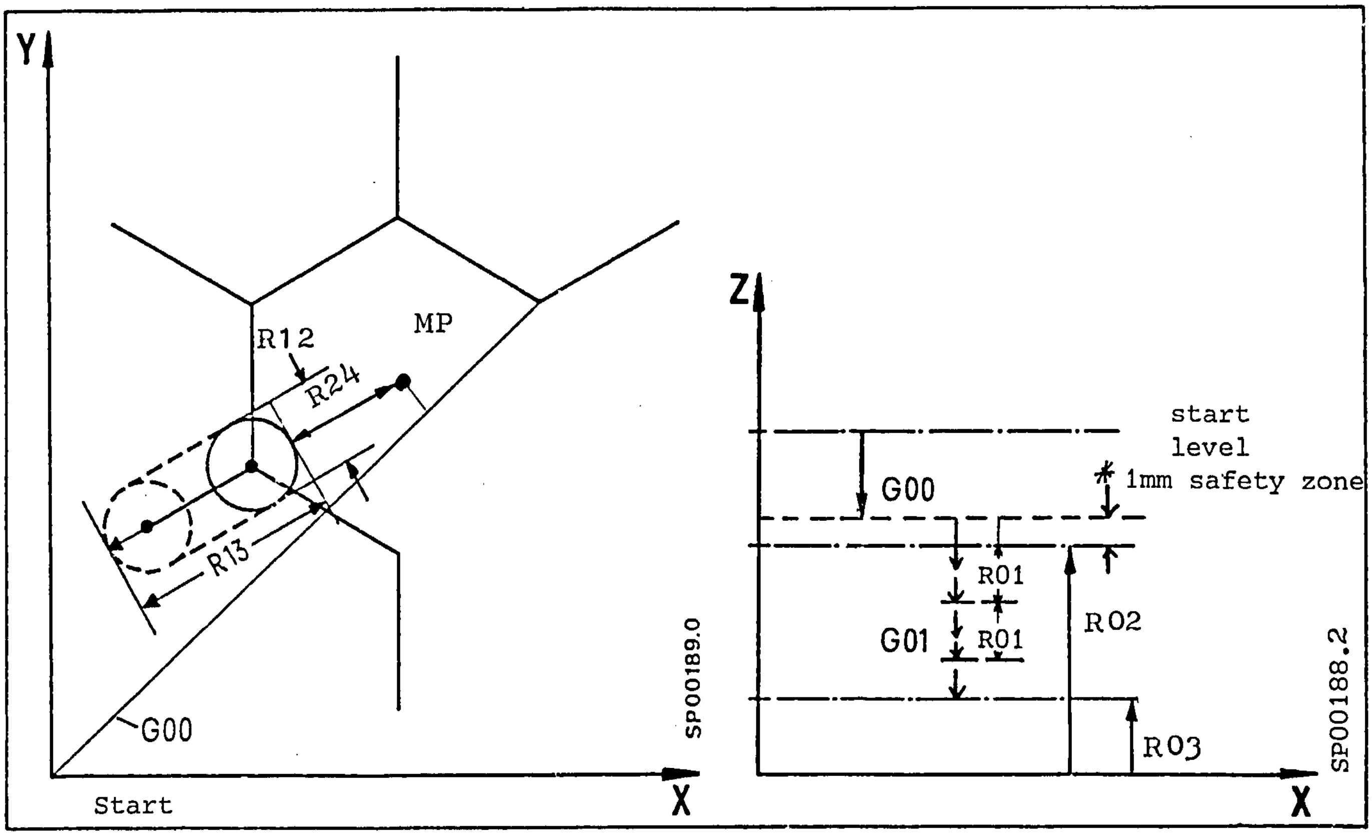
```
.
.
N10  R01  R02..  R03..  R22..  R23..  R24..  R25..
N15       R26..  R27..  R12..  R13..                LF
N20  D05  R11..  L90201                             LF
.
.
```

1) (Presupposition: programming of radius and polar coordinates)

6.7 Subroutine Milling Pattern "Elongated Hole" L903 , machining axis Z *

The "elongated hole" subroutine L903 functions only in the XY-plane. The following parameters must be defined before call-up.

R01	First depth advance
R02	Plane of reference
R03	Depth of the elongated hole
R22, R23	MP - centre point of the milling pattern, in reference to the part's zero point
R24	Radius
R25	Initial angle (in reference to the horizontal axis)
R26	Incremental angle If the incremental angle programmed is zero, the number of elongated holes will be divided accordingly
R27	Number of elongated holes
R12	Tool diameter
R13	Total length of hole



* is taken into account in the SR

Call-up L903

N10	R01	R02..	R03						LF
N15		R22..	R23..	R24..	R25..	R26..	R27..	R12..	R13.. LF
N20		D05	L90201						LF

.

.

6.8 Subroutine Milling Pattern "Elongated Hole" L904, axes switchable in X, Y, Z *

Subroutine L904 can be applied to other planes than the XY-plane. Additional to L903, the following must be defined:

R11 Boring axis (X = 1, Y = 2, Z = 3)

Subroutine call-up

.

.

N10	R01	R02..	R03..						LF
N15		R22..	R23..	R24..	R25..	R26..	R27..	R12..	R13.. LF
N20	D05	R11	L90401						LF

* (presupposition: programming of radius and polar coordinates).

7. Sprint 8M

7.1 Word Address System

The word address format and thereby the input format is defined by EIA RS 274-C and DIN 66025.

Sprint 8M: Metric

```
%04 N04 G02 XL+053 YL+053 ZL+053 UL+053 ID053 JD053 KD053
AL+035 PD053 F05 S04 H06 D02 T04 L5 R2 RL+08 M02 *
```

Inch

```
%04 N04 G02 XL+044 YL+044 ZL+044 UL+044 ID044 JD044 KD044
AL+035 PD044 F05 S04 H06 D02 T04 L5 R2 RL+08 M02 *
```

7.2 Motion Dimension, fourth axis

Sprint 8M

X, Y, Z and for the fourth axis addresses A, B, C, E, Q, U, V, and W can be used.

Fourth axis:

- The fourth axis only can be used als rotary axis.
- The fourth axis can be defined parallel to one of the primary motion axes X, Y, Z with a machine parameter.
- Which of the parallel axes is the secondary motion axes (e.g. Z or 4th axis) is determined by a signal from the interface. The signal may not change state after the start of the program.
- Circular interpolation between two parallel axes is not permitted. A secondary axis can be used in place of the primary axis to perform circular interpolation.
- Cutter radius compensation with the secondary motion is not possible.
- With constant surface speed and a zero point shift a tool length compensation for a facing head can be simulated.

7.3 Circular Interpolation with Interpolation Parameters

The start point of the circle or the circle arc is defined by the preceeding block. The end point is defined by the corresponding axes' values. The circle centre point is defined by the corresponding interpolation parameters.

I	Increment (signed) from circle start point to circle centre point	Parallel to X-axis
J		Parallel to Y-axis
K		Parallel to Z-axis

- When only one axis dimension is programmed, the missing dimension address is assumed to be from the plane selected by G17, G18 or G19. The last programmed value of this axis is used.
- The missing primary motion axis is always determined.
- The 4th axis may be defined as parallel to the X,Y or Z axis with machine parameters.
- The address of the circular interpolation parameter of the 4th axis is equal to the associated parallel primary axis.
- If an interpolation parameter is not programmed, zero is assumed.

Example:

N5 G17 G42 D03 ... LF	Plane and tool offset selection
N10 G03 X17. Y30. I-9. J8. LF	Complete definition of the circle with direction, circle end point coordinates and interpolation parameters.
.	
.	
.	
N25 G03 X17. I-9. LF	Circle programming with missing addresses. If no other plane or traverse distance in the Y-axis has been programmed between N10 and N25, the control generates the following:

N25 G17 G03 X17. Y30. I-9. J0. LF

7.4 Helical Interpolation *

(Presupposition: 3D-Interpolation)

Helical interpolation is possible between any three perpendicular axes. A block is programmed with one arc path and one linear path. The linear departure must be perpendicular to the plane in which the arc motion is generated. The programmed feed is maintained for the arc motion.

Example: Semi-circle with radius = 100 mm

```
N... G91 G02 X200. Y0. Z200. I100. J0. LF
N... G91 G02 X200. Y0. Z200. P+100.    LF
```

Circular interpolation plane (given by G17, G18 or G19) (both axes must be programmed) — X200. Y0.

The motion dimension for linear interpolation may be written before or after the P-word. — Z200.

Interpolation parameter or radius word — I100. J0. / P+100.

For example, if the 4th motion axis is declared parallel to the X primary motion axis, the following circular motion planes are valid for helical interpolation.

Linear	Circular		Interpolation Parameters	
X	Y	Z	J	K
Y	X	Z	I	K
Z	X	Y	I	J
4	Y	Z	J	K
Y	4	Z	I	K
Z	4	Y	I	J

The 4th motion axis uses the interpolation parameter assiciated with the parallel primary motion axis.

Note: Helical interpolation is not possible with Sprint 8ME.

7.5 Constant lead tapered threads

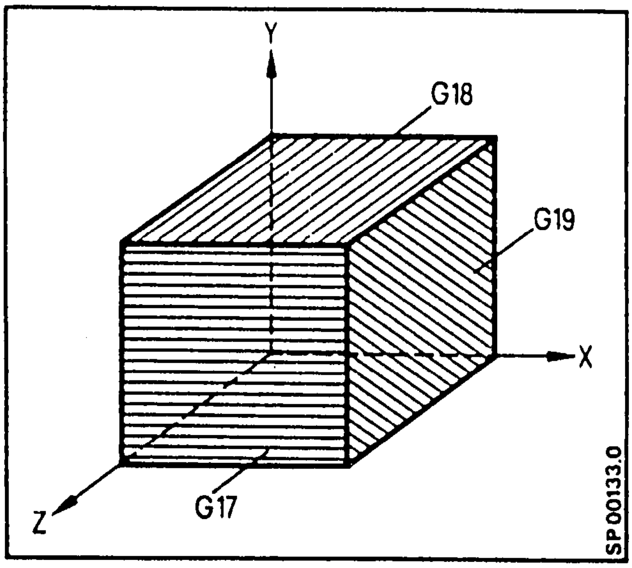
- For constant lead tapered threads, the thread lead is programmed for the leading axis.
- The leading axis is defined as the axis traversing the longest distance.
- For equidistant traverse in all axes, the leading axis is defined by the first axis programmed.
- The address pairing for thread cutting is I to X, J to Y, K to Z.
- I,J and K should always be entered in incremental without a sign. The input increment is equivalent to 1 or 0.001 mm/rev. (.0001 in/rev)
- The 4th axis may be used to cut threads irrespective of the "4th axis = main axis" signal.
- The thread pitch of the 4th axis m.y be programmed using either J (!) or K.
- G09 has to be programmed, if velocity reduction is desired at the end of the block.
- The thread length including acceleration and deceleration distance is programmed under the appropriate dimension address. In addition, the tool width must be taken into consideration.
- When thread cutting, the feed override, feed hold, spindle speed override, and single block switches are disabled.
- Pairing of thread lead and spindle speed - see chapter 8.2.4.
- The interpolation parameter of the non leading axis is not tested for validity: it may also be zero.

<u>Example: Incremental Dimensioning</u> (G91)

```
G91
G33  X20.  Z10.  I0.2                 Thread lead = 0.2

G33  X10.  Z20.  I0.2                 Incorrect programming
G33  X10.  Z20.  J0.2                 Z is the leading axis
                                      The thread lead must be
                                      programmed using K
G33  X10.  Z10.  I0.2                              = 0.2
G33  X10.  Z10.  K0.2                              = 0.2
G33  X10.  Z10.  I0.2  K0.2                        = 0.2 -|
      L----+----'     |                                   |
           L----------J                                   |  Permissible
                                                          |
G33  Z10.  X10.  I0.2  K0.2                        = 0.2 -'
      |     L----J     |
      L----------------J
```

7.6 G17, G18, G19 Machining plane selection



By programming G17 thru G19, the plane is defined in which cutter radius compensation is available. If a plane has not been selected at the start of the program, the default plane will be defined by G17 (default setting).

With a 4th axis the cutter compensation plane is defined as follows:

G-code \ 4th axis	Parallel to X	Parallel to Y	Parallel to Z	Machine without a 4th axis
G17	X-Y 4-Y Z-X	X-Y X-4	X-Y Z-X	X-Y
G18	Z-4	Z-X	4-X	Z-X
G19	Y-Z	Y-Z 4-Z	Y-Z Y-4	Y-Z

(4th axis - see chapter 7.2)

7.7 G40/G41/G42 Intersectional (look ahead) Cutter Radius Compensation

G40 Cutter compensation off
G41 Tool to the left of the part
G42 Tool to the right of the part

When mirror imaging is used and the sign is considered, the traversed path is as follows:

	Both axis are mirrored or Neither axis is morrored		one axis is mirrored	
Sign for the radius compensation value of the cutter				
	+	-	+	-
G41	left	right	right	left
G42	right	left	left	right

G40, G41 and G42 may be programmed in blocks programmed without motion preparatory functions. However, the function is not active until axis motion is programmed in at least one axis.

Fromsoftware stand 2 possible:

7.8.1 Intersection CRC with simultaneous tool length offset G43/G44

G43 Tool length compensation positive (reset state)
G44 Tool length compensation negative

Cancelling CRC	Cancelling of the length offset
with G40 or D00	only with D00

When selecting tool length compensation with D.. , G43 is active providing that G44 has not previously been programmed.
With G41 G17 D.. X.. Y.. Z.. the length compensation is active in Z and the CRC in the X-Y plane.

Selecting and Cancelling the Intersectional (look ahead) Cutter Radius Compensation

The selection is only possible, when G00 or G01 are active.

N10	G01	G17	G41	D07	X...	Y...	LF	When the end position of this block is reached, the path is correctly compensated. Only the radius is compensated, because the axis with the length compensation is not programmed.

or

N10	G17					Plane selection
N10	G41	D07			LF	Compensation selection
N20	G01		X...	Y...	LF	When the end position of this block is reached, the path is correctly compensated. Only the radius is compensated, because the Z axis is not programmed.
N25	G40		X...		LF	Compensation is cancelled, the length compensation remains active.

or

N25	G41	D00	X...		LF	Radius compensation is cancelled with D00. The length compensation is then cancelled, when Z is programmed.

With G40, the compensations G41/G42 are cancelled. However, at least one axis' motion must be programmed in order to restore the tool to its uncompensated path.
Length and radius compensation can both be cancelled, when D00 and the respective axis are programmed.

Switching from G41 to G42

N10 G01 G17 G41 D12 X... Y... Z... LF	Length and radius compensated
N15 G42 X... LF	Only the direction of the radius compensation is changed
N20 X... Z... LF	Length is not changed.

Calling a different tool offset function

The G-function (G41, G42) has not to be programmed.

N10 G01 G17 G41 D12 X... Y... LF	
N15 D10 Z... LF	Change in the length compensation
N20 X... Y... LF	Change in the radius compensation

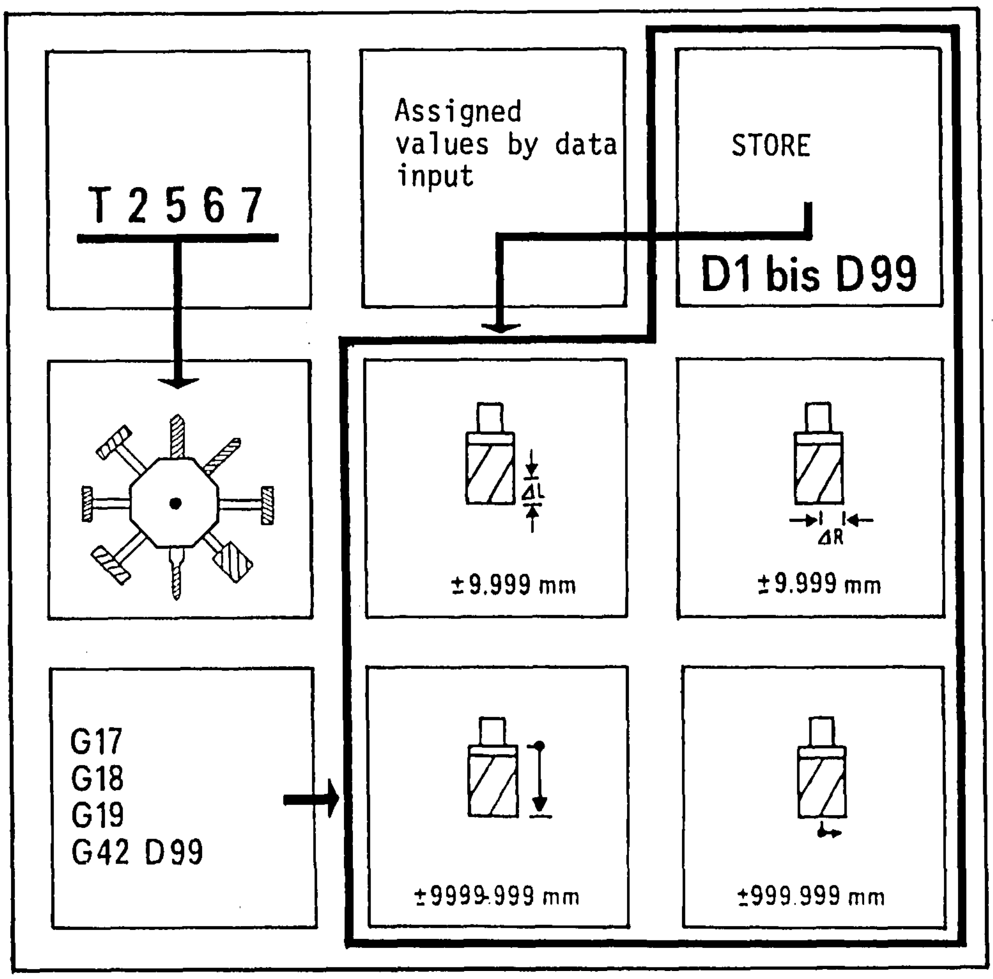
7.8 Tool Offset

Under the tool offset number the tool offset dimensions are stored.

The tool offset consists of:

Tool wear compensation:	Length ± 0,9999 in	(± 9.999 mm)
	Radius ± 0,9999 in	(± 9,999 mm)
Tool offset geometry:	Length ± 9999.9999 in	(± 9999.999 mm)
	Radius ± 99.9999 in	(± 999.999 mm)

A total of 99 offsets is available



Under the tool offset number the length and radius dimensions are stored. The wear compensation values for the length and and radius dimensions are input via the operator's panel.

Tool offset call and input (geometry)

A tool offset is called via a two digit designator D01 ..... D99 (length and radius pair).

- Input via tape

```
% TO LF
G92 D01 D... P... LF
.
.
.
.
G92 D99 D... P... LF
M02 or M30 LF
```

- Input via program

```
N11 ... LF
G92 D01 D... P... LF
.
.
.
.
G92 D99 D... P... LF
N12 ... LF
```

The length dimension is stored under D, the radius dimension under P.

Selecting and cancelling the length compensation

The selection is only possible, when G00 or G01 are active. At least one plane perpendicular to the plane in which the compensation should act, must be selected.

```
N5 G00 G17 D... Z...
```

Only the length compensation is used from the store D... . The compensation value contained in the D word is always taken in the calculation according to the programmed axis.

The cancellation of the length compensation is done via D00. The uncompensated position is reached when the respective axis is programmed.

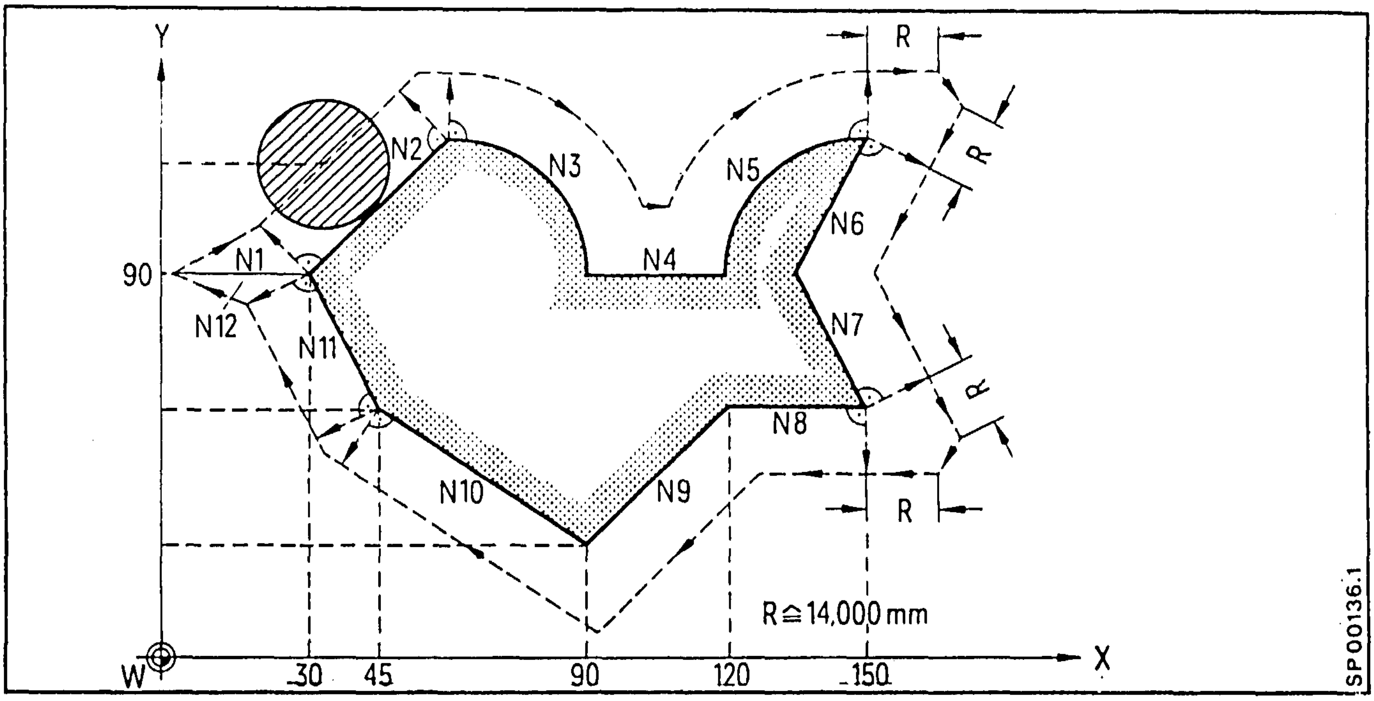
1. Length compensation without CRC

N5 G90 G00 G17 D01 Z500.	Selection of the length compensation (e.g. Boring tool)
.	
.	
N50 D00 Z0.	Cancellation of the length compensation via D00

2. Length compensation with CRC

N5 G90 G00 G17 G41 D02 X.. Y..	Automatic selection of the CRC with length compensation
N10 Z..	
.	
.	
N50 G40 X.. Z..	Cancellation of the CRC <u>only</u>. The length compensation is not cancelled.

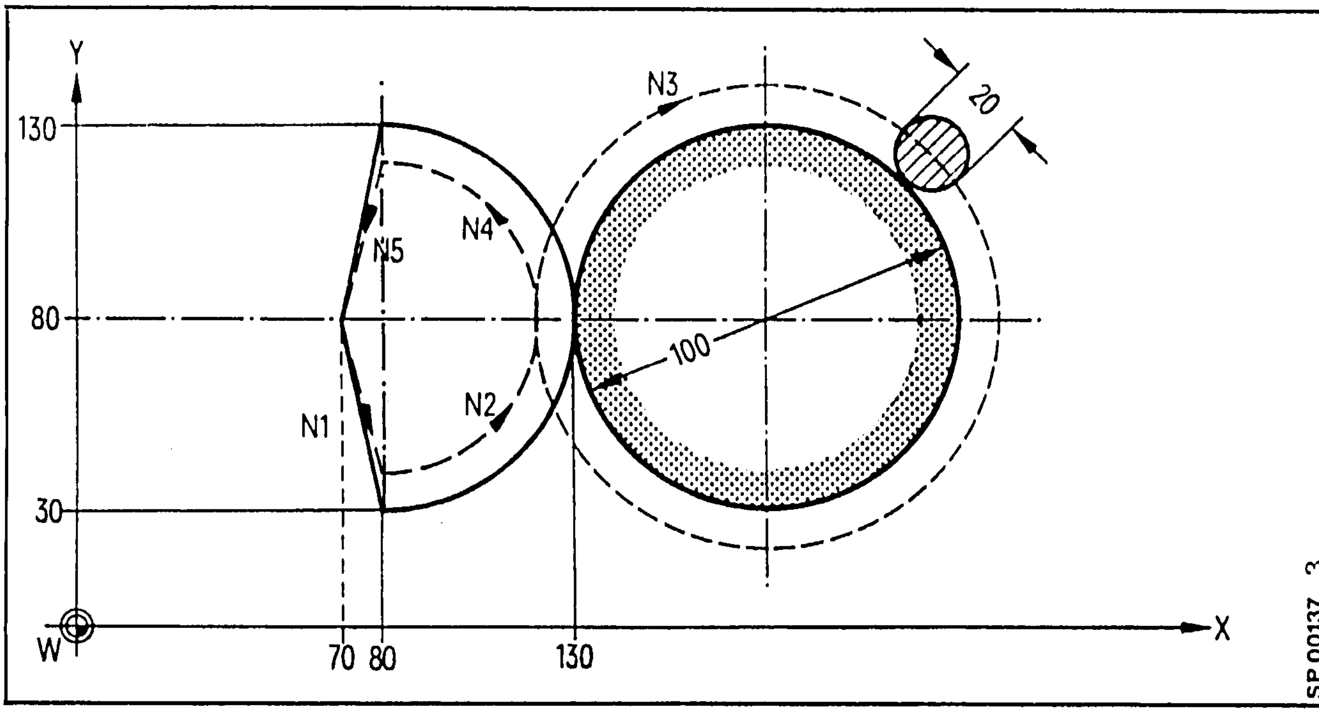
Example: Milling with CRC



N1	G01	G41	D01	G90	G17	X3.0	Y9.0	F5000	S56	M03	LF
N2	G91					X3.0	Y3.0				LF
N3	G02					X3.0	Y-3.0	I0.	J-3.0		LF
N4	G01					X3.0					LF
N5	G02					X3.0	Y3.0	I3.0	J.0		LF
N6	G01					X-1.5	Y3.0				LF
N7						X1.5	Y-3.0				LF
N8						X-3.0					LF
N9						X-3.0	Y-3.0				LF
N10						X-4.5	Y3.0				LF
N11						X-1.5	Y3.0				LF
N12	G40	G90				X0.	Y9.0				LF
N13. . .											

The cutter being used has a radius of 1 inch. The radius dimension is stored under address D01.

Example: 4 Quadrant Arc Programming with CRC



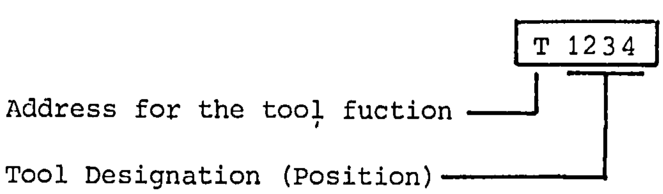
--------- Cutter center path

_________ Part contour

N1	G90	G00	G17	G41	D01	X80.	Y30.			LF
N2	G03					X130.	Y80.		J50.	
N3	G91	G02				X0.	Y0.	I50.	J0.	LF
N4	G90	G03				X80.	Y130.	I-50	J0.	LF
N5	G00	G40				X70.	Y80.			LF

7.9 Tool Function T

The tool function (T-function) designates the tool necessary for a machine operation.



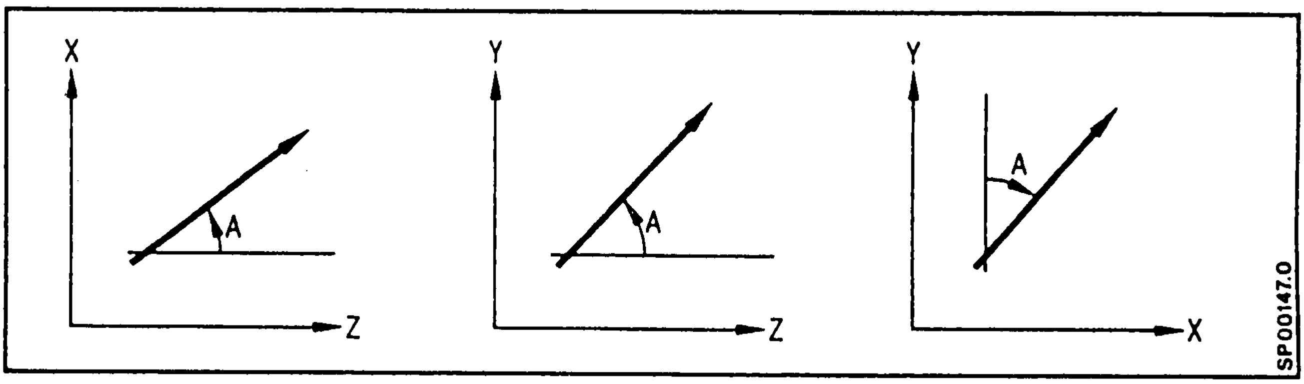
7.10 Blue Print Programming

The contour is described by multipoint paths programmed directly from the part drawing. The intersection point of two straight lines is determined from the coordinate values or an angle.

The transition between two straight lines may be abrupt when a sharp corner is desired. A radius or chamfer may be inserted at the intersection point. The chamfer and the radius are defined by a length dimension value. The geometric calculation is performed by the control. Absolute or incremental dimensioning may be used to define the end point coordinate.

Angle (A): Input resolution 0.00001°
The given angle (maximum 359,99999°) is always positive and measured with respect to the positive axis of the highest axis address.

Axis address value Z – Y – X



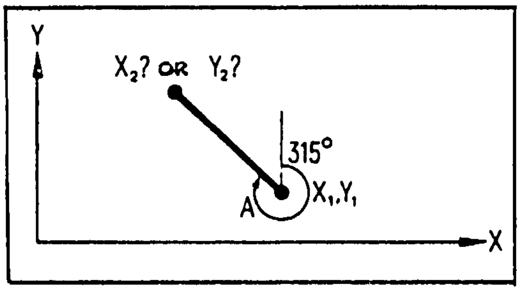
Caution:

Blue print programming works only in the selected plane. 3D machining is not possible.

7.10.1 Geometric Path - Programming

The examples 1) thru 8) illustrate the basic elements in geometric path programming. The basic patterns can be combined in various other ways (see pg. 7-20 and 7-21).

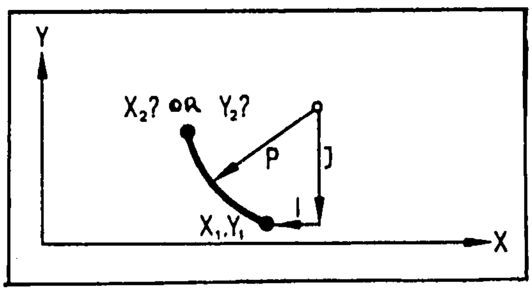
1) 2 Point Connected Path



N . . . A . . . X_2 . . . (or Y_2 . . .)

one end point is known, the other end point is calculated by the control.

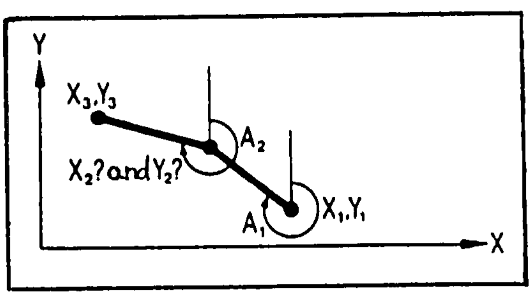
2) Arc Path



N . . . G02 (or G03) I . . . J . . P . . X_2(or Y_2)

The arc is limited to one quadrant. The end point is calculated by the control from the circle center location, radius and one end point dimension. Both the I and J-word must be programmed, even if they are zero.

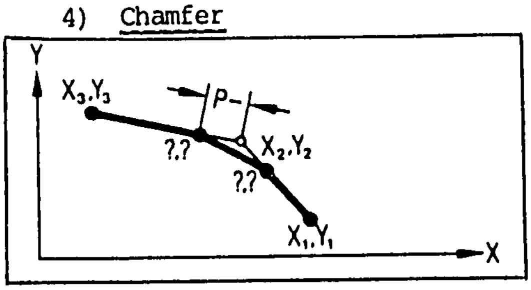
3) 3 Point Connected Path



N . . . A_1 . . . A_2 . . . X_3 . . . Y_3. . .

The control calculates the intersection point coordinate (X_2, Y_2) and generates two blocks of information. The angle A_2 measurement is with respect to the intersectional point X_2, Y_2.

4) Chamfer



N . . . X_2 . . . Y_2 . . . P-. . .

N . . . X_3 . . . Y_3 . . .

P designates a chamfer which is fitted to the two straight line paths at the intersection point

The "minus"sign does not mean that the P is signed, but characterizes P as a chamfer.

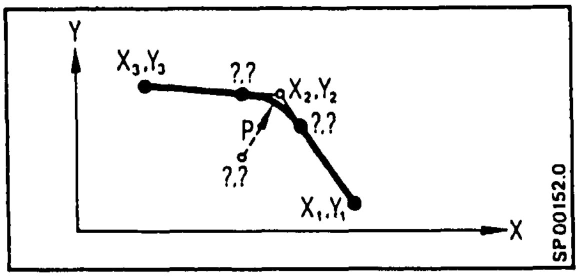
A distinction is drawn between a geometric path and an analytic path. A geometric path is one defined by angles, radii and end point coordinates while an analytic path is defined in the traditional sense ie. with end point coordinates circle center point parameters and user calculated intersection points for lines, chamfers, and radii.

5) Radius

N . . . X_2 . . . Y_2 . . . P . . .

N . . . X_3 . . . Y_3 . . .

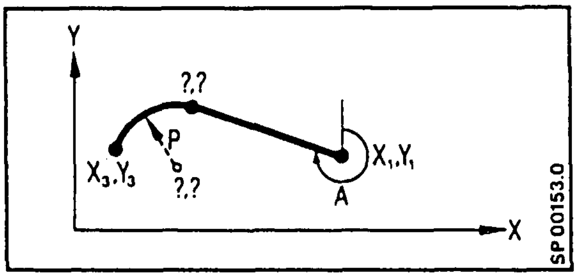
The insert radius may not be smaller than the shortest line segment length (X_2Y_2, X_3Y_3 or X_1Y_1, X_2Y_2)



6) A Straight-Tangential to an Arc Path

N . . . G02 (or G03) A . . P . . X_3 . . Y_3

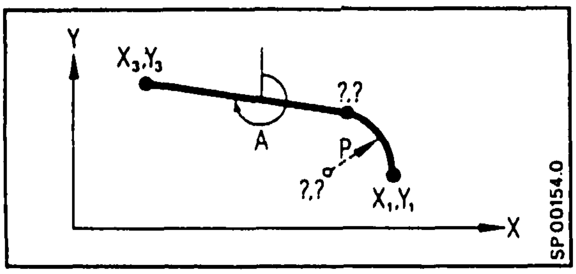
The arc must have a subtended angel less than 180°
The sequence A (angle) and P (radius) must be maintained



7) A Arc Tangential to a Straight Line Path

N . . . G02 (or G03) P . . A . . X_3 . . Y_3

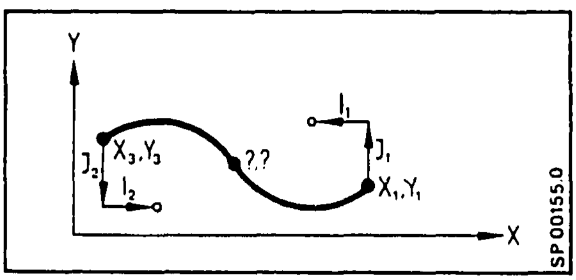
The sequence P (radius) and A (angle) must be maintained. A radius cannot be inserted at Y_3 Y_3. The arc must have a subtended angle less than 180°



8) Arc-Arc (Tangential) Path

N . . . G02 (or G03) I_1 . . J_1 . . I_2 . . J_2 . . X_3 . . Y_3 . .

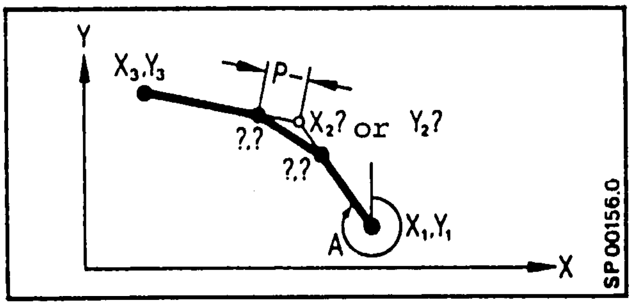
The preparatory function is always programmed for the first arc. The second preparatory function is always opposite to the first preparatory function.



1) & 4)
<u>2 Point Connected Path
+ Chamfer</u>

N . . . A . . X_2 . . (or Y_2) P- . .

N . . . X_3 . . . Y_3 . . . *

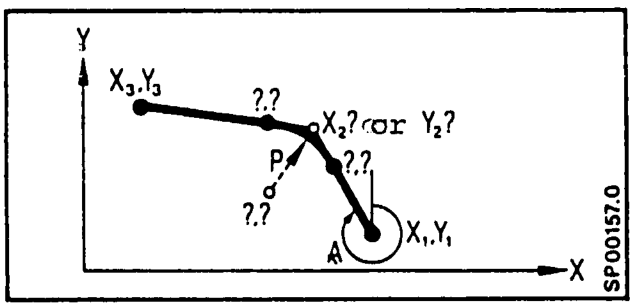


1) & 4)
<u>2 Point Connected Path
+ Chamfer</u>

N . . . A . . X_2 . . (or Y_2) P . .

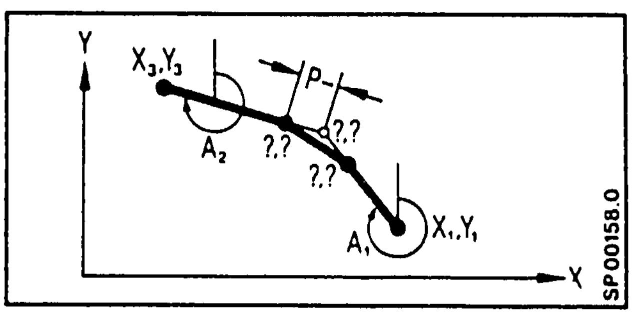
N . . . X_3 . . . Y_3 . . . *

The inserted radius must be smaller than the shortest line segment.



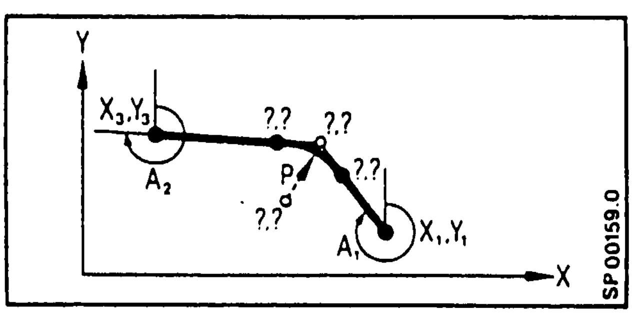
3) & 4)
<u>3 Point Connected Path
+ Chamfer</u>

N . . A_1 . . A_2 . . X_3 . . Y_3 . . P- . .



3) & 5)
<u>3 Point Connected Path
+ Radius</u>

N . . A_1 . . A_2 . . X_3 . . Y_3 . . P . .

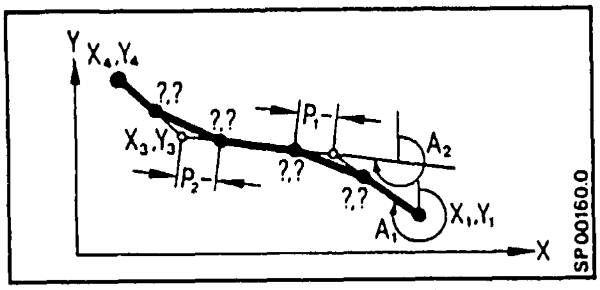


3) & 4) & 5)
3 Point Connected Path
+ 2 Chamfer

N 15 A_1 . . A_2 . . X_3 . . Y_3 . . $P-_1$ . . P_2-

N16 X_4 . . . Y_4 . . .

Inserting a 2nd chamfer at the end point (X_3, Y_3).

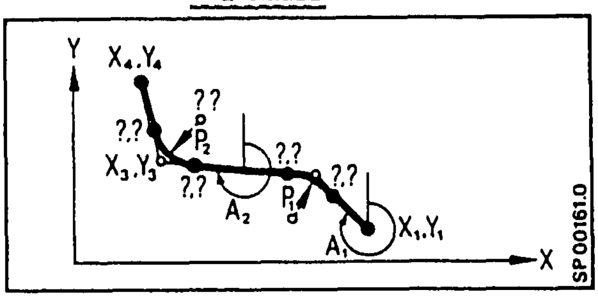


3) & 5) & 5)
3 Point Connected Path
+ 2 Radii

N15 A_1 . . A_2 . . X_3 . . Y_3 . . P_1 . . P_2 . .

N16 X_4 . . . Y_4 . . . *

Inserting a 2nd radius at the end point (X_3, Y_3).

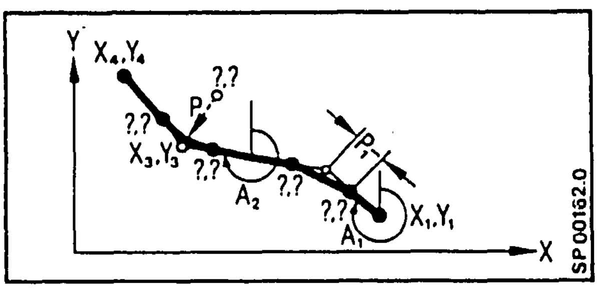


3) & 4) & 5)
3 Point Connected Path
+ Chamfer + Radius

N15 A_1 . . A_2 . . X_3 . . Y_3 . . P_1- . . P_2 . .

N16 X_4 . . . Y_4 . . . *

Inserting a radius at the end point (X_3, Y_3). The next path movement is automatically taken into consideration.

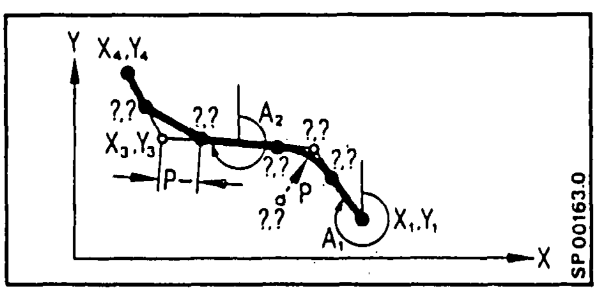


3) & 4) & 5)
3 Point Connected Path
+ Radius + Chamfer

N15 A_1 . . A_2 . . X_3 . . Y_3 . . P . . P- . .

N16 X_4 . . . Y_4 . . . *

Inserting a chamfer (P-) at the end point



* The second block may also be programmed as a contour path

When a sharp corner without radius or chamfer is desired, address P is programmed as P0.*
A radius or chamfer may be inserted, if the next programmed block describes an arc path.

Angles and radii must be written in the previously described sequence (first angle before second angle, first radius before second radius - in the direction of machining).

7.10.2 Geometric Path Programming with G09, F, S, T, H or M

When a G09 is programmed in a contour path, the function is not in effect until the end of the block ie. when the end position is approached.

Within the contour path G09 is generated automatically by the control at transition points (corners and edges).

If F, S, T, or H is programmed within a contour path, then they act at the start of the block.

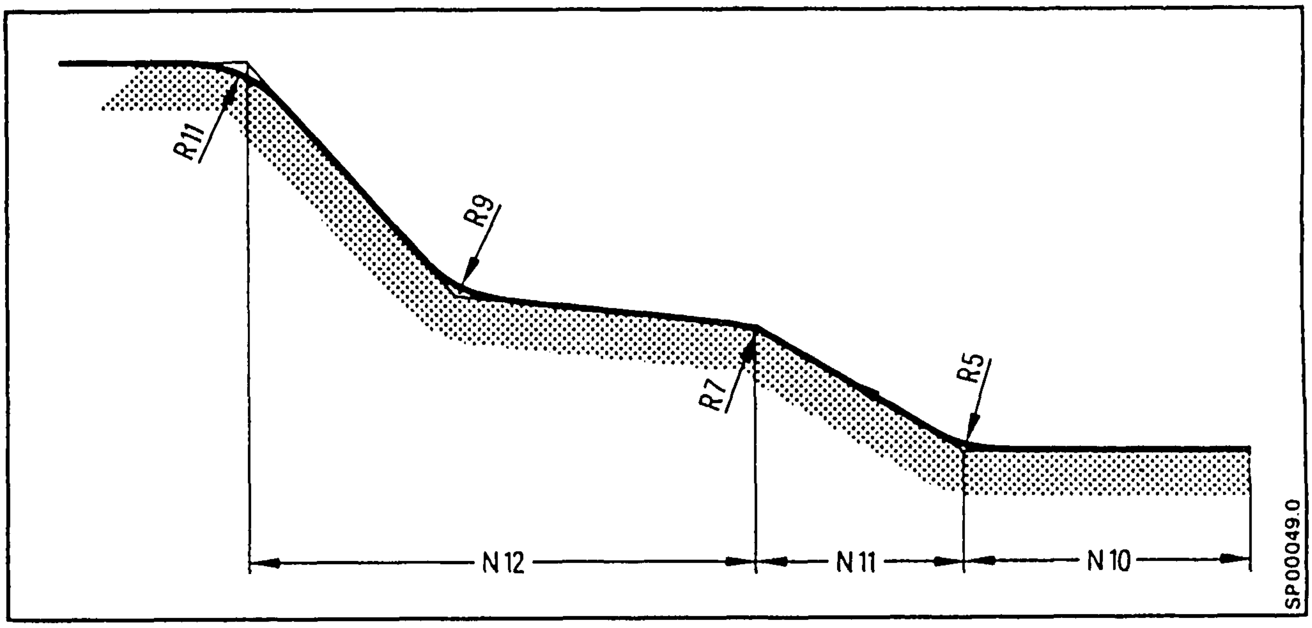
Within a contour path a programmed M00, M01, M02, M17 or M30 will be output at the end of the block.

* Attention: With this type of programming a block with a movement = 0 will be generated from the description of the contour. This must be noted together with the application and effect of the cutter radius compensation.(see section 8.1.6)

7.10.3 Linking Geometric Path Blocks

Several contiguous blocks using the blue print programming (geometric path) method may be linked arbitrarily.

All combinations linking straight line paths with or without radii and chamfers are possible.



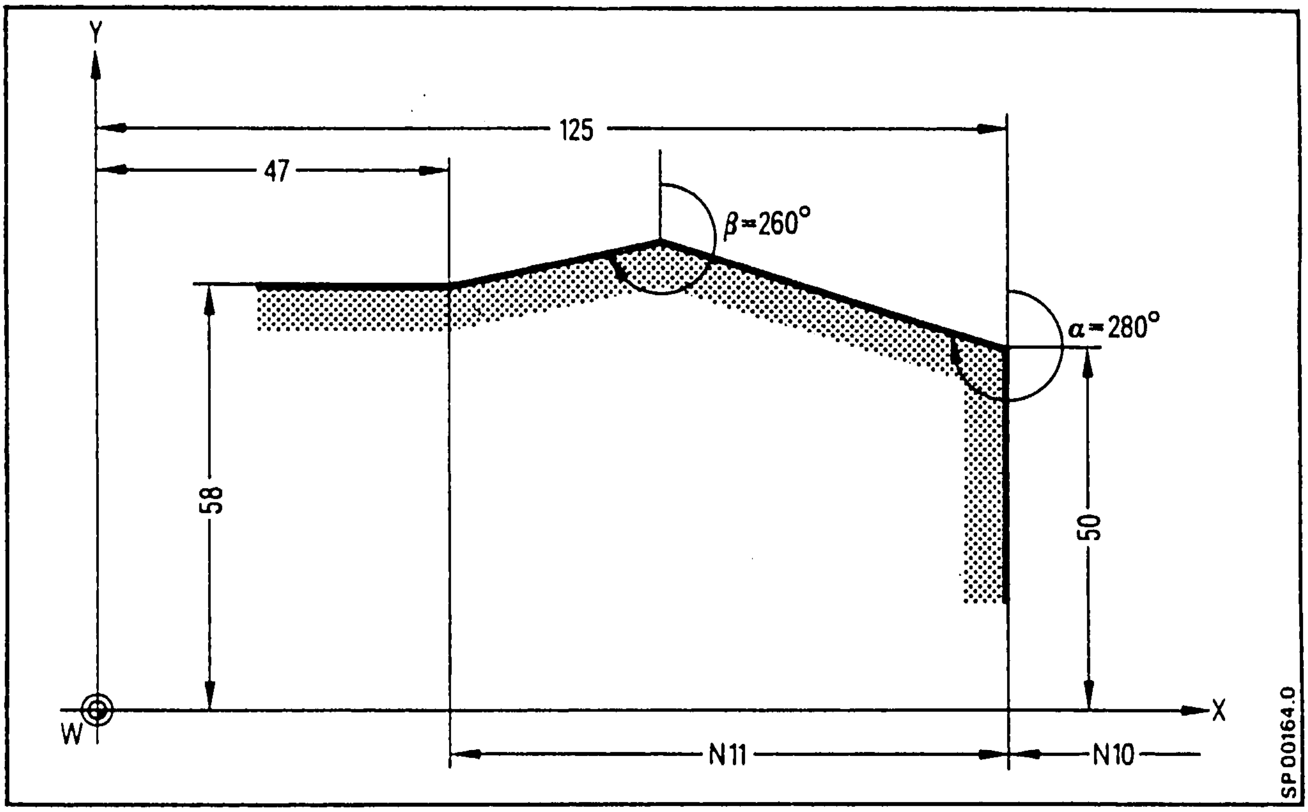

```
N10  Z . . .  P5.  LF
N11  A . . .  X . . .  P7.  LF
N12  A . . .  A . . .  X . . .  Y . . .  P9.  P11.  LF
```

7.10.4 Examples

The angle α is associated with the start point and the angle β with the as yet undetermined intersection point.

The end point may be programmed in absolute dimensions G90 or in incremental dimensions G91. Both end point coordinates must be known. The control calculates the intersecton point knowing the start point, the end point, and the two angles.

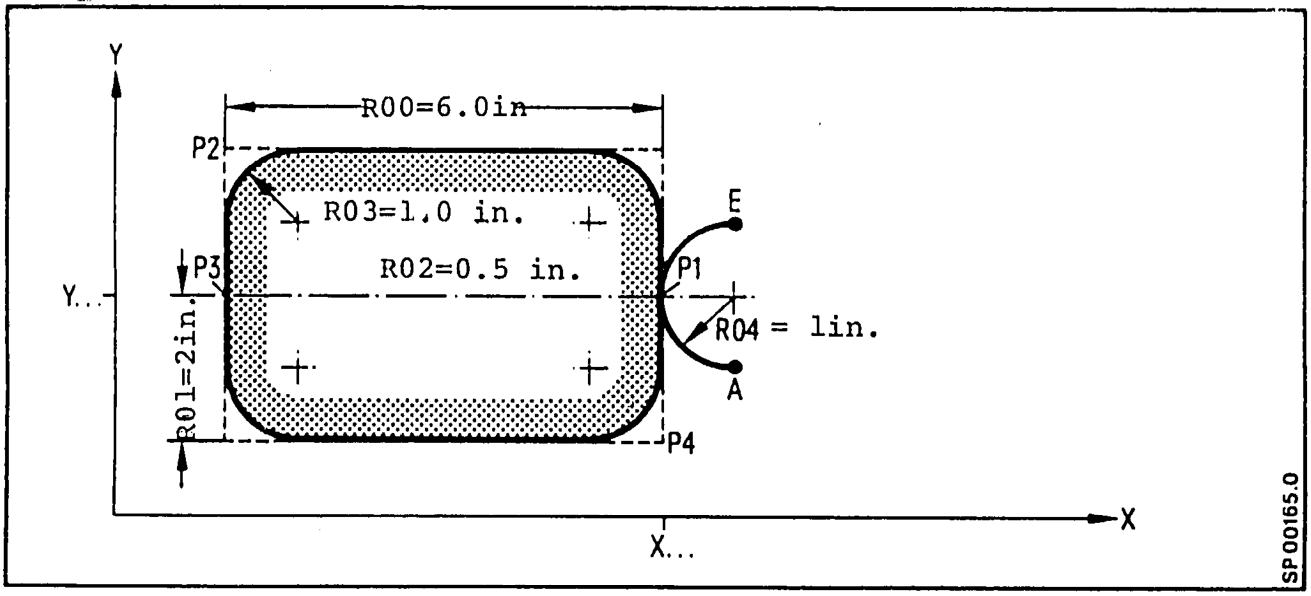
Example:



.
.
N10 G00 G90 X125. Y50. LF
N11 G01 A280. A260. X47. Y58. F . . . LF

Example: Rectangular Pattern

The following subroutine describes a rectangular pattern. The rectangle sides, corner radii as well as the depth advance is variable. The radius parameter R03 must be smaller than R01, i.e. one half of R00.



Example for 8M.

Subroutine:

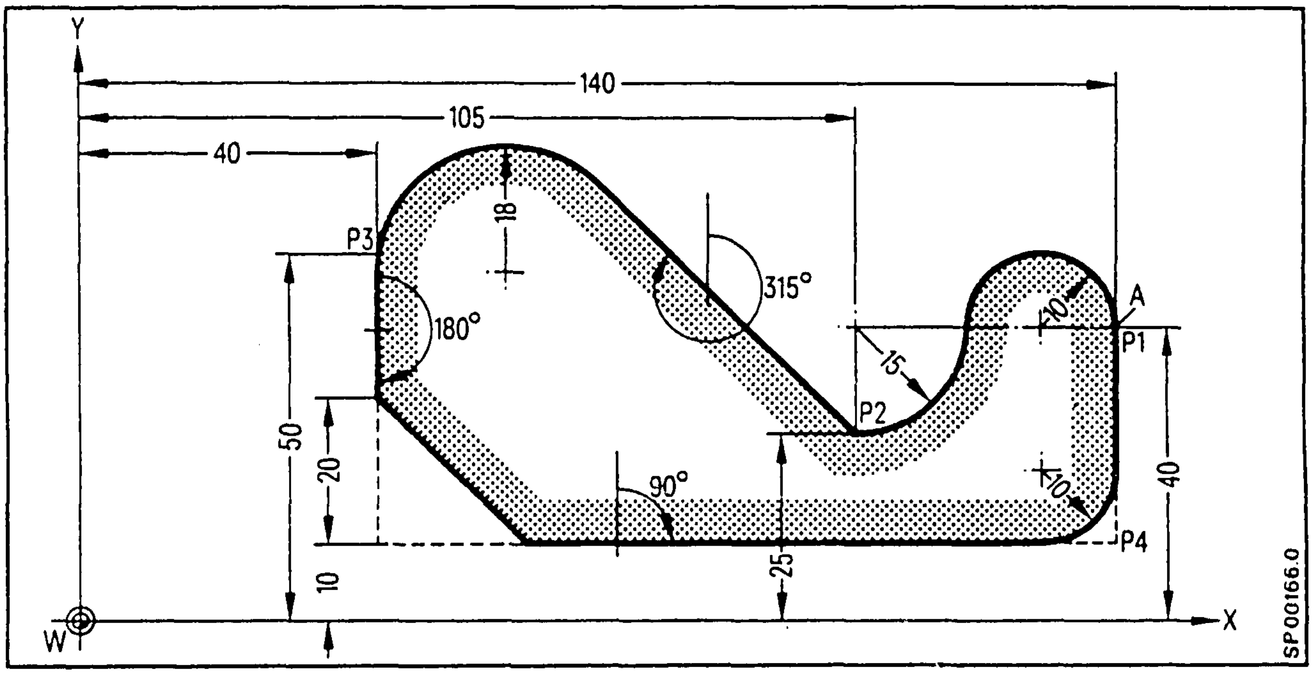
```
L46100
N0 G01 G91 Z-R02 LF                          (A)
N1 G02 X-R04 Y R04 P R04 LF                  (P1)
N2 G01 A0. A270. Y R01 X-R00 P R03 P R03  LF (P2)
N3 Y-R01 LF                                  (P3)
N4 A180.  A90. Y-R01 X R00 P R03 P R03  LF   (P4)
N5 Y R01  LF                                 (P1)
N6 G02 X R04 Y R04 P R04  LF                 (E)
N7 G01 Z R02  LF                             (E)
N8 M17  LF
```

Subroutine Call:

```
N25 G90    G42 D18 X... Y...
N30 L46101     R00 60. R0120. R025. R0310. R0410. LF
```

Example: Geometric Path Programming

In the following example, geometric path programming is used to program: an arc to arc path, a straight to arc path, and a three point connected path + chamfer + radius.



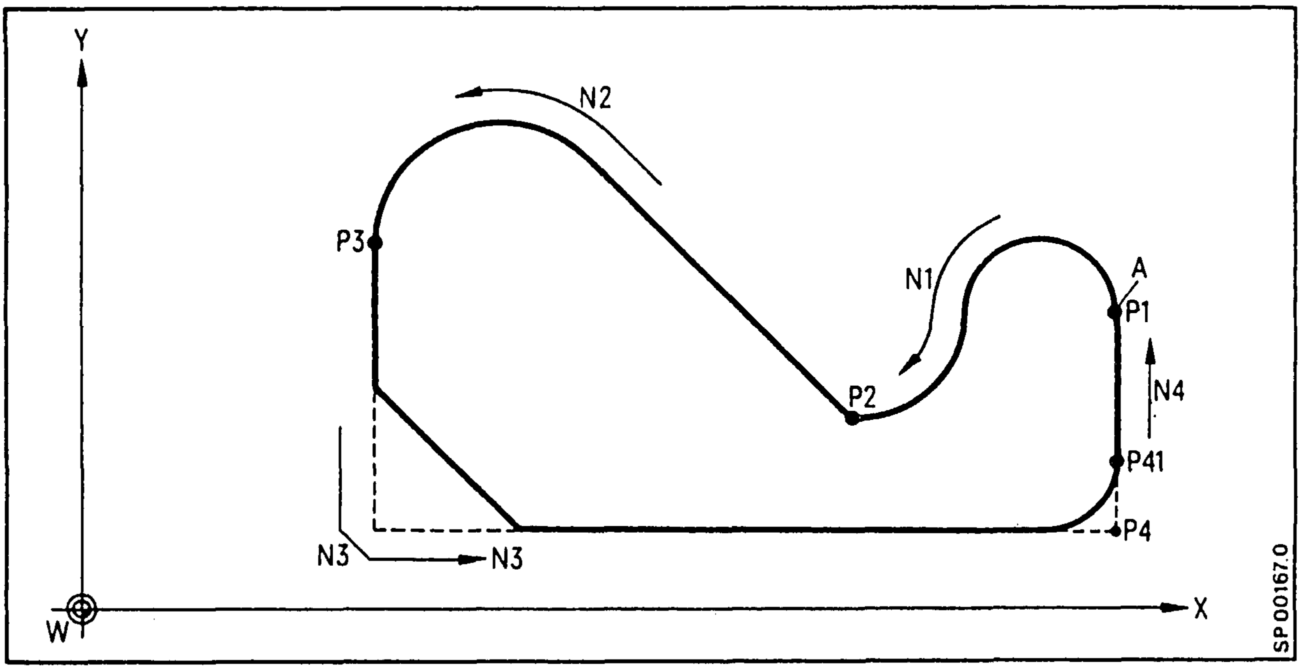

```
L16800
N1 G90 G03 I-10. J0. I0. J15. X105. Y25.      LF   (P2)   arc to arc path
N2     G03 A315. P18.X40. Y50.                LF   (P3)   straight to arc path
N3     G01 A180. A90. X140. Y10. P-20. P10.   LF   (P4)   3 point connected
                                                          path + chamfer +
                                                          radius
N4         Y40.                               LF   (P1)   straight path
N5 M17                                        LF
```

In block N2 G03 must be programmed. With an arc to arc path the second arc direction is opposite to the first. See "Geometric path programming", example 8.

7.10.5 Miscellaneous and Auxillary Fuctions in Linked Blocks

A block is considered linked to an adjacent block when a radius or chamfer is used to connect the two blocks.

Example:



A block containing miscellaneous and auxillary functions may be written between linked blocks.

Example: See above figure and page 7-26.

```
N3 A180. A90. X140. Y10. P-20. P10.  LF    (geometric path P3 - P4)
N4 M . . .    H . . .          . . . LF
N5 Y40.                              LF
```

The miscellaneous and auxillary functions become effective at point P4. A dwell mark will result at point P41.

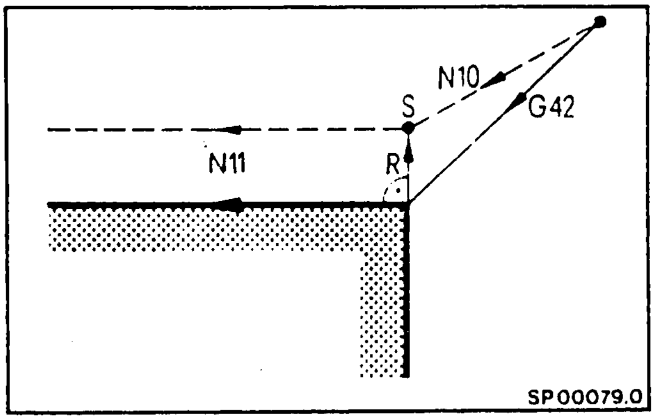
8. Appendix

8.1 Intersectional Cutter Radius Compensation

In the following all stop points are designated by an S.

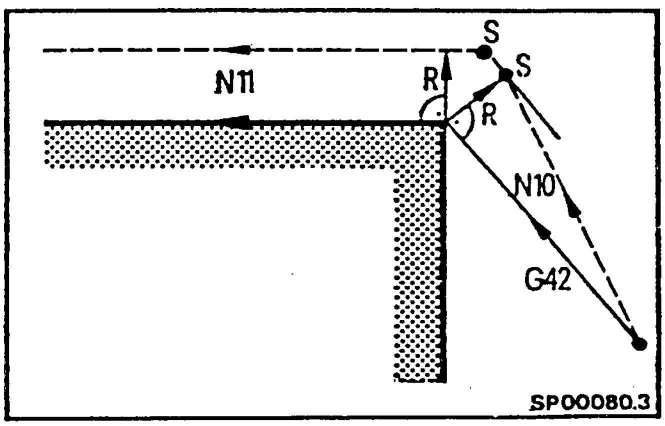
8.1.1 Selecting the CRC

- Inside contours (the included angle formed by blocks N10 and N11 is less than 180°).

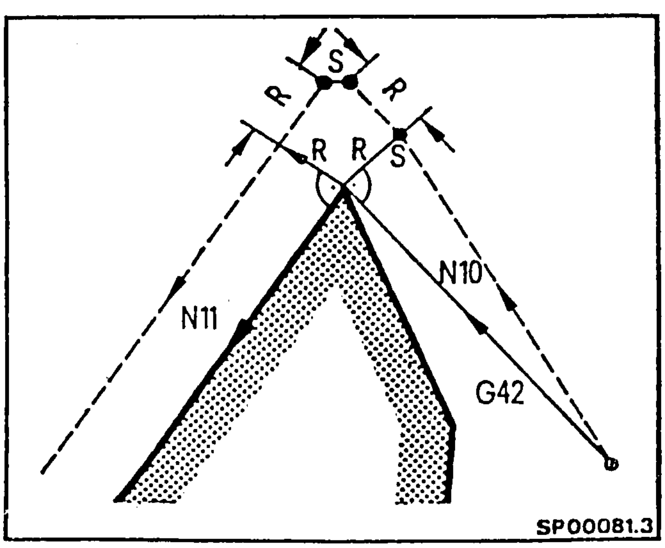


In a block following a block which selects the CRC, a vector of length R perpendicular to the programmed path is calculated.

- Outside contours (the included angle formed by blocks N10 and N11 is less than 270° and greater than 180°).



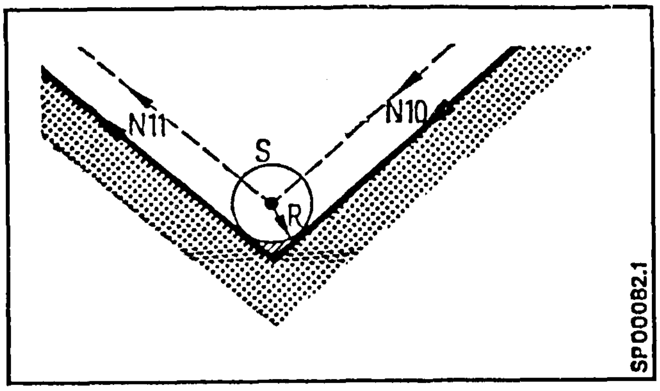
- Outside contours (the included angle formed by blocks N10 and N11 is greater than 270°).



8.1.2 CRC Used in a Program

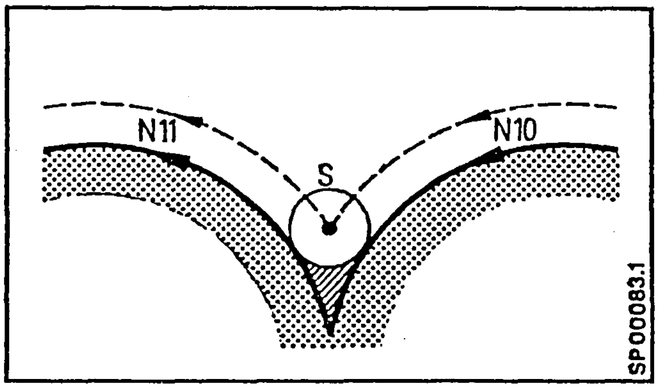
- Inside contour (the included angle formed by two blocks < 180°)

Linear to linear



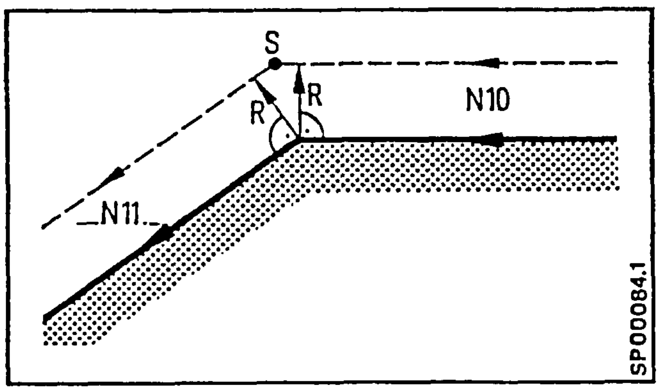
The intersectional point is calculated for the compensated path

Circular to circular



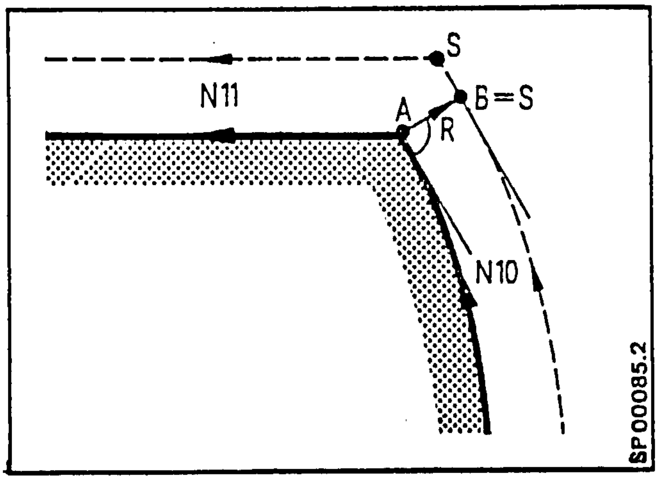
- Outer contour (the included angle formed by two blocks is less than 270° and greater than 180°)

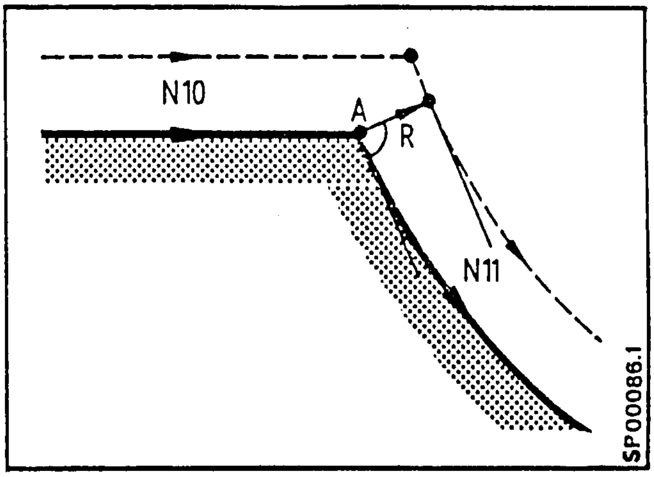
Linear to linear



The intersection point of the cutter compensation path is calculated.

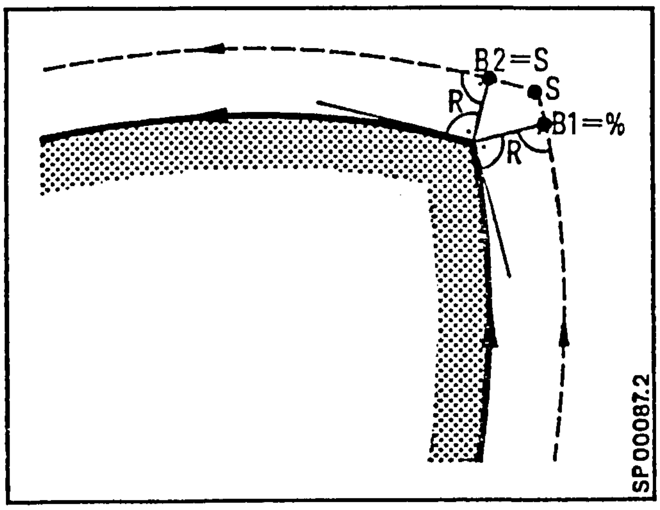
Linear to circular





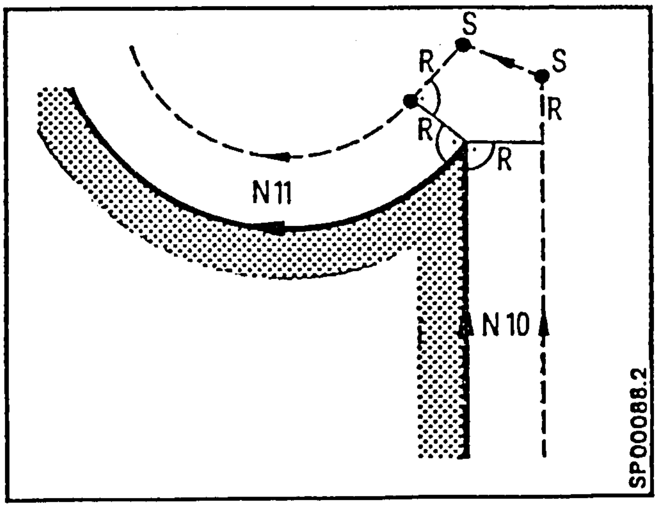
At the arc end point A (or arc start point), a normal of length R is calculated. The intersection point is calculated from the tangent at point B and the cutter compensated path of N11 (or N10).

Circular to circular



At the arc end point (or the arc start point), a normal vector of length R is created for both arcs. The tangent to point B2 and the tangent to point B1 is determined and the tangent-tangent intersection point calculated.

Linear to circular (the included angle is greater than 270°)

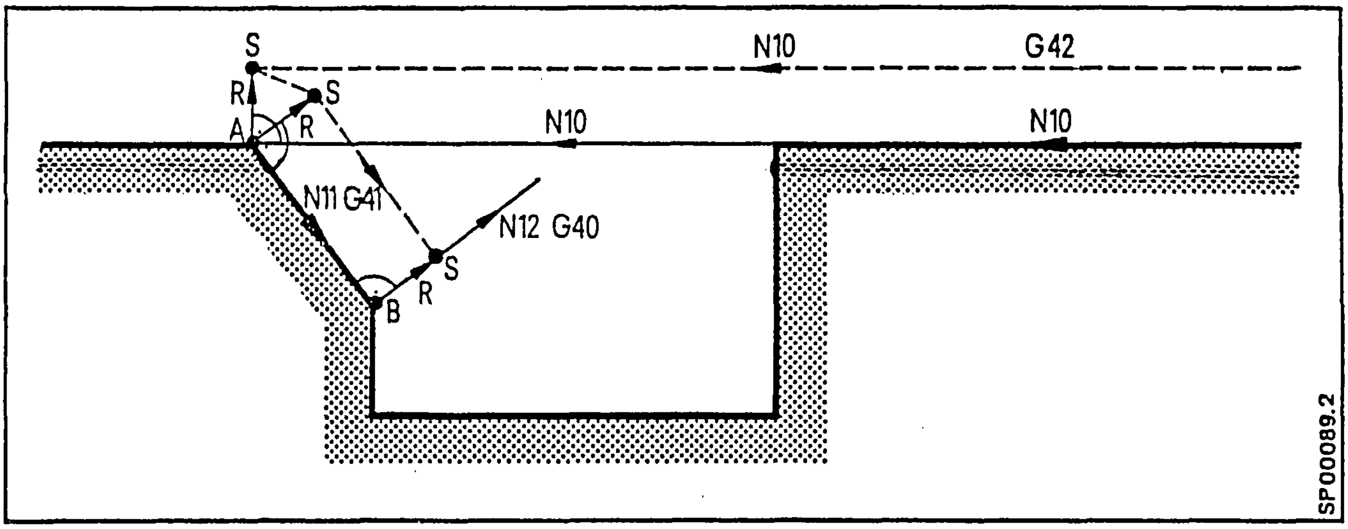


At the end point and start point respectively of blocks N10 and N11,a normal vector of length R is calculated for each path. The cutter will traverse a path that results when the two path endpoint tangents of length R are connected. The traversed path is the point connected path. The part contour is machined exactly.

Changing the Cutter Compensation Direction

At the block end point the old compensation direction (G41,G42) is changed to start the next block. The compensation direction is switched in the following manner:

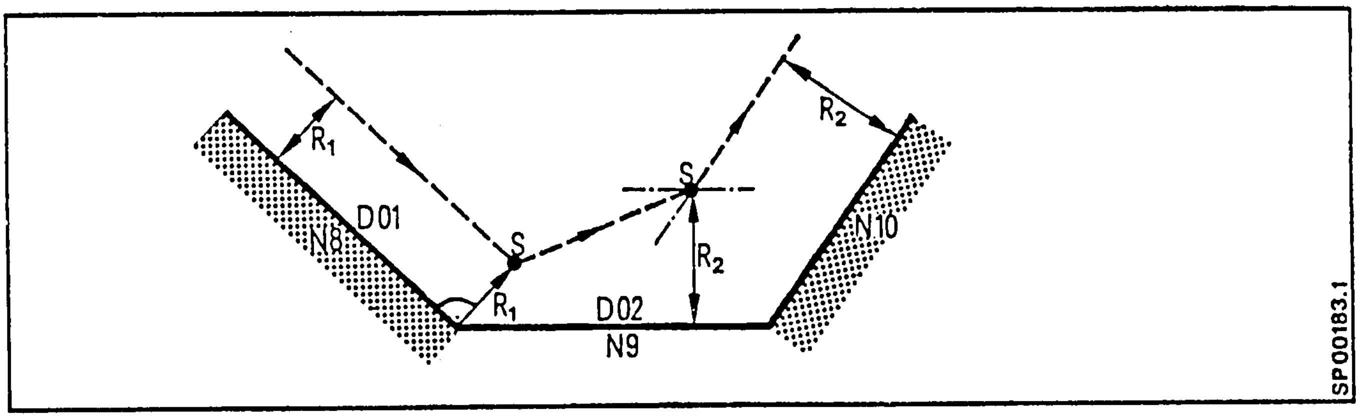
Normal vectors of length R are calculatied at the end point and start point of the new blocks respectively.



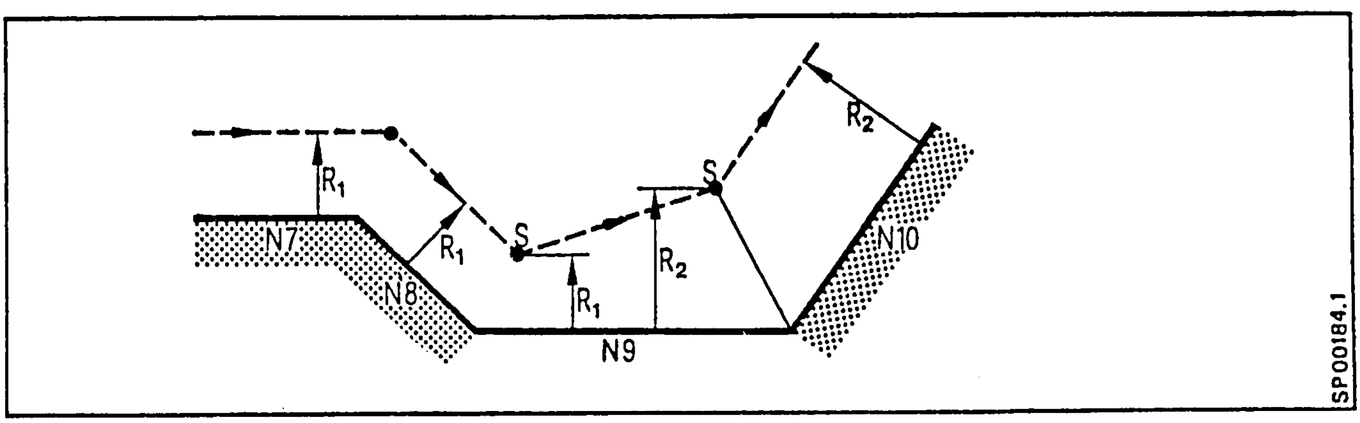
Calling a New Tool Offset Number (G41 D . . ., G41 D . . .)

When the tool offset number is changed, the following control action results:

No block start intersection point calculation is performed using the old compensation.
A nominal vector of length R1 is erected at the end point of the block containing the old offset.
The block end intersection point is calculated using the new offset.

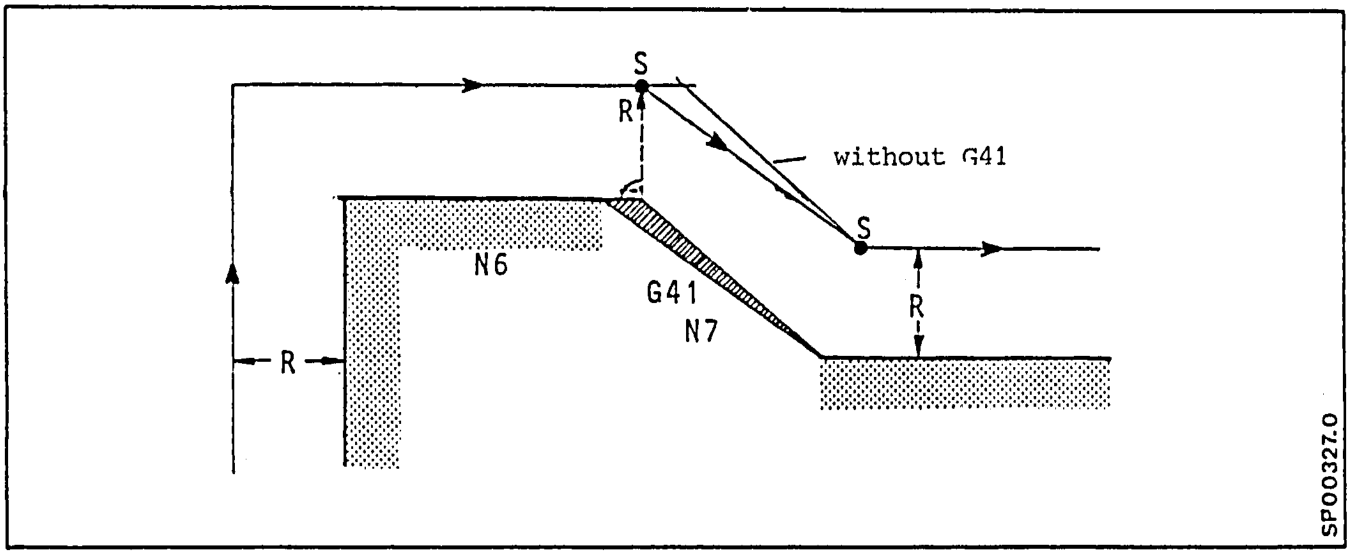


The tool offset dimensions can be changed from the operators panel, with a perforated tape, with the external tool offset or in the part program. The new offset is active in the next block.



8.1.3 Repeating the already selected G-code (G41,G42) with the same offset number (incorrect programming)

When an already programmed G41 or G42 is repeated, a normal vector of length R will be erected on the programmed path at the end of the previous block.



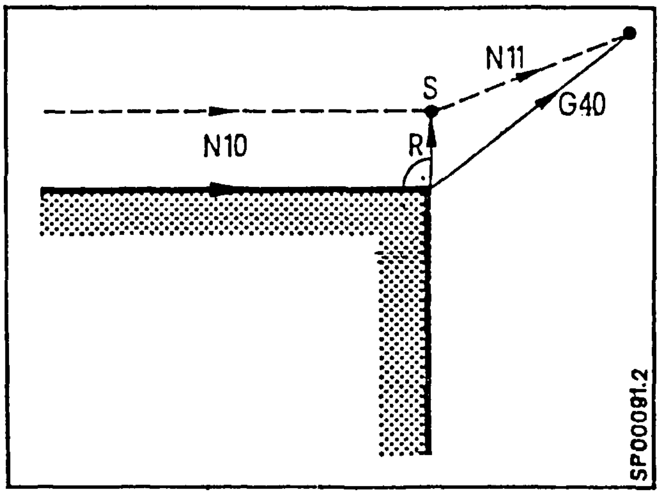
The block start intersection point is calculated for the following block:

```
N4   G91   D10   G41   X....          LF
N5                     X....          LF
N6                     Y....          LF
N7   G41               X....   Y....  LF
N8                     Y....          LF
```

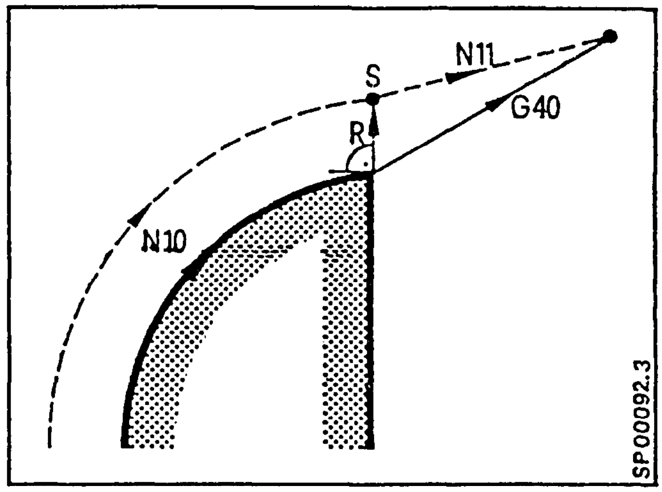
8.1.4 Cancelling CRC

- Inside contour (angle formed by block N10 and N11 is less than 180°)

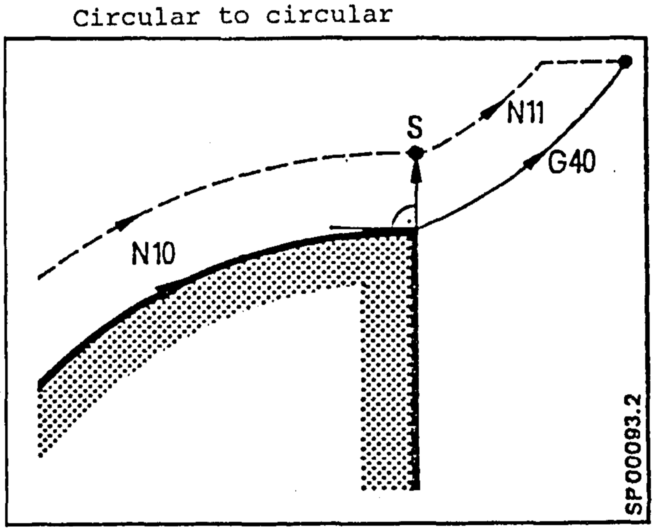
Linear to linear



Circular to linear



Circular to circular

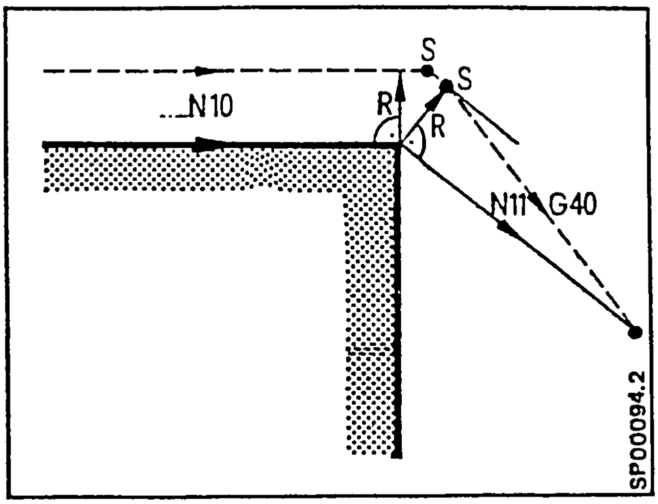


The last block in which CRC is active, a normal vector of length R is erected for the programmed path.

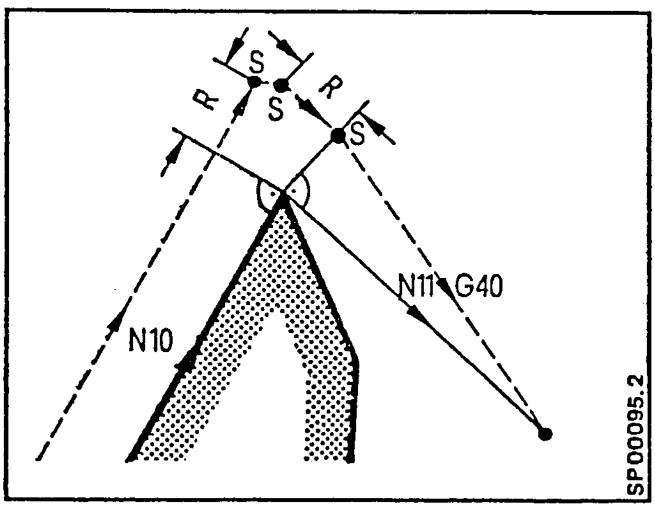
When a transition is made to a linear path, the programmed end point is approached directly.

When a transition is made to a circular path, a displaced arc path is traversed to the perpendicular intersectional point. The remaining distance is traversed along the perpendicular to the end point.

- Outside contours (angle formed by blocks N10 and N11 is less than 270° and greater than 180°)



(The included angle formed by blocks N10 and N11 is greater than 270°)



The compensated path is calculated,the tool traverses to the next calculated intersection point of the new block. CRC is cancelled.

8.1.5 M00, M01, M02, and M30 with CRC selected

M00, M01 — The NC stops at point S in the single block mode. (The positions are shown in the figure).

M02, M30 —
- The CRC is cancelled, if it is deleted with G40 and at least one axis is programmed

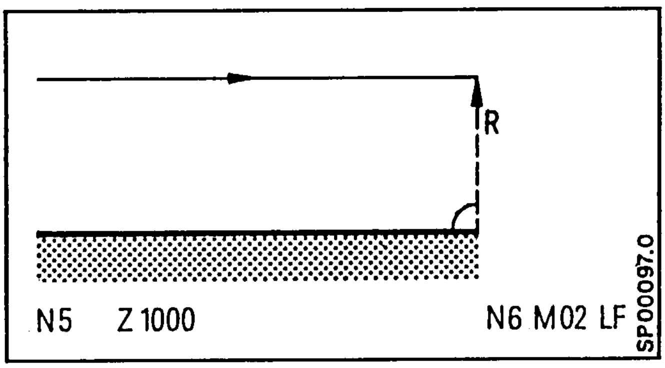
```
N150  X..  Y..        LF
N200  G40  X..  M30   LF
```

- The CRC is not cancelled (incorrect programming !)

```
N150  X..  Y..        LF
N200  G40             LF
N250  M30             LF
```

- Contour error with

```
N150  X..  Y..        LF
N200  M30  G40        LF
```



8.1.6 Combination of different types of blocks

The examples refer to the X-Y-plane.

Type: - Distances in the CRC-plane
Example:

N... G91 X1000 LF

- "Distance = 0"
Preparatory functions are programmed in the CRC plane, no movemnets take place, because the distance is zero.
Example:

N.. G91 X0 LF

- "Block without traverse information (auxillary block)
There are only movement addresses outside the CRC plane programmed , or only miscellaneous functions, dwell and block functions, subroutine definition, subroutine program, used alone in a block.
Example:

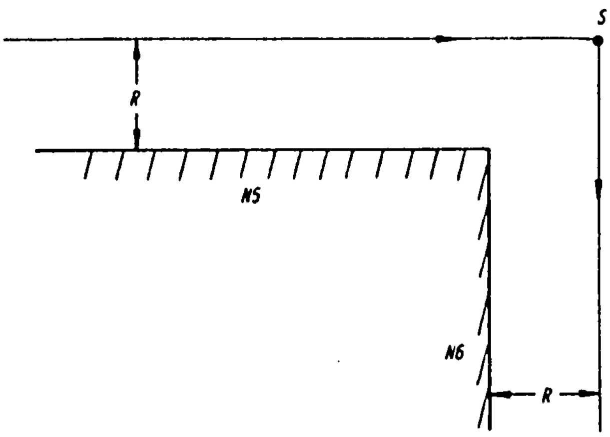
N... Y1000 LF
N... M08 LF
N... G04 X1000 LF

N... T0101 LF

- Not in the CRC-plane
Blocks that do not lie in the CRC-plane.
Example:

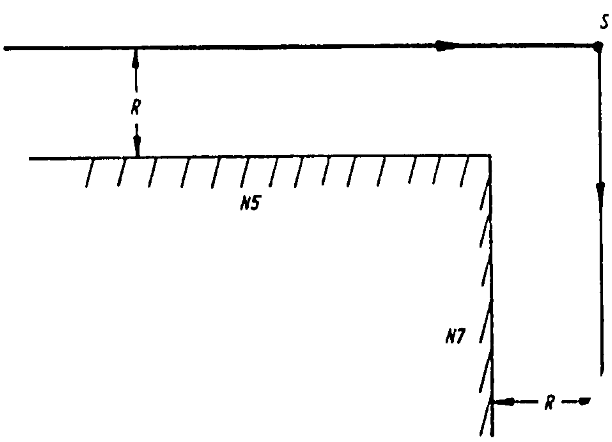
N... G02 X1000 Z1000 I0 I1000 LF

Two distances in the CRC-plane




```
N5 G91 X10000  LF
N6     Y-10000 LF
```

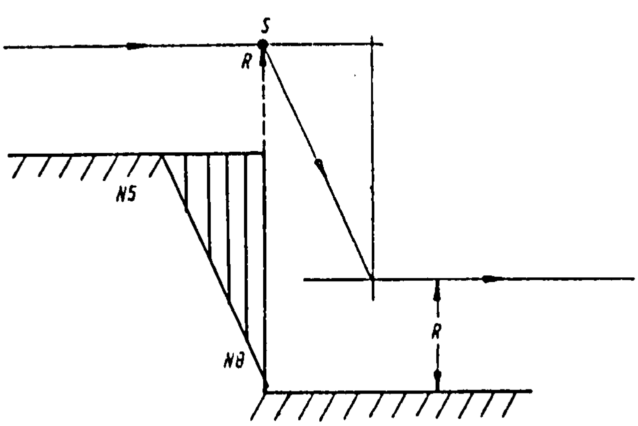
One "miscellaneous block" between distances in the CRC-plane




```
N5 G91 X1000   LF
N6 M08         LF
N7     Y-10000 LF
```

The block N6 is executed at point S.

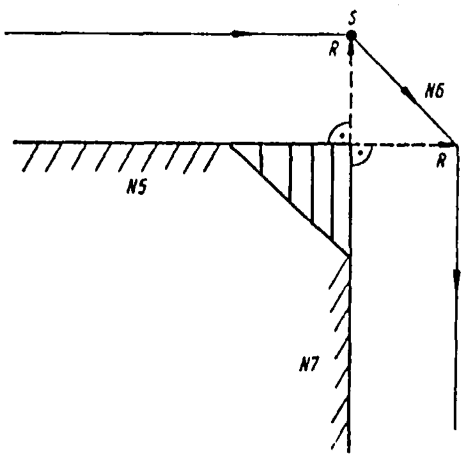
Two "miscellaneous blocks" between distances in the CRC-plane




```
N5 G91 X10000  LF
N6 M08         LF
N7 M09         LF
N8     Y-10000 LF
N9     X1000   LF
```

The blocks N6 and N7 are executed at point S.
With the exception of tangential transitions, contour adulterations result.

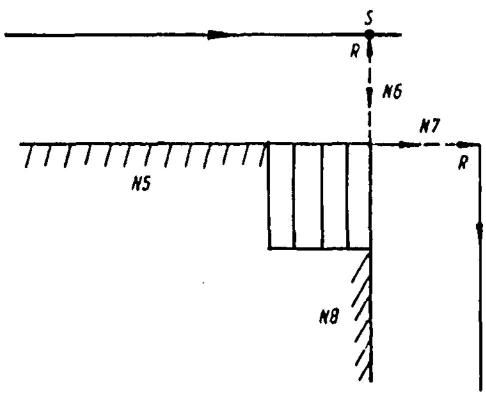
One block "distance = 0" between distances in the CRC-plane




```
N5 G91 X10000    LF
N6     X0        LF
N7     Y-10000   LF
```

With the exception of tangential transitions, contour adulterations result.

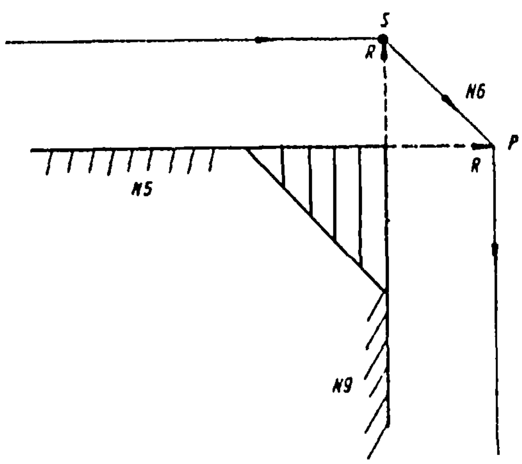
Two blocks "distance = 0" between distances in the CRC-plane




```
N5 G91 X10000    LF
N6     X0        LF
N7     X0        LF
N8     Y-10000   LF
```

With the exception of tangential transitions, contour adulterations result.

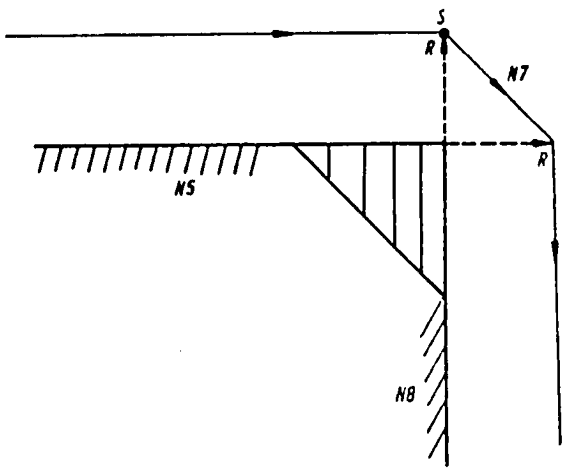
One block "distance = 0" and one "miscellaneous block" between distances in the CRC-plane




```
N5 G91 X10000    LF
N6     X0        LF
N7 M08           LF
N8     Y-10000   LF
```

The block N7 is executed at point P.
With the exception of tangetial transitions, contour adulterations result.

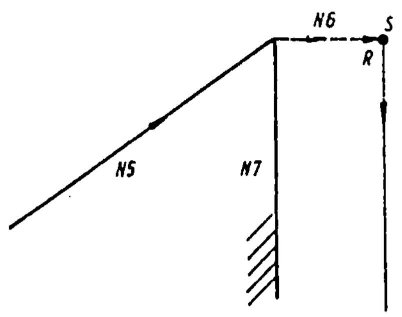
One "miscellaneous block" and one block "distance = 0" between distances in the CRC-plane




```
N5 G91 X10000    LF
N6 M08           LF
N7     X0        LF
N8     Y-10000   LF
```

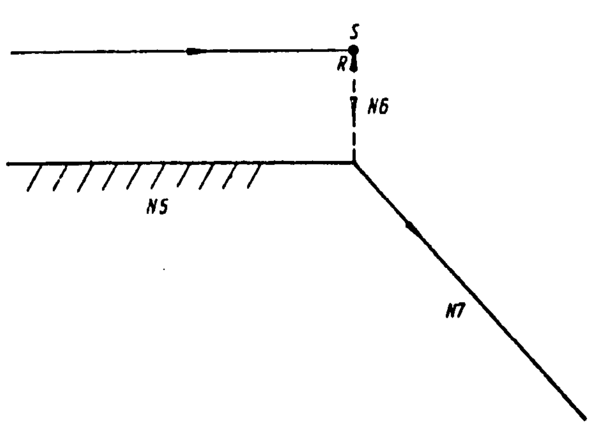
The block N6 is executed at point S.
With the exception of tangential transitions, contour adulteration results.

Selection of CRC in one block "distance = 0"



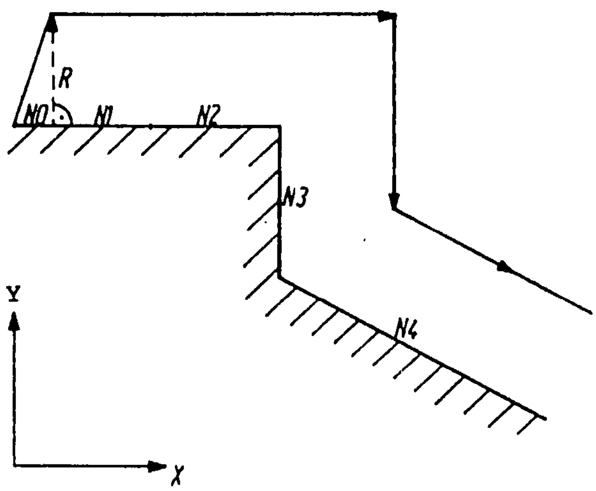

```
N5 G91 X10000 Y10000    LF
N6 G41 D01 X0 Y0        LF
N7            Y-10000   LF
```

Deletion of CRC in one block "distance = 0"




```
N5 G91 X10000            LF
N6 G40 X0                LF
N7     X10000 Y-10000    LF
```

One block "not in the CRC-plane" between distances in the CRC-plane



The figures are projections on the CRC-plane.

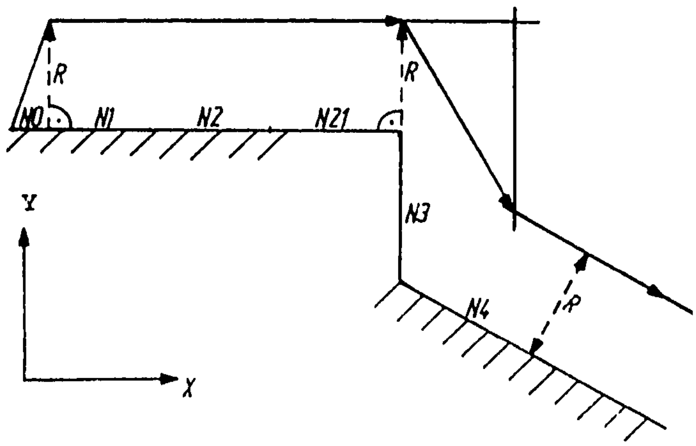
Valid for SP8M

N0 G17 G41 D1 G91 X10000 F10000
N1 X50000
*N2 G18[1)]G03 X50000 Z 50000 I50000 K0
N3 G17 G01 Y-50000
N4 X100000 Y-60000

Valid for 8M/8MC

N0 G41 D1 G91 X10000 Y0 F10000
N1 X50000
*N2 G03 Z50000 X50000 K0 I50000
N3 G01 Y-50000
N4 X100000 Y-60000

Two blocks "not in the CRC-plane" between distances in the CRC-plane



Valid for SP8M

N0 G17 G41 D1 G91 X10000 F10000
N1 X50000
*N2 G18[1)]G03 X50000 Z 50000 I50000 K0
*N21 X50000 Z-50000 I0 K-50000
N3 G17 G01 Y-50000
N4 X100000 Y-60000

Valid for 8M/8MC

N0 G41 D1 G91 X10000 Y0 F10000
N1 X50000
*N2 G03 Z50000 X50000 K0 50000 I50000
*N21 Z50000 X-50000 K50000 I0-50000
N3 G01-Y-50000
N4 X100000 Y-60000

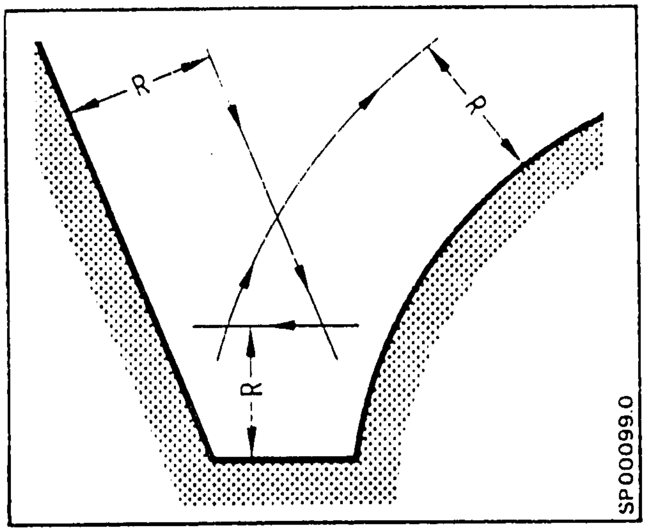
* Block not in the CRC plane

1) If G18 is missing then alarm 504 is displayed.

With the exception of tangential transitions, contour adulteration results.

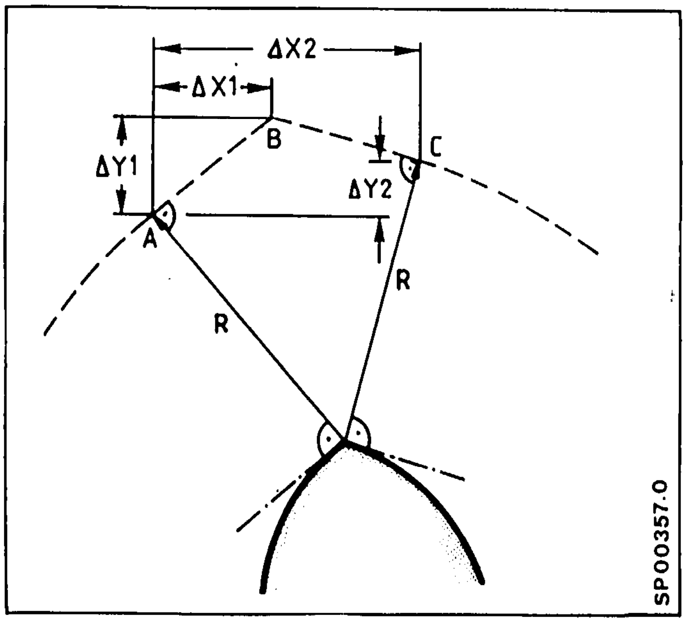
8.1.7 Special Cases when Using the CRC

Since the control always uses the information of the next block to calculate the intersectional path, a contour distortion will result under the following circumstances.

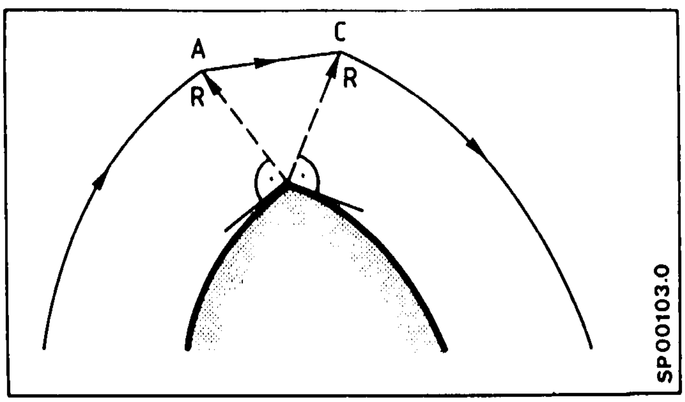


The tool offset dimension is larger than the distance between two paths. Machining is not interrupted, however, an alarm is signaled 506 and again cancelled at the end of the program.

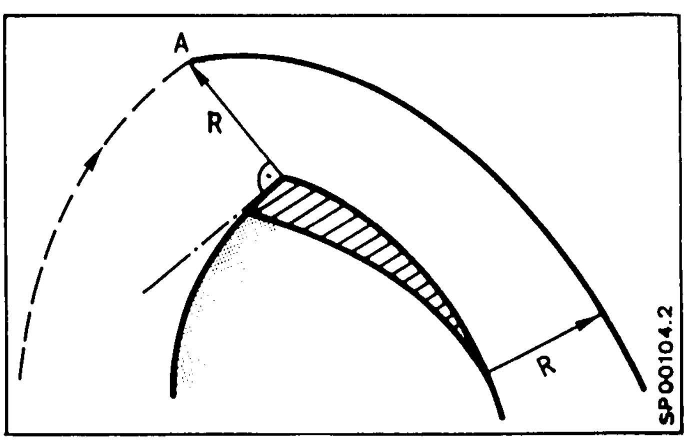
For external contours with an obtuse angle the following applies:



To avoid transition paths, generated by the control, that are of such short time duration that axis motion is temporarily halted, the distances AB and BC may be omitted by the NC. The path that results depends upon the tolerance set into machine parameter (maximum 32000 um) = d, during commissioning.

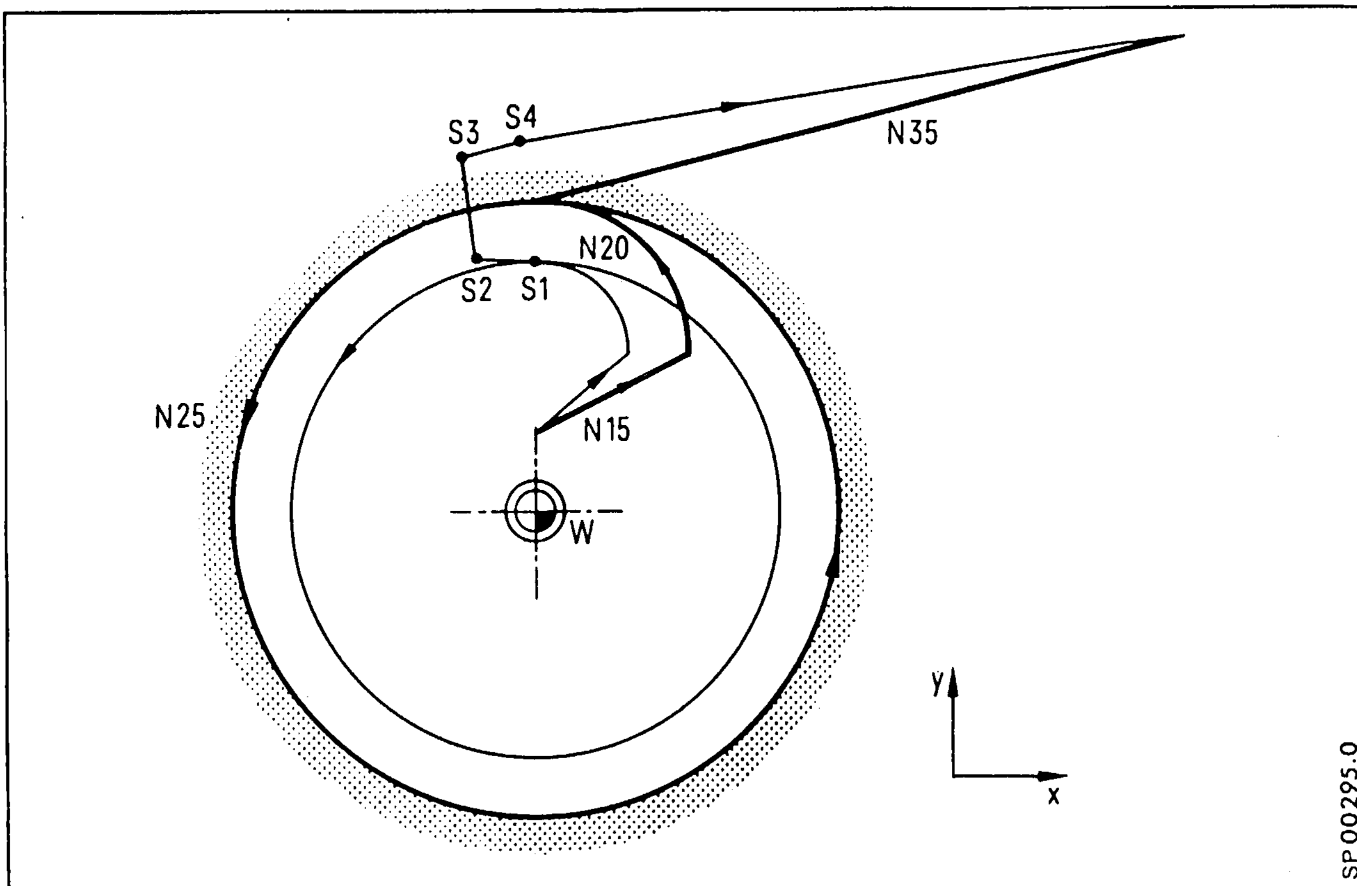


With X1 and Y1 less than d, the control moves directly from A to C.



With X1, Y1, X2 and Y2 smaller than d, there is no compensation movement generated. From point A machining continues with the new radius.

There are resultant exchanging of block numbers observable in the display when machining inside contours with acute angular contour transitions (programmed) and intermediate lying axis movements that do not lie in the CRC plane.
In order that the workpiece is not damaged the following procedure should be observed.




```
N5  G00 Z100.
N10 X0. Y10.
N15 G41 D01 X20. Y20.
N20 G03 X0. Y40. I-20. J0.
N25     X0. Y40. I0. J-40.
N30 G01 Z0.
N35 G40 X80. Y60.
```

The points S1, S2, S3, S4 belong logically to block N25. The machining sequence (observable in single block) is :
..., N20, N25 (S1), N30 (withdrawal of the tool from the workpiece), N25 (S2), N25 (S3), N25 (S4), N35... .
The same procedure applies when N25 is a linear block.

8.2 Input System, Diagrams, and Tables

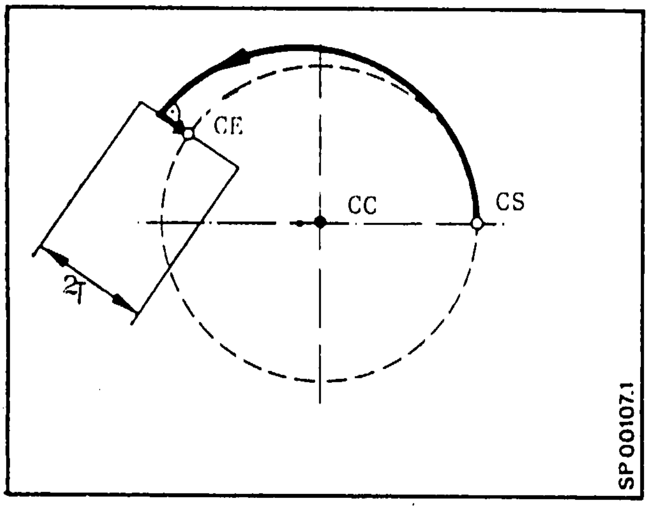
8.2.1 Inaccurately specifying the interpolation parameters or the arc radius

An arc end point programming error is recognized by the control (assuming the tolerance window is exceeded). Circular interpolation will not begin, instead, an alarm is signaled.

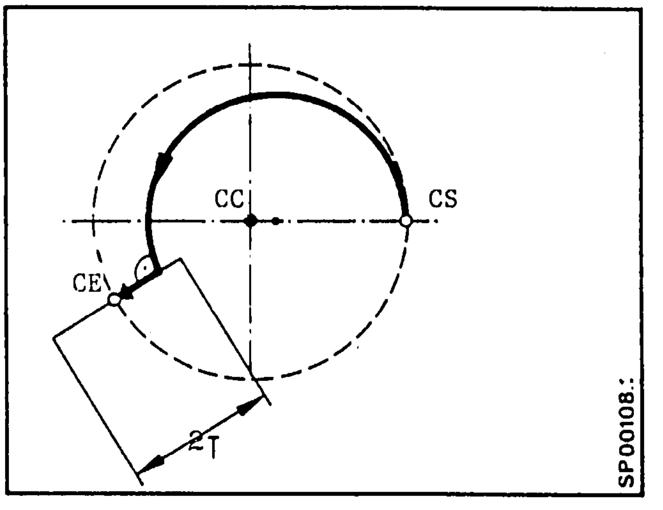
If the programming error lies within the arc tolerance window, the control will position accurately to the end point, however, the path will deviate from the desired arc as shown:

Interpolation Parameter or Radius

too large



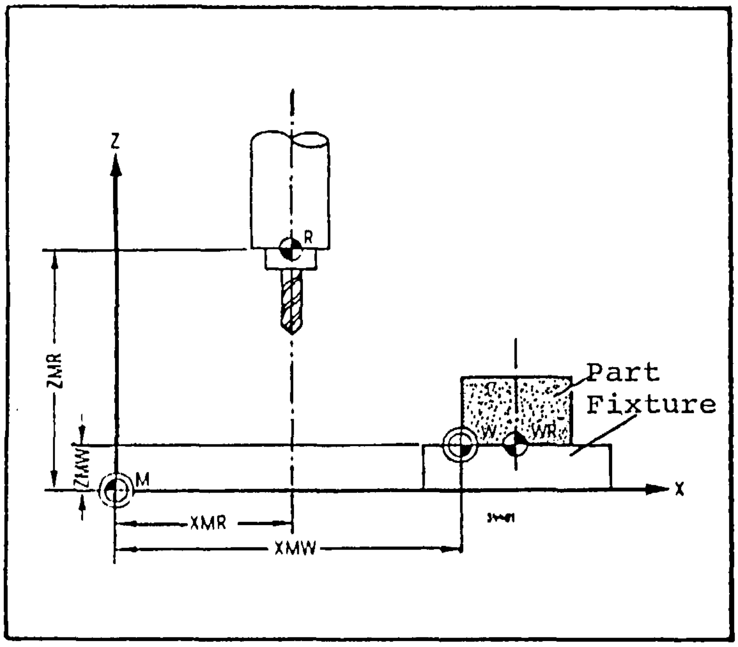
too small



The tolerance window T about the arc end point CE is adjustable from ±0.0001 to ±3.2 (±1 µm to ±32000 µm).

The monitor can be suppressed by setting a large dimensional value. The tolerance window is input as an unsigned dimension stored under a machine parameter address.

8.2.2 Reference Point Definitions



M = Machine zero point

W = Part zero point

R = Machine home position

WR = Part reference point

XMR, ZMR etc. = Reference point coordinates for each axis

XMW, ZMW etc. = Sum of all null offsets for each axis

Total Null offset = settable offset (G54..G57) + additive null offset (G59) + ext. null offset + additive ext. null offset.

8.2.3 Path calculation

G91 in the first block with motion

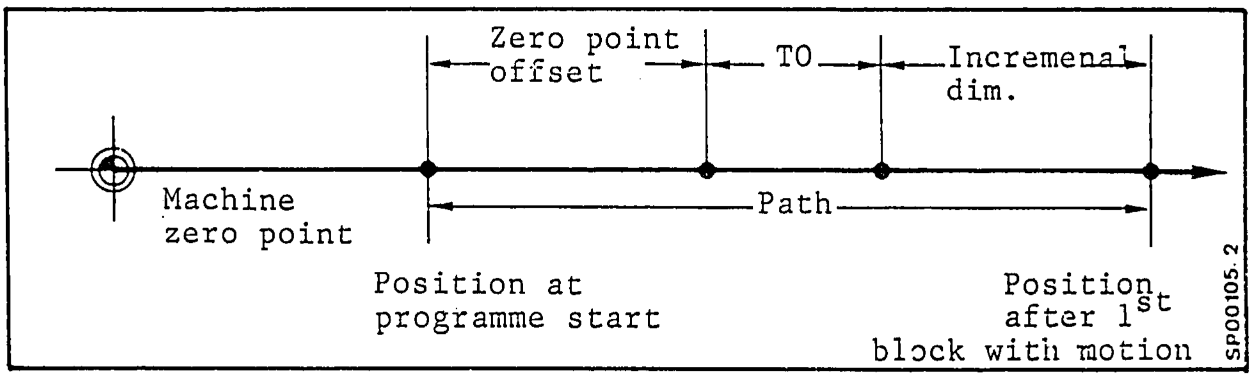
Path = Incremental dim. + ZO + TO

G91 from the second block with a motion

$$\text{Path} = \text{incremental dim.} + ZO_{(new)} - ZO_{(old)} + TO_{(new)} - TO_{(old)}$$

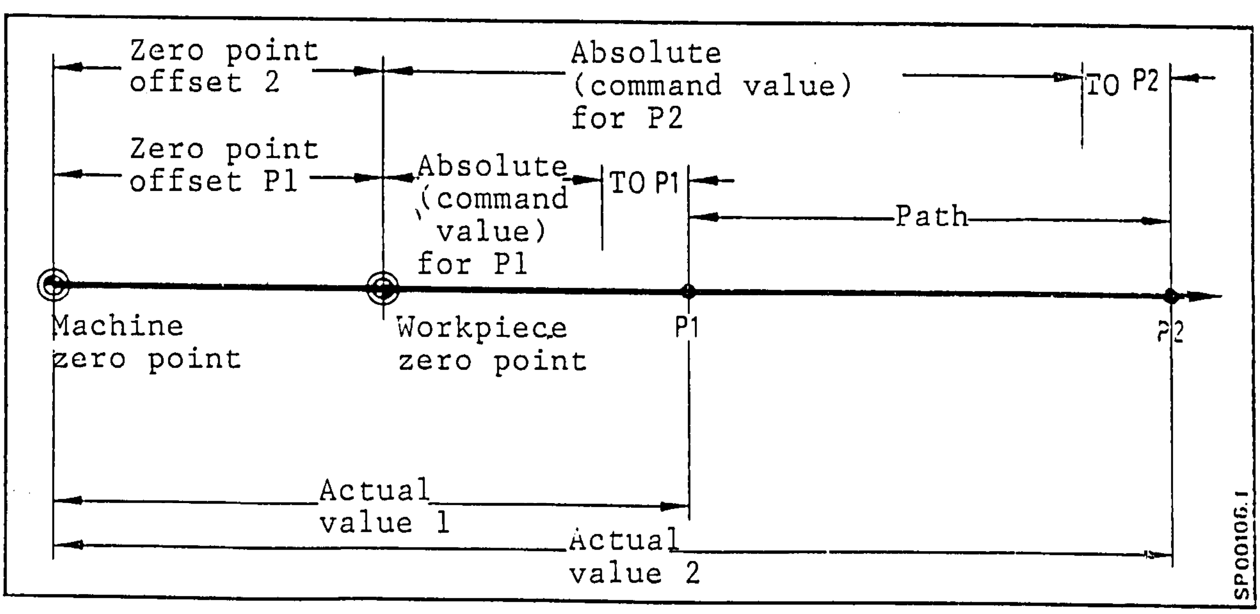
When the ZO and TO are not changed, the formula is simply:

Path = Incremental dimension

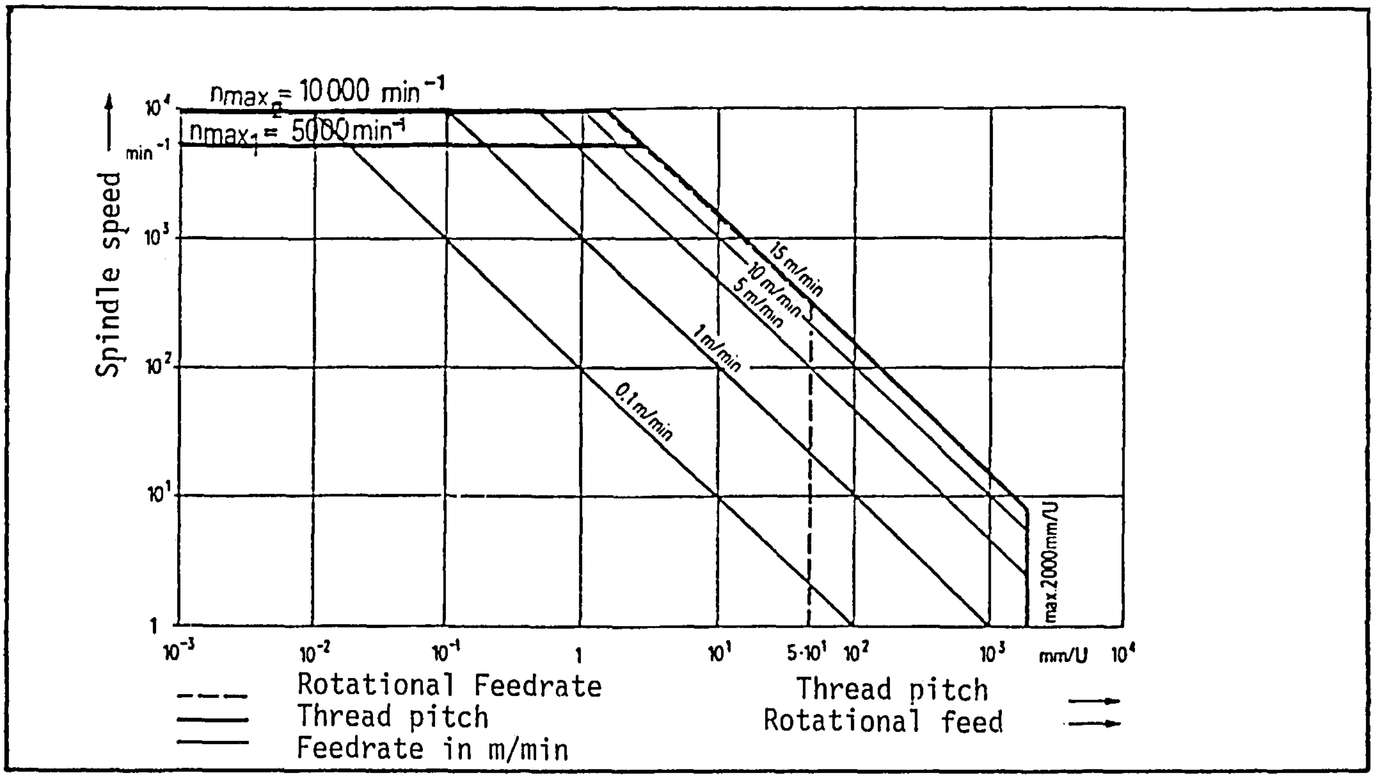


G90 in any block with a motion

$$\text{Path} = \text{Absolute dim.}_{(new)} - \text{absolute dim.}_{(old)} + ZO_{(new)} - ZO_{(old)} + TO_{(new)} - TO_{(old)}$$



8.2.4 Limit data for ratational feedrate

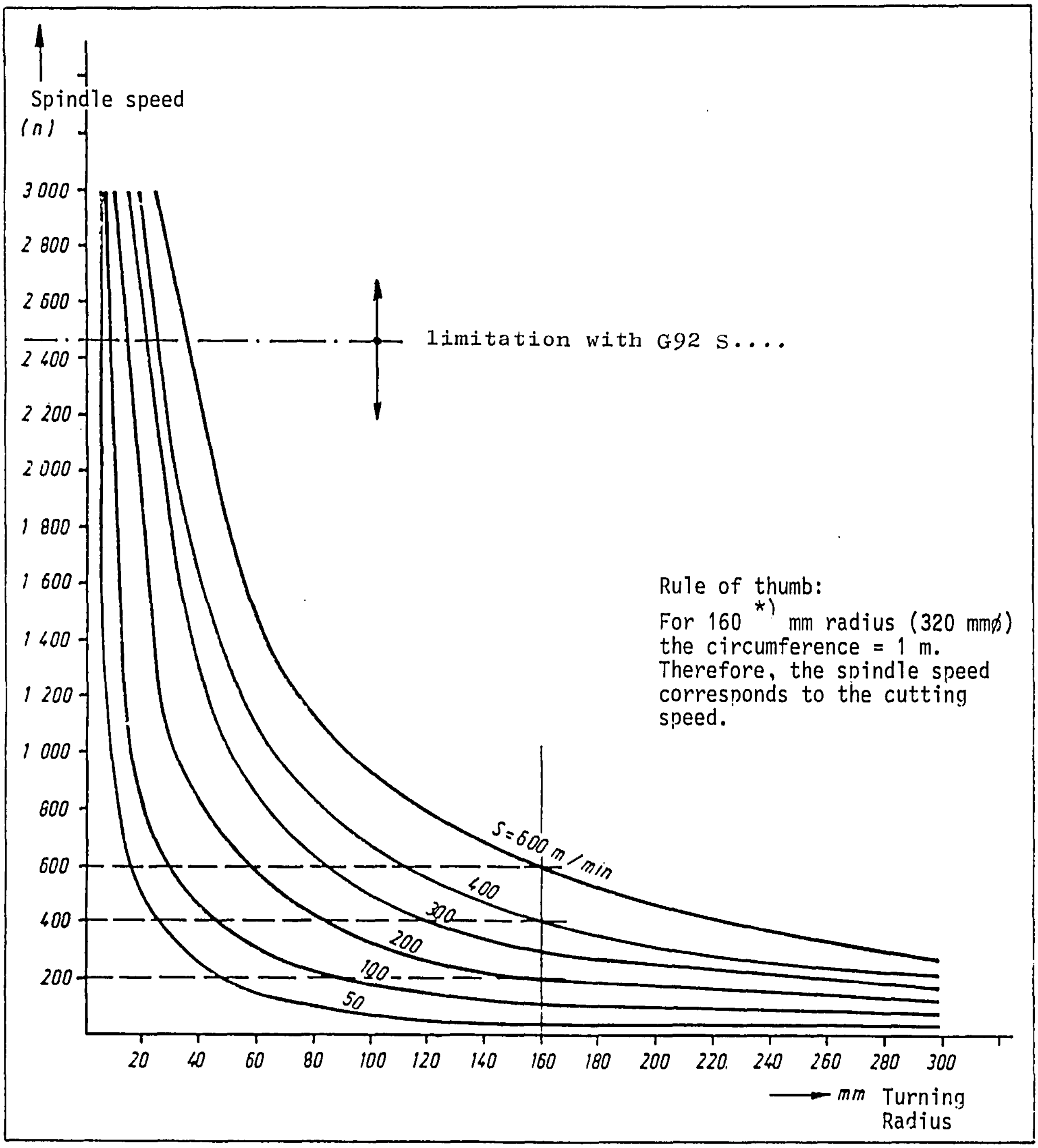


Relationship between rotational feedrate and spindle speed

Relationship between pitch and spindle speed (thread cutting G33)

n $max._1$ can be achieved with ROD encoder connected 1:1
N $max._2$ can be achieved with ROD encoder connected 1:2

8.2.5 Spindle speed as a function of turning radius for V = constant



*) exactly 159.164 mm

8.2.6 Input Formats

Address definition	Metric Decades	Metric smallest increment	Inch Decades	Inch smallest increment	Inch Decades	Inch smallest increment	Degrees Decades	Degrees smallest increment
Path data (linear axes) Interpolation parameter Radius	± 5.3		± 4.4		± 3.5		-	
Path data (rotary axes)	-		-		-		± 5.3	
Chamfer	- 5.3	10^{-3} mm	- 4.4	10^{-4} inch	- 3.5	10^{-5} inch	-	10^{-3} degrees
Work area limitation Zero point offset	± 5.3		± 4.4		± 3.5		± 5.3	
Thread pitch	4.3		3.4		2.5		-	
Cutting speed S (Weighting factor set during commissioning) Spindle speed S	4.0	1 or 0.1 m/min 1 or 0.1 min^{-1}	4.0	1 or 0.1 Ft/min 1 or 0.1 min^{-1}	4.0	1 or 0.1 Ft/min 1 or 0.1 min^{-1}		
Linear feedrate (F)	5.0	mm/min	4.1	10^{-1} inch/min	4.1	10^{-1} inch/min	5.0	degrees/min
Thread pitch increase or decrease F Rotary feedrate (F)	2.3	10^{-3} mm/rev	1.4	10^{-4} inch/rev	1.4	10^{-4} inch/rev		
Tool offset – Lgth add. lgth	± 4.3		± 3.4		± 2.5 Inch			
Tool offset – Radius	± 3.3	10^{-3} mm	± 2.4	10^{-4}	± 1.5	10^{-5}		
Tool offset – Wear	± 1.3		± 0.4		± 0.04			
Tool offset – Form	1		1		1			
Factor (G92)	3.5		3.5		3.5			
Dwell X	5.3	10^{-3} sec	5.3	10^{-3} sec	5.3	10^{-3} sec		
Dwell F	2.3		2.3		2.3			
Angle							3.5	10^{-5} degree
Angle for oriented spindle stop							3.0	1 degree
R - Parameter	Dimension depends on associated (internal floating point) all combinations (2 decades for call-up)							
G. prep. function	2		2		2			
M. functions	2		2		2			
H. functions	1 to 6		1 to 6		1 to 6			
Block number	1 to 4		1 to 4		1 to 4			
Special functions @	2		2		2			

Using inch input (G70) the smallest input increment can be changed from 10^{-4} inch to 10^{-5} inch by modification of setting datum.

The parameters (R00-R99) and special functions @ 00 - @ 99 are always written as 2 decades. For all other functions (except address L) the leading zeros can be omitted.

8.2.7 Axis Numbers

Axis Number

	8M/Sprint 8M	8MC
X-Axis	1	1
Y-Axis	2	2
Z-Axis	3	3
4th Axis	4	4
5th Axis	-	5
6th Axis	-	6
7th Axis	-	7
8th Axis	-	8
9th Axis	-	9
10th Axis	-	10

8.2.8 Boring cycles - axis switchable

```
% SP
L8100(8.3.82EVAR.3)R77 0 R59 0
@25
@00 R77
N0 R59 87
N1 R59 R11                       Address parameter, dependent on the stage of
@20 @92 R59                      development of the software
G G60 G90 @92 R02                Rapid traverse on the reference plane
G1 @92 R03                       Boring on reaching programmed depth
G @92 R02 M17                    Rapid traverse back
L8200 R77 0 R59 0
@25
@00 R77
N0 R59 87
N1 R59 R11
@20 @92 R59
G G60 G90 @92 R02                Rapid traverse on reference plane
G1 @92 R03                       Boring on reaching  programmed depth
G4 FR04                          Dwell on programmed depth
G @92 R02 M17                    Rapid traverse back
L8300 R59 0 R77 0 R65 0 R67 2    Starting preparations
@25 R64 0 R01 R63 0 R02 R66 1
@00 R77
N0 R59 87
N1 R59 R11
@20 @92 R59
R67 . R05                        R67 = 2 x Degression
R63 -- R03
@03 2 R63 R65                    Call-up of boring direction
R66 --1                          R66 = sign
N2 R63 . R66
R78 . R66
G G60 G90 @92 R02                Driving reference plane
N3 R63 - R64
@03 4 R65 R63                    Recognition of end R63 <= 0
R63 . R66 R62 0 R03
R62 R63                          R62 = Boring depth absolute
G1 @92 R62
G4 FR04
G @92 R02
G4 FR00
R62 R78                          Driving on security
@92 R62 -- R78                   For next boring
R63 . R66
@03 4 R05 R63                    R63 <= degression => end
R64 -- R05
@02 -3 R64 R05                   Calculate next move
R64 0 R05

R64 / R62                        Bisection of move
@00 --3
N4 G1 @92 R03                    Boring last move
G4 FR04
G @92 R02 M17                    End
```

```
L8400 R77 0 R59 0
@25
@00 R77
N0 R59 87
N1 R59 R11
@20 @92 R59
G G60 G90 @92 R02          Rapid traverse on reference plane
G1 G63 @92 R03             Thread cutting on reaching programmed depth G63
MR06                       Spindle reverse
@92 R02                    Back with G63
G G60 MR07 M17             Basic position
L8500 R77 0 R59 0
@25
@00 R77
N0 R59 87
N1 R59 R11
@20 @92 R59
G G60 G90 @92 R02          Rapid traverse on reference plane
G1 @92 R03                 Boring on reaching programmed depth
G @92 R10 M17              Rapid traverse on retract R10
L8600 R77 0 R59 0
@25
@00 R77
N0 R59 87
N1 R59 R11
@20 @92 R59
MR07                       Direction of spindle rotation R07
G G60 G90 @92 R02          Rapid traverse on reference plane
G1 @92 R03                 Boring on reaching programmed depth
M5                         Spindle stop
G @92 R10 M17              Rapid traverse back on retract R10
L8700 R77 0 R59 0
@25
@00 R77
N0 R59 87
N1 R59 R11
@20 @92 R59
MR07                       Direction of spindle rotation R07
G G60 G90 @92 R02          Rapid traverse on reference plane
G1 @92 R03                 Boring on reaching programmed depth
M M5                       Spindle stop and program stop M00
G @92 R02 M17              Rapid traverse back on reference plane
```

```
L8800 R77 0 R59 0
@25
@00 R77
N0 R59 87
N1 R59 R11
@20 @92 R59
MR07                            Direction of spindle rotation R07
G G60 G90 @92 R02               Rapid traverse on reference plane
G1 @92 R03                      Boring on reaching programmed depth
G4 FR04                         Dwell before spindle stop
M M5                            Spindle stop and program stop
G @92 R02 M17                   Rapid traverse back on reference plane
L8900 R77 0 R59 0
@25
@00 R77
N0 R59 87
N1 R59 R11
@20 @92 R59
G G60 G90 @92 R02               Rapid traverse on reference plane
G1 @92 R03                      Boring on reaching programmed depth
G4 FR04                         Dwell
@92 R02                         Feedrate back
G M17                           Basic position
L9000 R77 0 R59 0
@25
@00 R77
N0 R59 87
N1 R59 R11
@20 @92 R59
G G60 G90 @92 R02               Rapid traverse on reference plane
@02 3 R11 R77
@00 R11
N1 G33 @92 R03 IR09             Thread cutting on reaching programmed depth G33
@92 R02 IR09 MR06               Spindle reverse, back with G33
@00 5
N2 G33 @92 R03 JR09
@92 R02 JR09 MR06
@00 5
N3 G33 @92 R03 KR09
@92 R02 KR09 MR06
N5 G MR07 M17                   Basic position
M02
```

8.2.9. Special case with "cancel distamce to go"

Cancelling the distance to go and then continuing with incremental programming leads to incorrect positioning since the programmed incremental movement is added to the old programmed value and is thereby active.
That means that the the approached actual position is incorrect by the amount of the cancelled distance to move.

Error case:

G90 G01 X100
- Cancel distance to go with X60

G00 G91 X60
- New actual position X160

Remedy:

@ 24 X - load actual position into R93 using @ 24
G90 X R93 - move to actual position

e.g.:
G90 G01 X100
- Cancel distance to go with X60

@ 24 X
G90 XR93
G00 G91 X60 new position X120

After "cancel distance to go" a G90 or G92(set actual value) block must be written for the "cancelled axis" !

8.2.10 Block preparation time

The block preparation time is that time that the contol requires in order to declare the block the current working block.
With the 8M/8MC and Sprint 8M the block preparation times are approximately: 80 ms without CRC } working from memory
100 ms with CRC }

When working from the tape reader, one should add a maximum of 4 ms per character additionally.

In order to avoid free cutting (i.e. feedrate interruptions on the contour), the distance to be travelled per block must be so selected that the block preparation time is always exceeded.
The following applies:

t must be larger or equal to $\frac{s}{v}$

s = distance (mm)
v = speed (m/min)
t = 80 or 100 (ms)

Since the control always works some 4-8 blocks in advance, then individual blocks should also be less than the block preparation time.

Programming Key - 8M (continued)

Group	EIA	ISO	Code	Chapter	Function and description
	x	X	0.001 to ±99999.999	2.1	Position data in mm
			0.001 to +99999.999	3.12	Dwell in sec.
	y	Y	0.001 to ±99999.999	2.1	Position data in mm
	z	Z	0.001 to ±99999.999	2.1	Position data in mm
	4.th axis	4.th axis	0.001 to ±99999.999	2.1	Position data in mm or degrees; possible addresses A,B,C,U,V,W,Q,E,P,H
	a	A +)	0 to 359.99999	3.4	Angle in degrees for polar coordinates
	p	P +)	+0.001 to +99999.999 ±0.001 to ±99999.999	3.4 3.5	Radius at polar coordinates in mm Radius for circular interpolation in mm
	i	I	0.001 to ±99999.999	2.2	Interpolation parameter for X-axis in mm
			1 to 2000.000	3.6	Thread lead in mm
	j	J	0.001 to ±99999.999	2.2	Interpolation parameter for Y-axis in mm
			1 to 2000.000	3.6	Thread lead in mm
	k	K	0.001 to ±99999.999	2.2	Interpolation parameter for Z-axis in mm
			1 to 2000.000	3.6	Thread lead in mm
	d	D	00 1 to 199	3.23 3.24	Cancellation of tool compensation Tool compensation number
	r	R	00 to 49	5.0	Parameter
	f	F	0.001 to 15.000	3.17	Feed mm/min (Inch/min see 8.2.6)
			0.001 to 99.999	3.12	Dwell in sec.
			0.001 to 50.000	3.17	Feed in mm/rev. (Inch/min see 8.2.6)
	s	S	1 to 9999	4.1 3.18	Spindle speed in rev./min or 0.1 rev./min or constant surface speed in m/min or 0.1 m/min
			1 to 9999	3.19/3.20	Spindle speed limitation in rev./min or 0.1 rev./min
			0 to 359	4.4	Spindle stop in degrees, distance from zero mark
	t	T	1 to 999999	4.3	Tool number (tool position)
			0,1,2,3,4,5	3.7	Time constants
	h	H +)	1 to 999999	4.2	Auxiliary function
	l	L	001.. to 999..	1.8	Sub-routine number
			...01 to ...99	1.9/5.5	Number of runs of sub-routine
			999	5.7	Lock buffer read in
M1	m	M	■ 00 01	4.4 4.4	Programmed stop, unconditional Programmed stop, optional
M2	m	M	02 17 30	4.4 4.4 4.4	End of program without rewind, is written into the last program block End of sub-routine, is written into the last sub-routine block End of program with rewind to rewind stop, is written into the last program block
M3	m	M	03 04 05 • 19 *	4.4 4.4 4.4 4.4	Direction of spindle rotation cw Direction of spindle rotation ccw Spindle stop Exact spindle stop, angle under S in degrees
M4	m	M	36 37	3.17/4.4 3.17/4.4	Feed rate as programmed under F } also active at G33 Feed rate downrated by 1:100 }
M5	m	M	00 to 99	4.4	Miscellaneous functions, unassigned (except groups M1 to M4)
	5-4-2 1) 7-4-2 1)	( )		1.6 1.6	Start of remark End of remark
	EOB	LF		1.4	End of block

1) Punch pattern

* No other functions can be written in this block

• Reset state

■ Blockwise, all others self-retaining

+) Other addresses selectable (A,B,C,U,V,W,Q,E,P,H)

8.3 Programming Keys

8.3.1 Programming Key - 8M

Group	EIA	ISO	Code	Chapter	Function and description
	EOR	%		1.	Rewind stop, program start for tape read-in
	EOR ..EOB	% ..LF	0 to 9999	1.7	Program number
	o n /o /n	: N /: /N	1 to 9999	1.4	Main block Subordinate block Deletable main block Deletable subordinate block
G1	g	G	00 01 ● 10 11 02 03 33	3.2 3.3 3.4 3.4 3.5 3.5 3.6	Rapid traverse Linear interpolation Polar coordinate programming rapid traverse Polar coordinate programming linear interpolation Circular interpolation clockwise Circular interpolation counter clockwise Thread cutting
G2	g	G	■ 04 *	3.12	Dwell mode, time duration is specified under address X or F written in an own block
G3	g	G	■ 09	3.8	Feed deceleration
G5	g	G	■ 25 * ■ 26 *	3.14 3.14 3.20	Setting min. value; machining area X,Y,Z,4th. Setting max. value; machining area X,Y,Z,4th. spindle speed supervision S
G6	g	G	40 ● 41 42	3.22 3.22 3.22	No cutter radius compensation Cutter radius compensation lefthand-side Cutter radius compensation righthand-side
			40 ● 43 44	3.24 3.24/3.25 3.24/3.25	No tool offset compensation Positive tool offset } Axis parallel compensation Negative tool offset } linear path only
G7	g	G	■ 53	3.15	No zero offset; G54, G55, G56, G57 remain stored
G8	g	G	54 ● 55 56 57	3.15 3.15 3.15 3.15	Zero offset 1 for groups 1 to 3 Zero offset 2 for groups 1 to 3 Zero offset 3 for groups 1 to 3 Zero offset 4 for groups 1 to 3
G9	g	G	■ 59 *	3.15 3.15.1	Programmable additive zero offset Loading the zero offset G59 N...
G10	g	G	60 ● 63 64	3.8 3.9 3.10	Exact stop Tapping with compensating chuck Contouring operation, continuous transitions
G11	g	G	70 71	3.13 3.13	Inch input system } Reset state via machine data Metric input system }
G12	g	G	90 ● 91	3.1 3.1	Absolut position data input Incremtal position data input
G13	g	G	■ 92 *	3.16 3.16 3.19 3.7 3.11 3.23	Setting of actual value stores X,Y,Z,4th. Resetting of actual value stores without X,Y,Z,4th. Spindle speed limitation under address S in rpm Acceleration ramp time T for thread cutting Normalized diameter P Loading the tool offset G92 D...D
G14	g	G	94 ● 95 96 97	3.17 3.17 3.17 3.18 3.17	Feed rate under address F in mm/min Feed rate under address F in mm/rev. Feed rate under address F in mm/rev. and constant surface speed (S ≡ m/min) Freezing G96, store last speed command from G9[illegible]
G15	g	G	80 81 82 83 84 85 86 87 88 89	6.1 6.1/6.2 6.1/6.2 6.1/6.2 6.1/6.2 6.1/6.2 6.1/6.2 6.1/6.2 6.1/6.2 6.1/6.2	Cancel G81 to G89 Boring, centering Boring, counter sinking Deep hole drilling, chip breaking Tapping Boring 1 Boring 2 Boring 3 Boring 4 Boring 5
G16	g	G	G36 G37	3.27	Coordinate transformation OFF Coordinate transformation ON

8.3.2 Programming Key - 8MC

Group	EIA	ISO	Code	Chapter	Function and description
	EOR	%		1.	Rewind stop, program start for tape read-in
	EOR ..EOB	% ..LF	0 to '9999	1.7	Program number
	o n /o /n	: N /: /N	1 to 9999	1.4	Main block Subordinate block Deletable main block Deletable subordinate block
G1	g	G	00 01 ● 10 11 02 03 33	3.2 3.3 3.4 3.4 3.5 3.5 3.6	Rapid traverse Linear interpolation Polar coordinate programming rapid traverse Polar coordinate programming linear interpolation Circular interpolation clockwise Circular interpolation counter clockwise Thread cutting
G2	g	G	■ 04 *	3.12	Dwell mode, time duration is specified under address X or F written in an own block
G3	g	G	■ 09	3.8	Feed deceleration
G5	g	G	■ 25 * ■ 26 *	3.14 3.14 3.20	Setting min. value; machining area X,Y,Z,4th. .. 10th. Setting max. value; machining area X,Y,Z,4th. .. 10th. spindle speed supervision S
G6	g	G	40 ● 41 42	3.22 3.22 3.22	No cutter radius compensation Cutter radius compensation lefthand-side Cutter radius compensation righthand-side
			40 ● 43 44	3.24 3.24/3.25 3.24/3.25	No tool offset compensation Positive tool offset } Axis parallel compensation Negative tool offset } linear path only
G7	g	G	■ 53	3.15	No zero offset; G54, G55, G56, G57 remain stored
G8	g	G	54 ● 55 56 57	3.15 3.15 3.15 3.15	Zero offset 1 for groups 1 to 3 Zero offset 2 for groups 1 to 3 Zero offset 3 for groups 1 to 3 Zero offset 4 for groups 1 to 3
G9	g	G	■ 59 *	3.15	Programmable additive zero offset
				3.15.1	Loading the zero offset G59 N...
G10	g	G	60 ● 63 64	3.8 3.9 3.10	Exact stop Tapping with compensating chuck Contouring operation, continuous transitions
G11	g	G	70 71	3.13 3.13	Inch input system } Reset state via machine data Metric input system }
G12	g	G	90 ● 91	3.1 3.1	Absolut position data input Incremtal position data input
G13	g	G	■ 92 *	3.16 3.16 3.19 3.7 3.11 3.23	Setting of actual value stores X,Y,Z,4th. .. 10th. Resetting of actual value stores without X,Y,Z,4th. .. 10th. Spindle speed limitation under address S in rpm Acceleration ramp time T for thread cutting Normalized diameter P Loading the tool offset G92 D...D
G14	g	G	94 ● 95 96 97	3.17 3.17 3.17 3.18 3.17	Feed rate under address F in mm/min Feed rate under address F in mm/rev. Feed rate under address F in mm/rev. and constant surface speed (S ≡ m/min) Freezing G96, store last speed command from G96
G15	g	G	80 81 82 83 84 85 86 87 88 89	6.1 6.1/6.2 6.1/6.2 6.1/6.2 6.1/6.2 6.1/6.2 6.1/6.2 6.1/6.2 6.1/6.2 6.1/6.2	Cancel G81 to G89 Boring, centering Boring, counter sinking Deep hole drilling, chip breaking Tapping Boring 1 Boring 2 Boring 3 Boring 4 Boring 5

Programming Key - 8MC (continued)

Group	EIA	ISO	Code	Chapter	Function and description
	x	X	0.001 to ±99999.999	2.1	Position data in mm
			0.001 to +99999.999	3.12	Dwell in sec.
	y	Y	0.001 to ±99999.999	2.1	Position data in mm
	z	Z	0.001 to ±99999.999	2.1	Position data in mm
	4.th to 10.th axis	4.th to 10.th axis	0.001 to ±99999.999	2.1	Position data in mm or degrees; possible addresses A,B,C,U,V,W,Q,E,P,H
	a	A +)	0 to 359.99999	3.4	Angle in degrees for polar coordinates
	p	P +)	+0.001 to +99999.999 ±0.001 to ±99999.999	3.4 3.5	Radius at polar coordinates in mm Radius for circular interpolation in mm
	i	I	0.001 to ±99999.999	2.2	Interpolation parameter for X-axis in mm
			1 to 2000.000	3.6	Thread lead in mm
	j	J	0.001 to ±99999.999	2.2	Interpolation parameter for Y-axis in mm
			1 to 2000.000	3.6	Thread lead in mm
	k	K	0.001 to ±99999.999	2.2	Interpolation parameter for Z-axis in mm
			1 to 2000.000	3.6	Thread lead in mm
	d	D	00 1 to 199	3.23 3.24	Cancellation of tool compensation Tool compensation number
	r	R	00 to 49	5.0	Parameter
	f	F	0.001 to 15.000	3.17	Feed mm/min (Inch/min see 8.2.6)
			0.001 to 99.999	3.12	Dwell in sec.
			0.001 to 50.000	3.17	Feed in mm/rev. (Inch/min see 8.2.6)
	s	S	1 to 9999	4.1 3.18	Spindle speed in rev./min or 0.1 rev./min or constant surface speed in m/min or 0.1 m/min
			1 to 9999	3.19/3.20	Spindle speed limitation in rev./min or 0.1 rev./min
			0 to 359	4.4	Spindle stop in degrees, distance from zero mark
	t	T	1 to 999999	4.3	Tool number (tool position)
			0,1,2,3,4,5	3.7	Time constants
	h	H +)	1 to 999999	4.2	Auxiliary function
	l	L	001.. to 999..	1.8	Sub-routine number
			...01 to ...99	1.9/5.5	Number of runs of sub-routine
			999	5.7	Lock buffer read in
M1	m	M	■ 00 01	4.4 4.4	Programmed stop, unconditional Programmed stop, optional
M2	m	M	02 17 30	4.4 4.4 4.4	End of program without rewind, is written into the last program block End of sub-routine, is written into the last sub-routine block End of program with rewind to rewind stop, is written into the last program block
M3	m	M	03 04 05 ● 19 *	4.4 4.4 4.4 4.4	Direction of spindle rotation cw Direction of spindle rotation ccw Spindle stop Exact spindle stop, angle under S in degrees
M4	m	M	36 37	3.17/4.4 3.17/4.4	Feed rate as programmed under F } also active at G33 Feed rate downrated by 1:100 }
M5	m	M	00 to 99	4.4	Miscellaneous functions, unassigned (except groups M1 to M4)
	5-4-2 1) 7-4-2 1)	( )		1.6 1.6	Start of remark End of remark
	EOB	LF		1.4	End of block

1) Punch pattern

* No other functions can be written in this block

● Reset state

■ Blockwise, all others self-retaining

+) Other addresses selectable (A,B,C,U,V,W,Q,E,P,H)

8.3.3 Programming Key - Sprint 8M

Group	EIA	ISO	Code	Chapter	Function and description
	EOR	%		1.	Rewind stop, program start
	EOR .. EOB	% ..LF	0 to 9999	1.7	Program number
	o n /o /n	: N /: /N	1 to 9999	1.4	Main block Sub-ordinate block Deletable main block Deletable sub-ordinate block
G1	g	G	00 01 ● 10 11 02 03 33	3.2 3.3 3.4 3.4 3.5/7.3/7.4 3.5/7.3/7.4 3.6/7.5	Rapid traverse Linear interpolation Rapid traverse at polar coordinates programming Linear interpolation at polar coordinates programming Circular interpolation cw Circular interpolation ccw Thread cutting
G2	g	G	■ 04 *	3.12	Dwell, predetermined addresses X or F in ms, written in an own block
G3	g	G	■ 09	3.8	Feed rate reduction
G4	g	G	17 ● 18 19	7.7/7.9 7.7/7.9 7.7/7.9	Selection of plane X-Y X-4th Selection of plane X-Z resp. Y-4th Selection of plane Y-Z Z-4th } Length compensation always in the main axis outside the selected plane
G5	g	G	■ 25 * ■ 26 *	3.14 3.14 3.20	Setting min. value; machining area X,Y,Z,4th Setting max. value; machining area X,Y,Z,4th; spindle speed supervision S
G6	g	G	40 ● 41 42	7.8/7.9 7.8/7.9 7.8/7.9	No cutter radius compensation Cutter radius compensation lefthand-side Cutter radius compensation righthand-side
G7	g	G	■ 53	3.15/7.6	No zero offset; G54, G55, G56, G57 remain stored
G8	g	G	54 ● 55 56 57	7.6 7.6 7.6 7.6	Zero offset 1 Zero offset 2 Zero offset 3 Zero offset 4
G9	g	G	■ 59 *	3.15 7.6.	Programmable additive zero offset Loading the zero offset G59 N...
G10	g	G	60 ● 63 64	3.8 3.9 3.10	Exact stop Tapping with compensating chuck Contouring operation, continuous transitions
G11	g	G	70 71	3.13 3.13	Inch input system Metric input system } Reset state via machine data
G12	g	G	90 ● 91	3.1 3.1	Absolute position data input Incremental position data input
G13	g	G	■ 92 *	3.16 3.16 3.19 3.7 7.9.	Setting of actual value stores X,Y,Z,4th; Resetting of actual value stores without X,Y,Z,4th; Spindle speed limitation under address S in rpm; Acceleration ramp time T for thread cutting; Normalized diameter P Loading the tool offset G92 D...D
G14	g	G	94 ● 95 96 97	3.17 3.17 3.17 3.18 3.17	Feed rate under address F in mm/min Feed rate under address F in mm/rev. Feed rate under address F in mm/rev. and constant surface speed (S ≙ m/min) Freezing G96, store last speed command from G96
G15	g	G	80 81 82 83 84 85 86 87 88 89	6.1 6.1/6.2 6.1/6.2 6.1/6.2 6.1/6.2 6.1/6.2 6.1/6.2 6.1/6.2 6.1/6.2 6.1/6.2	Cancel G81 to G89 Boring, centering Boring, counter sinking Deep hole drilling, chip breaking Tapping Boring 1 Boring 2 Boring 3 Boring 4 Boring 5
G16	g	G	G36 G37	3.27	Coordinate transformation OFF Coordinate transformation ON

Programming Key - Sprint 8M (continued)

Group	EIA	ISO	Code	Chapter	Function and description
	x	X	0.001 to ±99999.999	7.2	Position data in mm
			0.001 to +99999.999	3.12	Dwell in sec.
	y	Y	0.001 to ±99999.999	7.2	Position data in mm
	z	Z	0.001 to ±99999.999	7.2	Position data in mm
	4.th axis	4. th axis	0.001 to ±99999.999	7.2	Position data in mm or degrees; possible addresses A,B,C,U,V,W,Q,E,P,H
	a	A+)	0 to 359.99999 0 to 359.99999	7.11 3.4	Angle in degrees for contour description; Angle in degrees for polar coordinates
	p	P+)	0,-0 +0.001 to +99999.999 -0.001 to -99999.999 ±0.001 to ±99999.999	7.11 3.4/7.11 7.11 3.5	Corner at contour; Radius at contour and polar coordinates in mm; Chamfer at contour in mm; Radius for circular interpolation in mm
	i	I	0.001 to ±99999.999	2.2	Interpolation parameter for X-axis in mm
			1 to 2000.000	3.6/7.5	Thread lead in mm
	j	J	0.001 to ±99999.999	2.2	Interpolation parameter for Y-axis in mm
			1 to 2000.000	3.6/7.5	Thread lead in mm
	k	K	0.001 to ±99999.999	2.2	Interpolation parameter for Z-axis in mm
			1 to 2000.000	3.6/7.5	Thread lead in mm
	d	D	00 1 to 99	7.9 7.10	Cancellation of tool compensation Tool compensation number
	r	R	00 to 49	5.0	Parameter
	f	F	0.001 to 15.000	3.17	Feed mm/min (Inch/min see 8.2.6)
			0.001 to 99.999	3.12	Dwell in sec.
			0.001 to 50.000	3.17	Feed in mm/rev (Inch/min see 8.2.6)
	s	S	1 to 9999	4.1 3.18	Spindle speed in rev./min or 0.1 rev./min or constant surface speed in m/min or 0.1 m/min
			1 to 9999	3.19/3.20	Spindle speed limitation in rev./min or 0.1 rev./min
			0 to 359	4.4	Spindle stop in degrees, distance from zero mark
	t	T	1 to 9999	4.3	Tool number (tool position)
			0,1,2,3,4,5	3.7	Time constants
	h	H+)	1 to 999999	4.2	Auxiliary function
	l	L	001.. to 999..	1.8	Sub-routine number
			...01 to ...99	1.9/5.5	Number of runs of sub-routine
			999	5.7	Lock buffer read in
M1	m	M	■ 00 01	4.4 4.4	Programmed stop, unconditional Programmed stop, optional
M2	m	M	02 17 30	4.4 4.4 4.4	End of program without rewind, is written into the last prog. block End of sub-routine, is written into the last sub-routine block End of program with rewind to rewind stop, is written into the last program block
M3	m	M	03 04 05 • 19 *	4.4 4.4 4.4 4.4	Direction of spindle rotation cw Direction of spindle rotation ccw Spindle stop Exact spindle stop, angle under S in degrees
M4	m	M	36 37	3.17/4.4 3.17/4.4	Feed rate as programmed under F } also active at G33 Feed rate downrated by 1:100 }
M5	m	M	00 to 99	4.4	Miscellaneous functions, unassigned (except groups M1 to M4)
	5-4-2 1) 7-4-2 1)	( )		1.6 1.6	Start of remark End of remark
	EOB	LF		1.4	End of block

1) Punch pattern

* No other functions can be written in this block

• Reset state

■ Blockwise, all others self-retaining

+) Other addresses selectable (A,B,C,U,V,W,Q,E,P,H)

Issued by Siemens AG,
Numerical Controls and
Machine Tool Drives Department
Postfach 4848, D-8500 Nürnberg 1

Subject to change without prior notice

Siemens Aktiengesellschaft

Order No. E80210-T6-X-A8-7600
Printed in the Federal Republic of Germany
082428 PA 7851.

progress in automation: Siemens