SIEMENS

SINUMERIK 808D
SINUMERIK 808D, SINUMERIK 808D ADVANCED
Programming and Operating Manual (ISO Turning/Milling)
User Manual

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER
indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING
indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

6FC5398-0DP40-0BA2, 06/2017

Preface

Applicable products

This manual is valid for the following control systems:

Control system	Software version
SINUMERIK 808D ADVANCED T (Turning) SINUMERIK 808D ADVANCED M (Milling)	V4.8.1: PPU16x.3 with spindle/feed servo system
SINUMERIK 808D (Turning) SINUMERIK 808D (Milling)	V4.8.1: PPU141.3 with feed servo system

Documentation components and target audience

End-user documentation	Target audience
Programming and Operating Manual (Turning)	Programmers and operators of turning machines
Programming and Operating Manual (Milling)	Programmers and operators of milling machines
Programming and Operating Manual (ISO Turning/Milling)	Programmers and operators of turning/milling machines
Programming and Operating Manual (Manual Machine Plus (MM+), Turning)	Programmers and operators of turning machines
Diagnostics Manual	Mechanical and electrical designers, commissioning engineers, machine operators, and service and maintenance personnel
Manufacturer/service documentation	**Target audience**
Commissioning Manual	Installation personnel, commissioning engineers, and service and maintenance personnel
Function Manual	Mechanical and electrical designers, technical professionals
Parameter Manual	Mechanical and electrical designers, technical professionals
Service Manual	Mechanical and electrical designers, technical professionals, commissioning engineers, and service and maintenance personnel
Readme file	
Third-party software - Licensing terms and copyright information	

My Documentation Manager (MDM)

Under the following link you will find information to individually compile your documentation based on the Siemens content:
www.siemens.com/mdm

Standard scope

This manual only describes the functionality of the standard version. Extensions or changes made by the machine tool manufacturer are documented by the machine tool manufacturer.

Technical support

Country	Hotline [1)]	Further service contact information:
Germany	+49 911 895 7222	• Worldwide Web site: https://support.industry.siemens.com/cs/ww/en/ • Chinese Web site: http://www.siemens.com.cn/808D
China	+86 400 810 4288	

[1)] You can find more hotline information at the worldwide Web site given above.

EC Declaration of Conformity

The EC Declaration of Conformity for the EMC Directive can be found on the Internet at http://www.siemens.com/automation/service&support.

Here, enter the number "**67385845**" as the search term or contact your local Siemens office.

Table of contents

Preface **2**

1 Fundamental safety instructions **6**

1.1 General safety instructions 6

1.2 Industrial security 6

2 Fundamentals of programming **7**

2.1 Introductory remarks 7
2.1.1 Siemens mode 7
2.1.2 ISO dialect mode 7
2.1.3 Switching between Siemens and ISO modes 7
2.1.4 Display of the G code 8
2.1.5 Maximum number of axes/axis identifiers 8
2.1.6 Selecting G code system A, B or C 8
2.1.7 Programming the decimal point 9
2.1.8 Comments 11
2.1.9 Block skip 11
2.1.10 Notes on the PLC interface signal address representation 11

2.2 Prerequisites for the feed 12
2.2.1 Rapid traverse 12
2.2.2 Path feed (F function) 12
2.2.3 Linear feed (G98/G94) 14
2.2.4 Inverse-time feed (G93) (milling only) 14
2.2.5 Revolutional feedrate (G99/G95) 14

3 Turning **15**

3.1 G code table 15
3.1.1 SINUMERIK ISO dialect turning mode A 15
3.1.2 SINUMERIK ISO dialect turning mode B 17
3.1.3 SINUMERIK ISO dialect turning mode C 19

3.2 Travel commands 21
3.2.1 Interpolation commands 21
3.2.1.1 Rapid traverse (G00) 21
3.2.1.2 Linear interpolation (G01) 22
3.2.1.3 Circular interpolation (G02, G03) 23
3.2.1.4 Contour definition programming and insertion of chamfers and radii 26
3.2.1.5 Cylindrical interpolation (G07.1) 27
3.2.1.6 Polar coordinates interpolation (G12.1, G13.1) (TRANSMIT) 28
3.2.2 Reference point approach with G functions 30
3.2.2.1 Reference point approach with intermediate point (G28) 30
3.2.2.2 Checking the reference position (G27) 31
3.2.2.3 Reference point approach with reference point selection (G30) 31
3.2.3 Use of the thread cutting function 32
3.2.3.1 Thread cutting with constant lead (G32) 32
3.2.3.2 Interlinking of threads (G32) 34
3.2.3.3 Cutting of multiple-start threads (G32) 34
3.2.3.4 Thread cutting with variable lead (G34) 36

3.3 Measuring commands 37
3.3.1 The coordinate system 37
3.3.1.1 Machine coordinate system (G53) 37
3.3.1.2 Workpiece coordinate system (G50) 38
3.3.1.3 Resetting the tool coordinate system (G50.3) 38
3.3.1.4 Selection of a workpiece coordinate system (G54 to G59) 38
3.3.1.5 Writing work offset/tool offsets (G10) 39
3.3.2 Defining the input modes for the coordinate values 40
3.3.2.1 Diameter and radius programming for the X-axis 40
3.3.2.2 Input inch/metric (G20, G21) 40
3.3.3 Time-controlled commands (G04) 41

3.3.4 Tool offset functions....41
3.3.4.1 Tool offset data memory....41
3.3.4.2 Tool length compensation....42
3.3.4.3 Tool nose radius compensation (G40, G41/G42)....42
3.3.5 S-, T-, M- and B functions....46
3.3.5.1 Spindle function (S function)....46
3.3.5.2 Constant cutting rate (G96, G97)....47
3.3.5.3 Tool change with T functions (T function)....48
3.3.5.4 Additional function (M function)....48
3.3.5.5 M functions of spindle control (M19, M29)....49
3.3.5.6 M functions for subroutine calls (M98, M99)....49
3.3.5.7 Macro call via M function....49
3.3.5.8 M functions (M08, M09)....50

3.4 Additional functions....50
3.4.1 Program-support functions....50
3.4.1.1 Fixed cycles (G90, G92, G94)....50
3.4.1.2 Multiple repetitive cycles (G70 to G76)....58
3.4.1.3 Drilling cycle (G80 to G89)....71
3.4.2 Programmable data input....78
3.4.2.1 Changing the tool offset value (G10)....78
3.4.2.2 M function for calling subroutines (M98, M99)....79
3.4.3 Eight-digit program number (M98, G65/G66)....80
3.4.4 Measuring functions (G31)....81
3.4.5 Macro programs....81
3.4.5.1 Differences with subroutines....82
3.4.5.2 Macro program call (G65, G66, G67)....82
3.4.6 Special functions....87
3.4.6.1 G05....87
3.4.6.2 Switchover modes for DryRun and skip levels....87

4 Milling....88

4.1 G code table....88

4.2 Drive commands....90
4.2.1 Interpolation commands....90
4.2.1.1 Cylindrical interpolation (G07.1)....90
4.2.1.2 Rapid traverse (G00)....92
4.2.1.3 Linear interpolation (G01)....93
4.2.1.4 Circular interpolation (G02, G03)....94
4.2.1.5 Contour definition programming and addition of chamfers or radiuses....96
4.2.1.6 Helical interpolation (G02, G03)....97
4.2.2 Reference point approach with G functions....98
4.2.2.1 Reference point approach with intermediate point (G28)....98
4.2.2.2 Checking the reference position (G27)....99
4.2.2.3 Reference point approach with reference point selection (G30)....99

4.3 Motion commands....100
4.3.1 The coordinate system....100
4.3.1.1 Machine coordinate systems (G53)....100
4.3.1.2 Workpiece coordinate system (G92)....101
4.3.1.3 Resetting the tool coordinate system (G92.1)....101
4.3.1.4 Selection of a workpiece coordinate system (G54 to G59)....101
4.3.1.5 Writing work offset/tool offsets (G10)....102
4.3.1.6 Local coordinate system (G52)....103
4.3.1.7 Selection of the plane (G17, G18, G19)....103
4.3.1.8 Parallel axes (G17, G18, G19)....104
4.3.1.9 Rotation of the coordinate system (G68, G69)....104
4.3.1.10 3D rotation (G68/G69)....105
4.3.2 Defining the input modes of the coordinate values....106
4.3.2.1 Absolute/incremental dimensioning (G90, G91)....106
4.3.2.2 Inch/metric input (G20, G21)....107
4.3.2.3 Scaling (G50, G51)....107
4.3.2.4 Programmable mirroring (G50.1, G51.1)....109
4.3.3 Time-controlled commands (G04)....111
4.3.4 Tool offset functions....111
4.3.4.1 Tool offset data memory....111
4.3.4.2 Tool length compensation (G43, G44, G49)....112

4.3.4.3	Cutter radius compensation (G40, G41, G42)	113
4.3.4.4	Collision detection	116
4.3.5	S-, T-, M- and B functions	119
4.3.5.1	Spindle function (S function)	119
4.3.5.2	Tool function	119
4.3.5.3	Additional function (M function)	119
4.3.5.4	M functions of spindle control (M19, M29)	120
4.3.5.5	M functions for subroutine calls (M98, M99)	120
4.3.5.6	Macro call via M function	121
4.3.5.7	M functions (M08, M09)	121
4.3.6	Controlling the feedrate	122
4.3.6.1	Compressor in ISO dialect mode	122
4.3.6.2	Exact stop (G09, G61), continuous-path mode (G64), and tapping (G63)	122
4.4	Additional functions	123
4.4.1	Program supporting functions	123
4.4.1.1	Fixed drilling cycles (G73, G74, G76, G80 to G87, G89)	123
4.4.1.2	High-speed deep hole drilling cycle with chip breakage (G73)	126
4.4.1.3	Fine drilling cycle (G76)	128
4.4.1.4	Drilling cycle, counterboring (G81)	130
4.4.1.5	Countersink drilling cycle (G82)	132
4.4.1.6	Deep hole drilling cycle with chip removal (G83)	133
4.4.1.7	Boring cycle (G85)	134
4.4.1.8	Boring cycle (G86)	136
4.4.1.9	Boring cycle - reverse countersinking (G87)	137
4.4.1.10	Boring cycle (G89)	139
4.4.1.11	"Drilling a right-hand thread without any compensating chuck" cycle (G84)	140
4.4.1.12	"Drilling a left-hand thread without any compensating chuck" cycle (G74)	142
4.4.1.13	Left or right tapping cycle (G84/G74)	144
4.4.1.14	Deselection of a fixed cycle (G80)	146
4.4.1.15	Program example with a tool length compensation and fixed cycles	146
4.4.2	Programmable data input (G10)	148
4.4.2.1	Changing the tool offset value	148
4.4.2.2	M function for calling subroutines (M98, M99)	148
4.4.3	Eight-digit program number (M98, G65/G66)	149
4.4.4	Polar coordinates (G15, G16)	149
4.4.5	Measuring functions	150
4.4.5.1	Measuring with "delete distance-to-go" (G31)	150
4.4.5.2	"Tool life control" function	152
4.4.6	Macro programs	152
4.4.6.1	Differences with subroutines	152
4.4.6.2	Macro program call (G65, G66, G67)	152
4.4.6.3	Macro call via G function (G21)	157
4.4.7	Special functions	159
4.4.7.1	Contour repetition (G72.1, G72.2)	159
4.4.7.2	Switchover modes for DryRun and skip levels	161

1 Fundamental safety instructions

1.1 General safety instructions

WARNING

Danger to life if the safety instructions and residual risks are not observed

If the safety instructions and residual risks in the associated hardware documentation are not observed, accidents involving severe injuries or death can occur.

- Observe the safety instructions given in the hardware documentation.
- Consider the residual risks for the risk evaluation.

WARNING

Danger to life or malfunctions of the machine as a result of incorrect or changed parameterization

As a result of incorrect or changed parameterization, machines can malfunction, which in turn can lead to injuries or death.

- Protect the parameterization (parameter assignments) against unauthorized access.
- Respond to possible malfunctions by applying suitable measures (e.g. EMERGENCY STOP or EMERGENCY OFF).

1.2 Industrial security

Note

Industrial security

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens products and solutions only represent one component of such a concept.

The customer is responsible for preventing unauthorized access to its plants, systems, machines and networks. Systems, machines and components should only be connected to the enterprise network or the internet if and to the extent necessary and with appropriate security measures (e.g. use of firewalls and network segmentation) in place.

Additionally, Siemens' guidance on appropriate security measures should be taken into account. For more information about industrial security, please visit:

Industrial security (http://www.siemens.com/industrialsecurity).

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends to apply product updates as soon as available and to always use the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at:

Industrial security (http://www.siemens.com/industrialsecurity).

WARNING

Danger to life as a result of unsafe operating states resulting from software manipulation

Software manipulations (e.g. viruses, trojans, malware or worms) can cause unsafe operating states in your system that may lead to death, serious injury, and property damage.

- Keep the software up to date.
- Incorporate the automation and drive components into a holistic, state-of-the-art industrial security concept for the installation or machine.
- Make sure that you include all installed products into the holistic industrial security concept.
- Protect files stored on exchangeable storage media from malicious software by with suitable protection measures, e.g. virus scanners.

2 Fundamentals of programming

WARNING
Danger to life and/or damage to machine due to insecure part programs
Running an insecure part program on your machine may cause unexpected attacks to the machine, which in turn can lead to death, personal injuries, and/or machine damage.
• Make sure you use part programs that come from trusted sources.

2.1 Introductory remarks

2.1.1 Siemens mode

The following conditions are valid in Siemens mode:

- The default of the G commands can be defined for each channel via the machine data 20150 $MC_GCODE_RESET_VALUES.
- No language commands from the ISO dialects can be programmed in Siemens mode.

2.1.2 ISO dialect mode

The following conditions are valid in the active ISO dialect mode:

- ISO dialect mode can be set with machine data as the default setting of the control system. The control system reboots by default in ISO dialect mode subsequently.
- Only G functions from the ISO dialect can be programmed; programming of Siemens G functions is impossible in ISO Mode.
- The ISO dialect and Siemens language cannot be mixed in the same NC block.
- G commands cannot be used to switch between ISO dialect M and ISO dialect T.
- Subroutines that are programmed in Siemens mode can be called.
- If Siemens functions are to be used, switch the control system to Siemens mode first.

Note

Switching from Siemens mode to ISO dialect mode does not change the default value **0x04** of MD10890 EXTERN_TOOLPROG_MODE.

- **0x04**: select the tool offset only with an H number in ISO dialect mode.
- **0**: select the tool offset only with a D number in ISO dialect mode.

2.1.3 Switching between Siemens and ISO modes

The control system supports the following two programming language modes:

- Siemens language mode
- ISO dialect mode

Note that the active tool, tool offsets and workpiece offsets are not influenced by the mode changeover.

Switchover through G functions

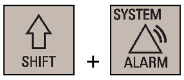
The following G functions can be used to switch between Siemens mode and ISO dialect mode:

- G290 - Siemens NC programming language active
- G291 - ISO Dialect NC Programming language active

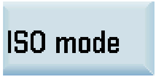
G290 and G291 must be programmed alone in an NC block.

Switchover on the HMI

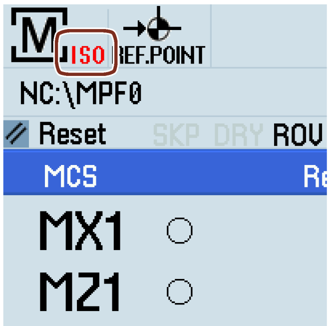
Operating sequence



1. Select the desired operating area.



2. Press this softkey on the PPU. The control system automatically starts the mode change from Siemens mode to ISO dialect mode. After the changeover, "ISO" is shown on the upper left corner of the screen.



To change from ISO mode back to Siemens mode, press the same softkey again.

2.1.4 Display of the G code

The G code is displayed in the same language type (Siemens or ISO dialect) as the relevant current block display. If the display of the blocks is suppressed with DISPLOF, the G codes continue to be displayed in the language type of the active block.

Example

The G functions of ISO dialect mode are used to call the Siemens standard cycles. To do this, DISPLOF is programmed at the start of the relevant cycle; this way the G functions that are programmed in ISO dialect language continue to be displayed.

```
PROC CYCLE328 SAVE DISPLOF
N10 ...
...
N99 RET
```

Procedure

The Siemens shell cycles are called via main programs. Siemens mode is selected automatically by calling the shell cycle.

With DISPLOF, the block display is frozen on calling the cycle; the display of the G code continues in ISO mode.

The G codes that were changed in the shell cycle are reset to their original status at the end of the cycle with the "SAVE" attribute.

2.1.5 Maximum number of axes/axis identifiers

The maximum number of axes in ISO dialect mode is 9. The axis identifiers for the first three axes are defined with X, Y and Z permanently. All other axes can be identified with letters A, B, C, U, V and W.

2.1.6 Selecting G code system A, B or C

ISO dialect T differentiates between G code systems A, B, and C. By default, the G code system A is active. Via the MD10881 $MN_MM_EXTERN_GCODE_SYSTEM, the G code system A, B or C is selected as follows:

$MN_MM_EXTERN_GCODE_SYSTEM = 0: G code system B

$MN_MM_EXTERN_GCODE_SYSTEM = 1: G code system A

$MN_MM_EXTERN_GCODE_SYSTEM = 2: G code system C

G code system A

If the G code system A is active, then G91 is unavailable. In this case, an incremental axis movement with the address characters U, V and W is programmed for axes X, Y and Z. The address characters U, V and W are not available in this case as axis identifiers, and as a result, the maximum number of axes reduces to 6.

The address H is used for programming the incremental movements of the C-axis in the G code system A.

In order for the shell cycle to work in the correct G code system, the corresponding system must be entered in GUD variable _ZSFI[39].

Note

- If not otherwise noted, the manual in hand describes G code system A.
- For the differences between G code systems A, B, and C, see Section "G code table (Page 15)".

2.1.7 Programming the decimal point

In ISO Dialect mode, there are two notations for evaluating the programmed values without decimal point:

- **Pocket calculator notation**
  Values without decimal point are interpreted as mm, inch or degree.
- **Standard notation**
  Values without decimal points are multiplied with the conversion factor.

Complete the setting via MD10884 EXTERN_FLOATINGPOINT_PROG.

Note

Switching from Siemens mode to ISO dialect mode does not change the current setting of MD10884 EXTERN_FLOATINGPOINT_PROG.

There are two different conversion factors, **IS-B** and **IS-C.** This evaluation refers to the addresses X, Y, Z, U, V, W, A, B, C, I, J, K, Q, R, and F.

Complete the setting over MD10886 EXTERN_INCREMENT_SYSTEM.

Example:

Linear axis in mm:

- X 100.5
  Corresponds to value with decimal point: 100.5 mm
- X 1000
  - Pocket calculator notation: 1000 mm
  - Standard notation:
    IS-B: 1000 * 0.001= 1 mm
    IS-C: 1000 * 0.0001 = 0.1 mm

ISO dialect turning

See the following table for different conversion factors for IS-B and IS-C.

Address	Unit	IS-B	IS-C
Linear axis	mm inch	0.001 0.0001	0.0001 0.00001
Rotary axis	Degree	0.001	0.0001
F feed G94 (mm/inch per min.)	mm inch	1 0.01	1 0.01

Address	Unit	IS-B	IS-C
F feed G95 (mm/inch per rotation) $MC_EXTERN_FUNCTION_MASK Bit8 = 0 Bit8 = 1	 mm inch mm inch	 0.01 0.0001 0.0001 0.000001	 0.01 0.0001 0.0001 0.000001
F thread lead	mm inch	0.0001 0.000001	0.0001 0.000001
C chamfer	mm inch	0.001 0.001	0.0001 0.0001
R radius. G10 toolcorr	mm inch	0.001 0.001	0.0001 0.0001
I. J, K interpolation parameters	mm inch	0.001 0.001	0.0001 0.0001
G04 X or U		0.001	0.001
A contour angle		0.001	0.001
G76, G92 thread cutting cycle $MC_EXTERN_FUNCTION_MASK Bit8 = 0 F as feed such as G94, G95 Bit8 = 1 F as thread lead			
G84, G88 tapping cycle $MC_EXTERN_FUNCTION_MASK Bit9 = 0 G95 F Bit8 = 1 G95 F	 mm inch mm inch	 0.01 0.0001 0.0001 0.000001	 0.01 0.0001 0.0001 0.000001

ISO dialect milling

See the following table for different conversion factors for IS-B and IS-C.

Address	Unit	IS-B	IS-C
Linear axis	mm inch	0.001 0.0001	0.0001 0.00001
Rotary axis	Degree	0.001	0.0001
F feed G94 (mm/inch per min.)	mm inch	1 0.01	1 0.01
F feed G95 (mm/inch per rev.)	mm inch	0.01 0.0001	0.01 0.0001
F thread lead	mm inch	0.01 0.0001	0.01 0.0001
C chamfer	mm inch	0.001 0.0001	0.0001 0.00001
R radius, G10 toolcorr	mm inch	0.001 0.0001	0.0001 0.00001
Q	mm inch	0.001 0.0001	0.0001 0.00001
I, J, K interpolation parameters	mm inch	0.001 0.0001	0.0001 0.00001

Address	Unit	IS-B	IS-C
G04 X or U	s	0.001	0.001
A contour angle	Degree	0.001	0.0001
G74, G84 tapping cycles $MC_EXTERN_FUNCTION_MASK Bit8 = 0 F as feed such as G94, G95 Bit8 = 1 F as thread lead			

2.1.8 Comments

In ISO dialect mode, brackets are interpreted as comment signs. In Siemens mode, ";" is interpreted as comment. To simplify matters, an ";" is also understood as comment in ISO dialect mode.

If the comment start sign "(" is used inside a comment again, the comment is ended only if all the open brackets are closed again.

Example:

```
N5 (comment) X100 Y100
N10 (comment(comment)) X100 Y100
N15 (comment(comment) X100) Y100
```

X100 Y100 is executed in blocks N5 and N10, but only Y100 in block N15, because the first bracket is closed only after X100. Everything up to that point is interpreted as a comment.

2.1.9 Block skip

The sign of skipping or suppression of blocks "/" can be used at any convenient position in a block, that is, even in the middle of the block. If the programmed block skip level is active on the date of the compilation, the block is not compiled from this point up to the end of the block. An active block skip level has the same effect as a block end.

Example:

N5 G00 X100. /3 YY100 --> Alarm 12080 "Syntax error"
N5 G00 X100. /3 YY100 --> no alarm, if block skip level 3 is active

Block skip signs within a comment are not interpreted as block skip signs

Example:

N5 G00 X100. (/3 Part1) Y100
;the Y axis is traversed even when the block skip level 3 is active

The block skip levels /1 to /9 can be active. Block skip values <1 and >9 lead to alarm 14060 "Impermissible skip level for differential block skip".

The function is mapped to the existing Siemens skip levels. Unlike the ISO Dialect original, "/" and "/1" are separate skip levels that must also be activated separately.

Note

The "0" in "/0" can be omitted.

2.1.10 Notes on the PLC interface signal address representation

Currently, PLC interface signal addresses are represented by the V structure on the HMI while the manual shows them by the DB structure.

See the following table for the relationship between the two representations.

V Structure		DB Structure	
Access	Example	Example	Access
Bit	V38000002.1	DB3800.DBX2.1	Bit
Byte	VB38000002	DB3800.DBB2	Byte
Word	VW38000002	DB3800.DBW2	Word
Double Word	VD38000004	DB3800.DBD4	Double word

2.2 Prerequisites for the feed

The following Section describes the feed function with which the feedrate (covered path per minute or per rotation) of a cutting tool is defined.

2.2.1 Rapid traverse

Rapid traverse is used for positioning (G00) as well as for manual traverse with rapid traverse (JOG). In rapid traverse, each axis is traversed with the rapid traverse rate set for the individual axes. The rapid traversing rate is defined by the machine manufacturer and it is specified by the machine data for the individual axes. As the axes traverse independently of each other, each axis reaches its target point at a different time. Hence, the resulting tool path is generally not a straight line.

2.2.2 Path feed (F function)

Note

Unless otherwise specified, the unit "mm/min" is always used for the feedrate of the cutting tool in this documentation.

The feed, with which a tool should be traversed in case of linear interpolation (G01) or circular interpolation (G02, G03), is programmed with the address character "F".

After the next address character "F", the feed of the cutting tool is specified in "mm/min".

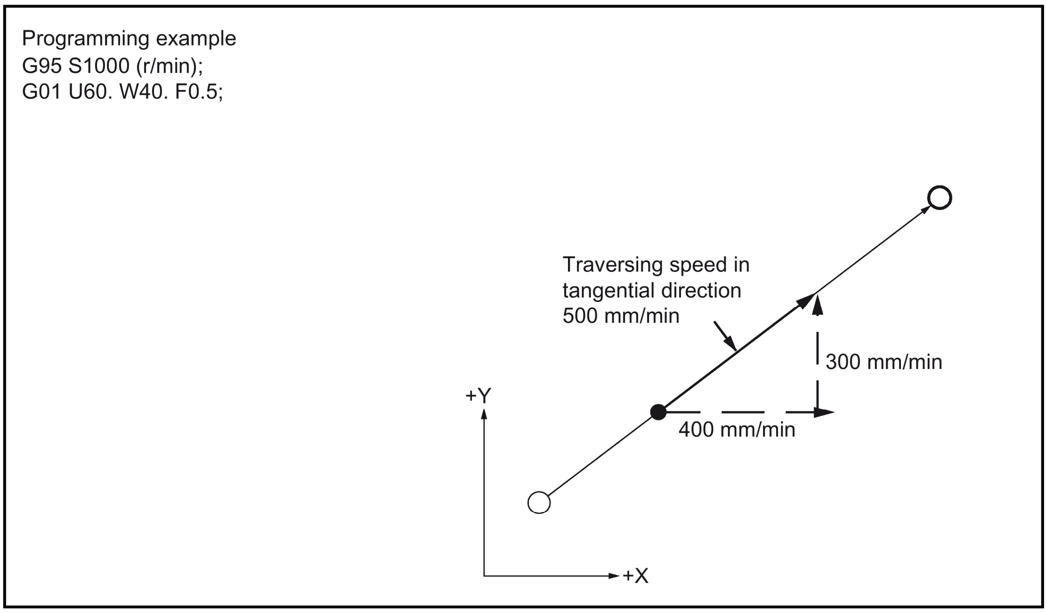
The permissible range of the F value is given in the documentation of the machine manufacturer.

Possibly, the feed is restricted upward by the servo-system and by the mechanics. The maximum feed is set through the machine data and is restricted before exceeding the value defined there.

The path feed is generally composed of the individual speed components of all geometry axes participating in the movement and refers to the center point (see following images).

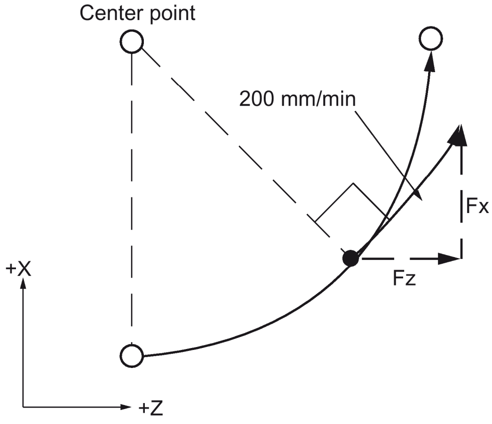
Turning:

Linear interpolation with two axes:



Circular interpolation with two axes:

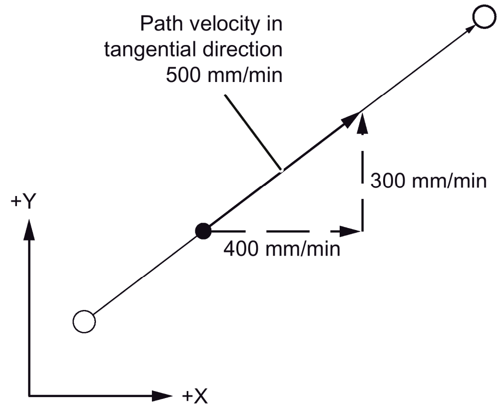
Programming example
G03 U...W...I...F0.2;
G95 S1000 (r/min);



Milling

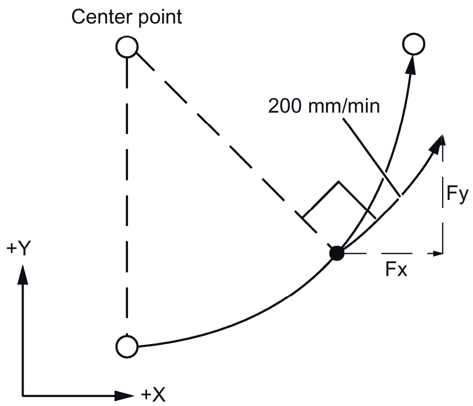
Linear interpolation with two axes:

Programming example with the
following program:
G91 (Incremental dimensioning)
G01 X40. Y30. F500.



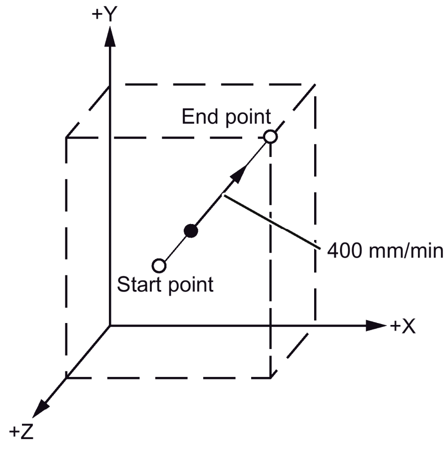
Circular interpolation with two axes:

Programming example with the
following program:
G91 (Incremental dimensioning)
G03 X ... Y ... I ... F200;



In 3D interpolation, the feed of the resulting straight lines programmed with F are maintained in the space.

Programming example with
the following program:
G01 X... Y... Z... F400



Note

If "F0" is programmed and the function "Fixed feedrates" is not activated in the block, then the alarm 14800 "Channel %1 Set %2 programmed path velocity is less than or equal to zero" is given off.

2.2.3 Linear feed (G98/G94)

On specifying G98 for turning/G94 for milling, the feed given after the address character F is executed in the mm/min, inch/min or degree/min unit.

2.2.4 Inverse-time feed (G93) (milling only)

On specifying G93, the feed given after the address character F is executed in the 1/min unit. G93 is a modally effective G function.

Example

```
N10 G93 G1 X100 F2 ;
```

i.e., the programmed path is traversed within half a minute.

2.2.5 Revolutional feedrate (G99/G95)

On entering G99 for turning/G95 for milling, the feed is executed in the mm/revolution unit or inch/revolution related to the master spindle.

Note

All of the commands are modal. If the G feed command is switched between G98 and G99 for turning or among G93, G94, and G95 for milling, the path feed must be reprogrammed. The feed can also be specified in degree/revolution for the machining with rotary axes.

3 Turning

3.1 G code table

3.1.1 SINUMERIK ISO dialect turning mode A

Name	Index	Description	Format
01. G-Group (modal)			
G0	1	Rapid traverse	G00 X... Z... ;
G1	2	Linear motion	G01 X... Z... F... ;
G2	3	Circle/helix in clockwise direction	G02(G03) X(U)... Z(W)... I... K... (R...) F... ;
G3	4	Circle/helix in counterclockwise direction	
G32	5	Thread cutting with constant lead	G32 X (U)... Z (W)... F... ;
G90	6	Longitudinal turning cycle	G.. X... Z... F...
G92	7	Thread cutting cycle	G... X... Z... F... ;
G94	8	Radial cutting cycle	G... X... Z... F... ;
G34	9	Thread cutting with variable lead	G34 X (U)... Z (W)... F... K... ;
02. G-Group (modal)			
G96	1	Constant cutting rate on	G96 S...
G97	2	Constant cutting rate off	G97 S...
04. G-Group (modal)			
G68	1	Dual slide/turret on	
G69	2	Dual slide/turret off	
05. G-Group (modal)			
G98	1	Linear feed	
G99	2	Revolutional feedrate	
06. G-Group (modal)			
G20	1	Input system inch	
G21	2	Input system metric	
07. G-Group (modal)			
G40	1	Deselection of cutter radius compensation	
G41	2	Compensation left of contour	
G42	3	Offset to right of contour	
08. G-Group (modal)			
10. G-Group (modal)			
G80	1	Drilling cycle off	G80;
G83	2	Front face deep hole drilling	G83 X(U)... C(H)... Z(W)... R... Q... P... F... M... ;
G84	3	Front face tapping	G84 X(U)... C(H)... Z(W)... R... P... F... M... K... ;
G85	4	Front face drilling cycle	G85 X(U)... C(H)... Z(W)... R... P... F... K... M... ;
G87	5	Side surface deep hole drilling	G87 Z(W)... C(H)... X(U)... R... Q... P... F... M... ;
G88	6	Side surface tapping	G88 Z(W)... C(H)... X(U)... R... P... F... M... K... ;
G89	7	Side drilling	G89 Z(W)... C(H)... X(U)... R... P... F... K... M... ;
12. G-Group (modal)			
G66	1	Macro module call	G66 P... L... <Parameters>;
G67	2	Delete macro-modal call	G67 P... L... <Parameters>;
14. G-Group (modal)			

Name	Index	Description	Format
G54	1	Selecting work offset	
G55	2	Selecting work offset	
G56	3	Selecting work offset	
G57	4	Selecting work offset	
G58	5	Selecting work offset	
G59	6	Selecting work offset	
16. G-Group (modal)			
G17	1	XY plane	
G18	2	ZX plane	
G19	3	YZ plane	
18. G-Group (non-modal)			
G4	1	Dwell time in [s] or spindle revolutions	G04 X...; or G04 P...;
G10	2	Write work offset/tool offset	G10 L2 Pp X... Z... ;
G28	3	1. Reference point approach	G28 X... Z... ;
G30	4	2./3./4. Reference point approach	G30 Pn X... Z... ;
G31	5	Delete measuring with distance-to-go	G31 X... Y... Z... F_;
G52	6	Programmable work offset	
G65	7	Macro call	G65 P_ L_ ;
G70	8	Finishing cycle	G70 P... Q... ;
G71	9	Stock removal cycle, longitudinal axis	G71 U... R... ;
G72	10	Stock removal cycle, transverse axis	G72 W... R... ;
G73	11	Closed cutting cycle	G73 U... W... R... ;
G74	12	Multiple repetitive grooving cycles in the longitudinal axis	G74 R... ;
G75	13	Multiple repetitive grooving cycles in the transverse axis	G75 R... ; or G75 X(U)... Z(W)... P... Q... R... F... ;
G76	14	Multiple thread cutting cycle	G76 P... (m, r, a) Q... R... ;
G50	15	Set actual value	G92 (G50) X... Z... ;
G27	16	Referencing check (under development)	G27 X... Z... ;
G53	17	Approach position in machine coordinate system	G53 X... Z... ;
G07.1	18	Cylindrical interpolation	G07.1 A (B, C) r ; G07.1 A (B, C) 0 ;
G5	20	High-speed cycle cutting	G05 Pxxxxx Lxxx ;
G30.1	21	Reference point position	-
G5.1	22	High-speed cycle -> Call of CYCLE305	-
G50.3	23	Delete actual value, reset the WCS	
G60	24	directed positioning	
21. G-Group (modal)			
G13.1 1)	1	TRANSMIT OFF	
G12.1	2	TRANSMIT ON	
31. G-Group (modal)			
G290	1	Select Siemens mode	-
G291	2	Select ISO dialect mode	-

3.1.2 SINUMERIK ISO dialect turning mode B

Name	Index	Description	Format
01. G-Group (modal)			
G0	1	Rapid traverse	G00 X... Y... Z... ;
G1	2	Linear motion	G01 X... Z... F... ;
G2	3	Circle/helix in clockwise direction	G02(G03) X(U)... Z(W)... I... K... (R...) F... ;
G3	4	Circle/helix in counterclockwise direction	
G33	5	Thread cutting with constant lead	G33 X (U)... Z (W)... F... ;
G77	6	Longitudinal turning cycle	G.. X... Z... F...
G78	7	Thread cutting cycle	G... X... Z... F... ;
G79	8	Face turning cycle	G... X... Z... F... ;
G34	9	Thread cutting with variable lead	G34 X (U)... Z (W)... F... K... ;
02. G-Group (modal)			
G96	1	Constant cutting rate on	G96 S...
G97	2	Constant cutting rate off	G97 S...
03. G-Group (modal)			
G90	1	Absolute programming	
G91	2	Incremental programming	
04. G-Group (modal)			
G68	1	Dual slide/turret on	
G69	2	Dual slide/turret off	
05. G-Group (modal)			
G94	1	Linear feedrate in [mm/min, inch/min]	
G95	2	Revolutional feedrate in [mm/rev, inch/rev]	
06. G-Group (modal)			
G20	1	Input system inch	
G21	2	Input system metric	
07. G-Group (modal)			
G40	1	Deselection of cutter radius compensation	
G41	2	Compensation left of contour	
G42	3	Offset to right of contour	
08. G-Group (modal)			
10. G-Group (modal)			
G80	1	Drilling cycle off	G80;
G83	2	Front face deep hole drilling	G83 X(U)... C(H)... Z(W)... R... Q... P... F... M... ;
G84	3	Front face tapping	G84 X(U)... C(H)... Z(W)... R... P... F... M... K... ;
G85	4	Front face drilling cycle	G85 X(U)... C(H)... Z(W)... R... P... F... K... M... ;
G87	5	Side surface deep hole drilling	G87 Z(W)... C(H)... X(U)... R... Q... P... F... M... ;
G88	6	Side surface tapping	G88 Z(W)... C(H)... X(U)... R... P... F... M... K... ;
G89	7	Side drilling	G89 Z(W)... C(H)... X(U)... R... P... F... K... M... ;
11. G-Group (modal)			
G98	1	Return to starting point in drilling cycles	
G99	2	Return to point R in drilling cycles	
12. G-Group (modal)			
G66	1	Macro module call	G66 P... L... <Parameters>;
G67	2	Delete macro-modal call	G67 P... L... <Parameters>;

Name	Index	Description	Format
14. G-Group (modal)			
G54	1	Selecting work offset	
G55	2	Selecting work offset	
G56	3	Selecting work offset	
G57	4	Selecting work offset	
G58	5	Selecting work offset	
G59	6	Selecting work offset	
16. G-Group (modal)			
G17	1	XY plane	
G18	2	ZX plane	
G19	3	YZ plane	
18. G-Group (non-modal)			
G4	1	Dwell time in [s] or spindle revolutions	G04 X...; or G04 P...;
G10	2	Write work offset/tool offset	G10 L2 Pp X... Z... ;
G28	3	1. Reference point approach	G28 X... Z... ;
G30	4	2./3./4. Reference point approach	G30 Pn X... Z... ;
G31	5	Delete measuring with distance-to-go	G31 X... Y... Z... F_;
G52	6	Programmable work offset	
G65	7	Macro call	G65 P_ L_ ;
G70	8	Finishing cycle	G70 P... Q... ;
G71	9	Stock removal cycle, longitudinal axis	G71 U... R... ;
G72	10	Stock removal cycle, transverse axis	G72 W... R... ;
G73	11	Closed cutting cycle	G73 U... W... R... ;
G74	12	Multiple repetitive grooving cycles in the longitudinal axis	G74 R... ;
G75	13	Multiple repetitive grooving cycles in the transverse axis	G75 R... ; or G75 X(U)... Z(W)... P... Q... R... F... ;
G76	14	Multiple thread cutting cycle	G76 P... (m, r, a) Q... R... ;
G92	15	Set actual value	G92 (G50) X... Z... ;
G27	16	Referencing check (under development)	G27 X... Z... ;
G53	17	Approach position in machine coordinate system	(G90) G53 X... Z... ;
G07.1	18	Cylindrical interpolation	G07.1 A (B, C) r ; G07.1 A (B, C) 0 ;
G5	20	High-speed cycle cutting	G05 Pxxxxx Lxxx ;
G30.1	21	Reference point position	
G5.1	22	High-speed cycle -> Call of CYCLE305	
G92.1	23	Delete actual value, reset the WCS	
G60	24	directed positioning	
21. G-Group (modal)			
G13.1[1)]	1	TRANSMIT OFF	
G12.1	2	TRANSMIT ON	
31. G-Group (modal)			
G290	1	Select Siemens mode	
G291	2	Select ISO dialect mode	

3.1.3 SINUMERIK ISO dialect turning mode C

Name	Index	Description	Format
01. G-Group (modal)			
G0	1	Rapid traverse	G00 X... Y... Z... ;
G1	2	Linear motion	G01 X... Z... F... ;
G2	3	Circle/helix in clockwise direction	G02(G03) X(U)... Z(W)... I... K... (R...) F... ;
G3	4	Circle/helix in counterclockwise direction	
G33	5	Thread cutting with constant lead	G33 X (U)... Z (W)... F... ;
G20	6	Longitudinal turning cycle	G... X... Z... R... F... ;
G21	7	Thread cutting cycle	G... X... Z... F... ;
G24	8	Face turning cycle	G... X... Z... F... ;
G34	9	Thread cutting with variable lead	G34 X (U)... Z (W)... F... K... ;
02. G-Group (modal)			
G96	1	Constant cutting rate on	G96 S...
G97	2	Constant cutting rate off	G97 S...
03. G-Group (modal)			
G90	1	Absolute programming	
G91	2	Incremental programming	
04. G-Group (modal)			
G68	1	Dual slide/turret on	
G69	2	Dual slide/turret off	
05. G-Group (modal)			
G94	1	Linear feedrate in [mm/min, inch/min]	
G95	2	Revolutional feedrate in [mm/rev, inch/rev]	
06. G-Group (modal)			
G70	1	Input system inch	G70 P... Q... ;
G71	2	Input system metric	G71 U... R... ;
07. G-Group (modal)			
G40	1	Deselection of cutter radius compensation	
G41	2	Compensation left of contour	
G42	3	Offset to right of contour	
08. G-Group (modal)			
10. G-Group (modal)			
G80	1	Drilling cycle off	G80;
G83	2	Front face deep hole drilling	G83 X(U)... C(H)... Z(W)... R... Q... P... F... M... ;
G84	3	Front face tapping	G84 X(U)... C(H)... Z(W)... R... P... F... M... K... ;
G85	4	Front face drilling cycle	G85 X(U)... C(H)... Z(W)... R... P... F... K... M... ;
G87	5	Side surface deep hole drilling	G87 Z(W)... C(H)... X(U)... R... Q... P... F... M... ;
G88	6	Side surface tapping	G88 Z(W)... C(H)... X(U)... R... P... F... M... K... ;
G89	7	Side drilling	G89 Z(W)... C(H)... X(U)... R... P... F... K... M... ;
11. G-Group (modal)			
G98	1	Return to starting point in drilling cycles	
G99	2	Return to point R in drilling cycles	
12. G-Group (modal)			
G66	1	Macro module call	G66 P... L... <Parameters>;
G67	2	Delete macro-modal call	G67 P... L... <Parameters>;

Name	Index	Description	Format
14. G-Group (modal)			
G54	1	Selecting work offset	
G55	2	Selecting work offset	
G56	3	Selecting work offset	
G57	4	Selecting work offset	
G58	5	Selecting work offset	
G59	6	Selecting work offset	
16. G-Group (modal)			
G17	1	XY plane	
G18	2	ZX plane	
G19	3	YZ plane	
18. G-Group (non-modal)			
G4	1	Dwell time in [s] or spindle revolutions	G04 X...; or G04 P...;
G10	2	Write work offset/tool offset	G10 L2 Pp X... Z... ;
G28	3	1. Reference point approach	G28 X... Z... ;
G30	4	2./3./4. Reference point approach	G30 Pn X... Z... ;
G31	5	Delete measuring with distance-to-go	G31 X... Y... Z... F_;
G52	6	Programmable work offset	
G65	7	Macro call	G65 P_ L_ ;
G72	8	Finishing cycle	G72 P... Q... ;
G73	9	Stock removal cycle, longitudinal axis	G73 U... R... ;
G74	10	Stock removal cycle, transverse axis	G74 W... R... ;
G75	11	Contour repetition	G75 U... W... R... ;
G76	12	Multiple repetitive grooving cycles in the longitudinal axis	G76 R... ;
G77	13	Multiple repetitive grooving cycles in the transverse axis	G77 R... ; or G77 X(U)... Z(W)... P... Q... R... F... ;
G78	14	Multiple thread cutting cycle	G78 P... (m, r, a) Q... R... ;
G92	15	Set actual value	G92 (G50) X... Z... ;
G27	16	Referencing check (under development)	G27 X... Z... ;
G53	17	Approach position in machine coordinate system	(G90) G53 X... Z... ;
G07.1	18	Cylindrical interpolation	G07.1 A (B, C) r ; G07.1 A (B, C) 0 ;
G5	20	High-speed cycle cutting	G05 Pxxxxx Lxxx ;
G30.1	21	Reference point position	
G5.1	22	High-speed cycle -> Call of CYCLE305	
G92.1	23	Delete actual value, reset the WCS	
G60	24	directed positioning	
21. G-Group (modal)			
G13.1[1)]	1	TRANSMIT OFF	
G12.1	2	TRANSMIT ON	
31. G-Group (modal)			
G290	1	Select Siemens mode	
G291	2	Select ISO dialect mode	

3.2 Travel commands

3.2.1 Interpolation commands

The positioning and interpolation commands, with which the tool path along the programmed contour, such as a straight line or a circular arc, is monitored, are described in the next Section.

3.2.1.1 Rapid traverse (G00)

You can use the rapid traverse movements to position the tool rapidly, to travel around the workpiece or to approach tool change points.

The following G functions can be used for positioning:

G function	Function	G group
G00	Rapid traverse	01
G01	Linear motion	01
G02	Circle/helix in clockwise direction	01
G03	Circle/helix in counterclockwise direction	01

Positioning with (G00)

Format

G00 X... Z... ;

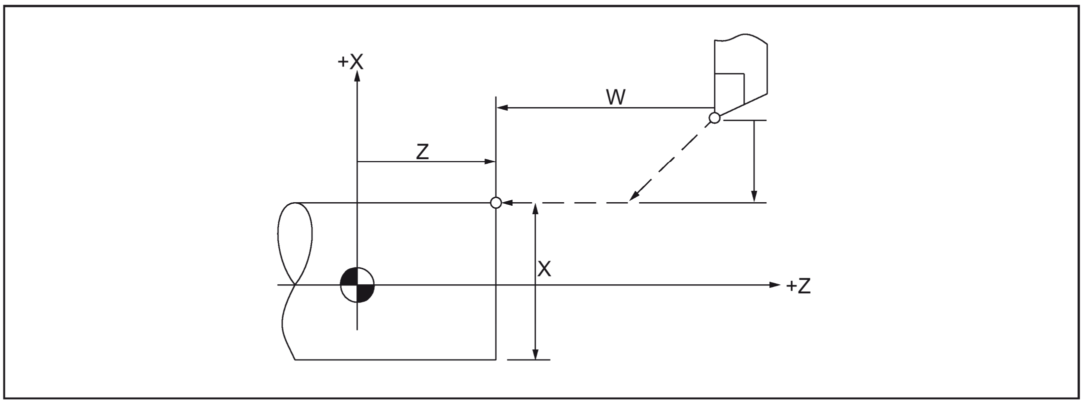
G00 with linear interpolation

The tool movement programmed with G00 is executed at the highest possible travel speed (rapid traverse). The rapid traverse rate is defined separately for each axis in the machine data. If the rapid traverse movement is executed simultaneously on several axes, the rapid traverse rate in case of linear interpolation is determined by the axis, which requires the most time for its section of the path.

G00 without linear interpolation

Axes not programmed in the G00 block are also not traversed. During positioning, the individual axes traverse with the rapid traverse rate specified for each axis independent of one another. The exact speeds for your machine are given in the documentation of the machine manufacturer.

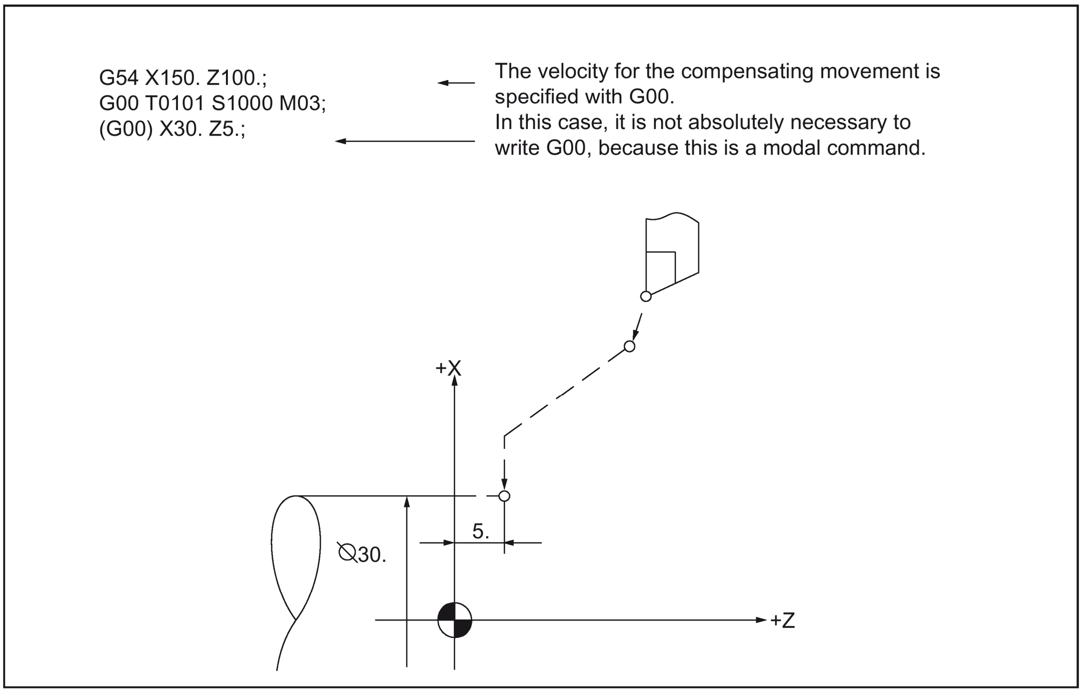
Rapid traverse with 2 non-interpolating axes:



Note

Since during positioning with G00 the axes traverse independent of one another (do not interpolate), each axis reaches its end point at a different point of time. Therefore, you must proceed extremely careful when positioning several axes, so that a tool does not collide with a workpiece or a device during positioning.

Programming example:



Linear interpolation (G00)

Hence all programmed axes traverse in rapid traverse with linear interpolation and reach their target position simultaneously.

3.2.1.2 Linear interpolation (G01)

With G01 the tool travels on axially parallel, inclined or straight lines arbitrarily positioned in the space. Linear interpolation permits machining of 3D surfaces, grooves, etc.

Format

G01 X... Z... F... ;

In case of G01 the linear interpolation is executed with the path feed. The axes not specified in the block with G01 are also not traversed. The linear interpolation is programmed as given in the example above.

Feed F for path axes

The feedrate is specified under the address F. Depending on the default setting in the machine data, the units of measurement specified with the G commands (G98, G99) are either in mm or inch.

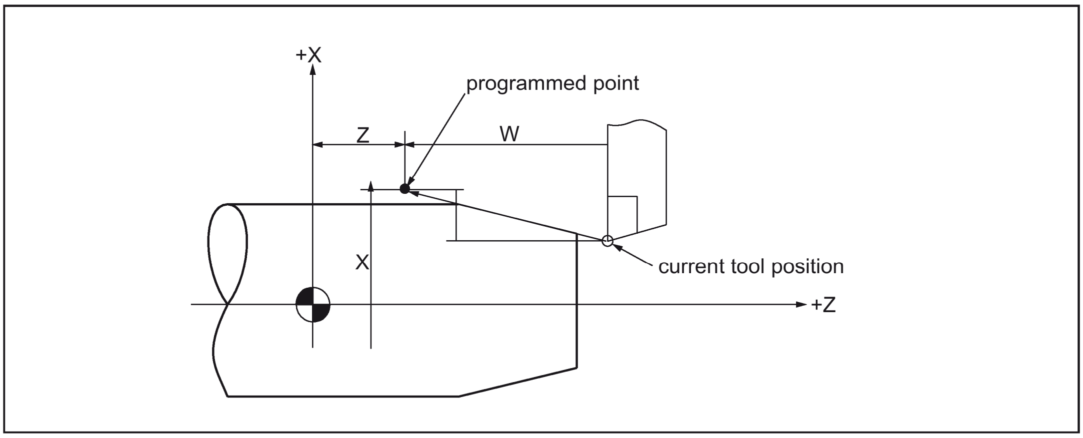
One F value can be programmed per NC block. The unit of the feedrate is defined via one of the G commands mentioned. The feed F acts only on path axes and remains active until a new feed value is programmed. Separators are permitted after address F.

Note

If no feed Fxx is programmed in a block with G01 or in the previous blocks, an alarm is given off when a G01 block is executed.

The end point can be specified either as absolute or incremental. For details see Section "Absolute/incremental dimensioning (G90, G91) (Page 106)".

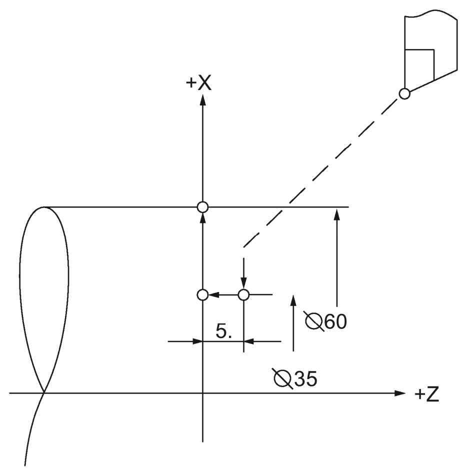
Linear interpolation:



Programming example:

```
Programming example
G54 X100. Z60.;
G00 T0202 S600 M03;
      X35. Z5.;
G01 Z0 F1.;
      X60. F0.2;
```

The axes are traversed in the linear interpolation mode (G01).



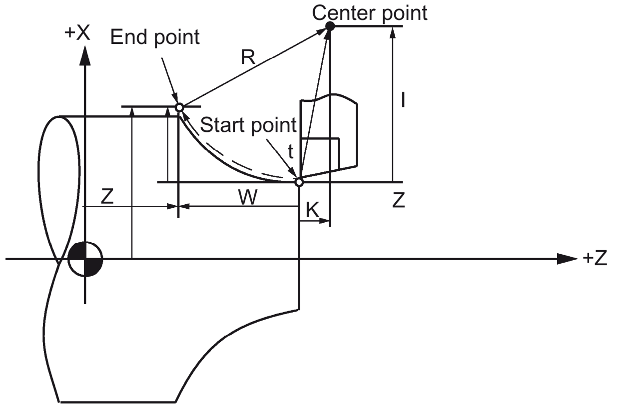
3.2.1.3 Circular interpolation (G02, G03)

Format

With the command given below the rotary tool traverses in the ZX plane on the programmed circular arc. The programmed path velocity is thereby maintained along the arc.

G02(G03) X(U)... Z(W)... I... K... (R...) F... ;

Circular interpolation:



In order to start the circular interpolation, the commands given in the following table are to be executed:

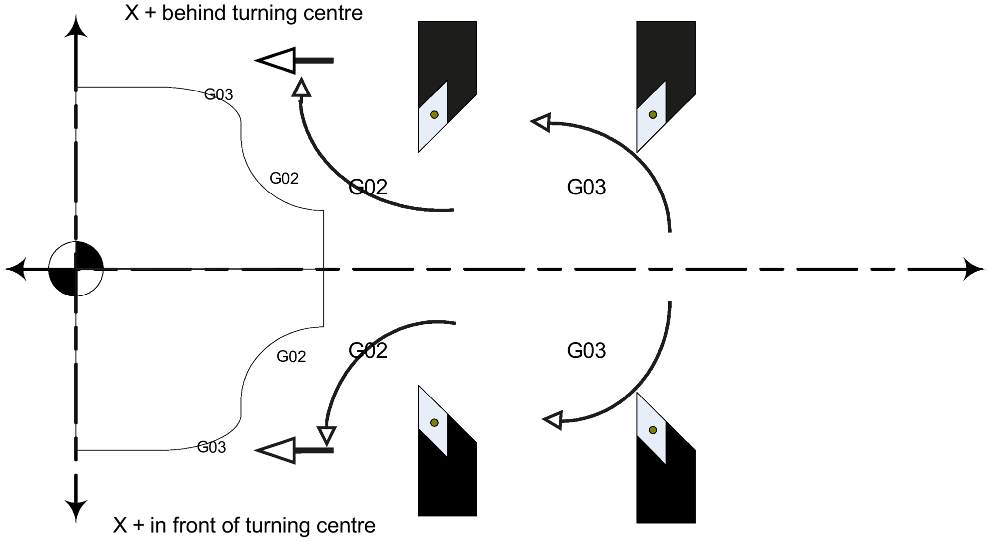
Element	Command	Description
Direction of rotation	G02	clockwise
	G03	Counterclockwise
End-point position	X (U)	X coordinate of the arc end point (diametric value)
	Z (W)	Z coordinate of the arc end point
	Y (V)	Y coordinate of the arc end point
Distance start point - center point	I	Distance of the start point to the center point of the arc in the X-axis
	J	Distance of the start point to the center point of the arc in the Y-axis
	K	Distance of the start point to the center point of the arc in the Z-axis
Radius of arc	R	Distance of the start point to the center of arc

Direction of rotation

The direction of rotation of the arc is defined with the G functions listed in the table given below:

Direction of rotation	
G02	clockwise
G03	Counterclockwise

Direction of rotation of the arc:



Programming of circular movements

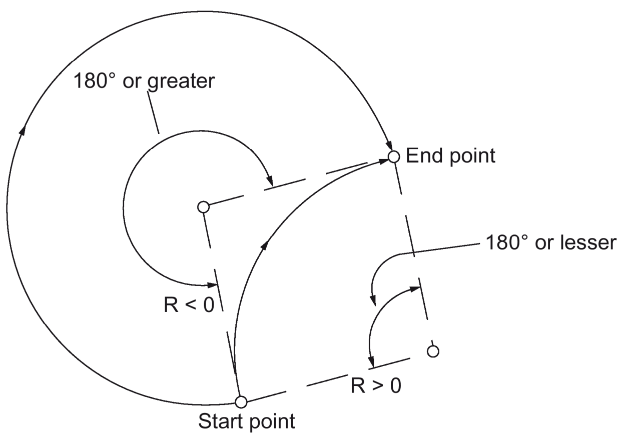
ISO mode offers two possibilities for programming the circular movements.

The circular motion is described by the:

- Center point and end point in the absolute or incremental dimension
- Radius and end point in Cartesian coordinates
  For a circular interpolation with a central angle <= 180 degree, "R > 0" (positive) is to be programmed.
  For a circular interpolation with a central angle > 180 degree, "R < 0" (negative) is to be programmed.

Circular interpolation with the specification of the radius R:

Programming example:
G02 X(U) ... Z(W) ... R ± ... F ...

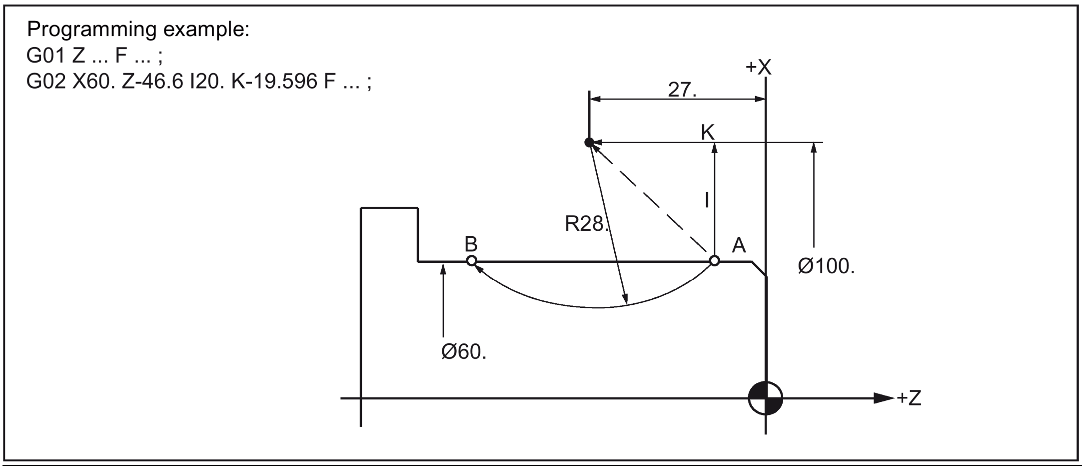


Feed

In the circular interpolation, the feed is programmed in the same way as for the linear interpolation (see Section "Linear interpolation (G01) (Page 22)").

Programming example

Circular interpolation over several quadrants:



Center of arc	(100.00, 27.00)
Value of "I"	$\frac{100 - 60}{2}$ = 20 mm
Value of "K"	$- \sqrt{28^2 - 20^2} = - \sqrt{384} = -19.596$ mm

3.2.1.4 Contour definition programming and insertion of chamfers and radii

Chamfers or radii can be inserted after each traversing block between linear and circular contours, for instance for deburring the sharp workpiece edges.

The following combinations are possible during insertion:

- between two straight lines
- between two arcs
- between an arc and a straight line
- between a straight line and an arc

Format

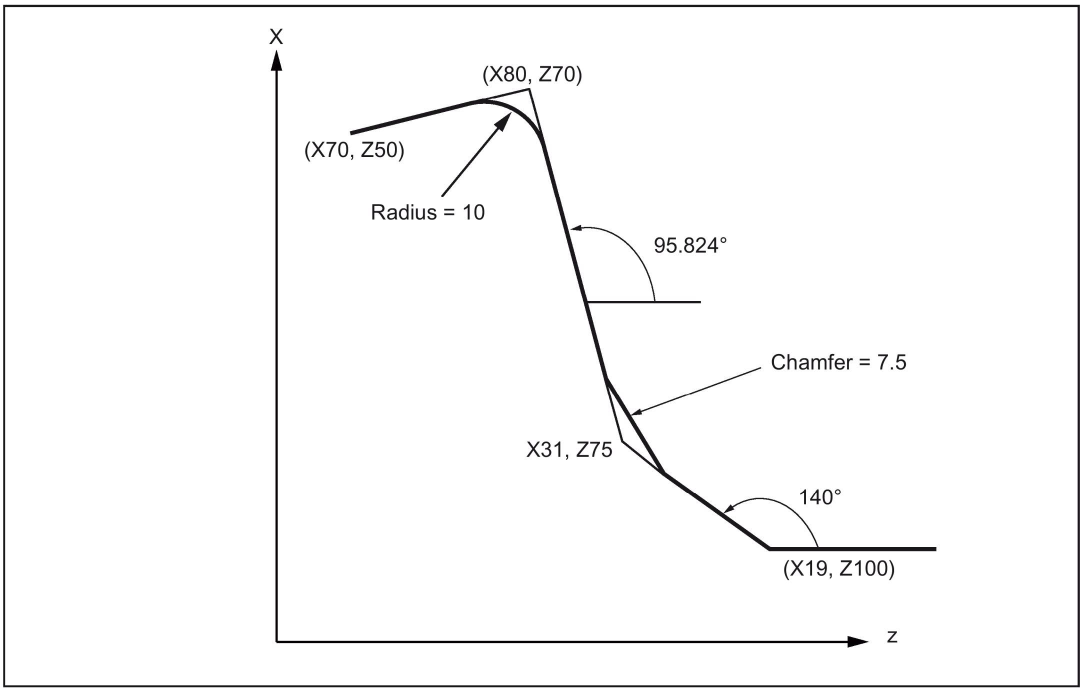
, C...; Chamfer

, R...; Rounding

Example

```
N10 G1 X9. Z100. F1000 G18
G1 X19 Z100
X31, A140, C7.5
N30 X80. Z70., A95.824, R10
```

3 straight lines:



ISO dialect mode

In the original ISO Dialect, the address C can be used as axis name as well as for naming a chamfer on the contour.

The address R can either be a cycle parameter or an identifier for the radius in a contour.

For differentiating between these two possibilities, a comma "," must be put before the address "R" or "C" while programming the contour definition.

3.2.1.5 Cylindrical interpolation (G07.1)

With the function of cylindrical interpolation, any number of traversing grooves can be machined at the cylindrical workpieces. The path of the grooves is programmed with reference to the unwrapped, level surface of the cylinder. The cylindrical interpolation is started with G07.1 with the specification of the cylinder radius (G07.1 C<cylinder radius>) and ended with G07.1 (radius 0). A programming is possible with absolute commands (C, Z) as well as also with incremental commands (H, W).

The following G function is used for the cylindrical interpolation:

G function	Function	G group
G07.1	Operation with cylindrical interpolation	18

Format

G07.1 A (B, C) r ;Activation of the operation with cylindrical interpolation

G07.1 A (B, C) 0 ;Deactivation of the operation with cylindrical interpolation

A, B, C: Address for the rotary axis

r: Radius of the cylinder

No further commands may be present in the block with G07.1.

The command G07.1 is modal. When G07.1 is specified once, the cylindrical interpolation remains active for so long till G07.1 A0 (B0, C0) is deselected. In case of closed position or after NC RESET the cylindrical interpolation gets deactivated.

Note

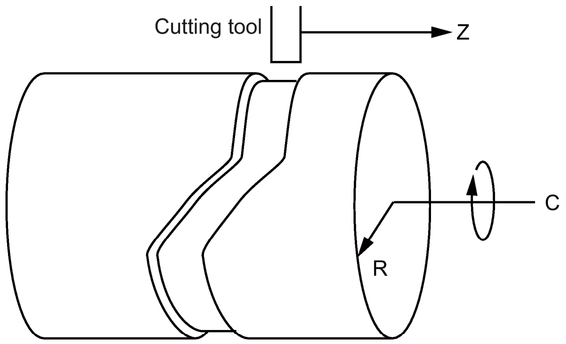
Cylindrical interpolation (G07.1)

- G07.1 is based on the Siemens option TRACYL. For this, the corresponding machine data are to be set.
- The corresponding specifications for this are given in the SINUMERIK 808D/SINUMERIK 808D ADVANCED Function Manual, Chapter "Kinematic Transformation (M1)".

The rotary axis for the cylindrical interpolation and also its name are defined with the machine data 24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1.

Example

Programming example for the cylindrical interpolation:



G00 X44.0 C0;	Positioning at the start point of the cut machining
G07.1 C45.0;	Cylindrical interpolation mode ON
G01 G42 Z47.5 F100 C60.0; Z32.5 120.0; C240.0; G03 Z40.0 C249.549 R7.5; G02 Z47.5 C259.099 R7.5; G01 C360.0; Z44.0 G07.1 C0;	Machining program
M30	Cylindrical interpolation mode ON

3.2.1.6 Polar coordinates interpolation (G12.1, G13.1) (TRANSMIT)

With G12.1 and G13.1 an interpolation is activated and deactivated in the machining plane between a rotary axis and a linear axis. A further linear axis is present perpendicular to this plane.

This function corresponds to the TRANSMIT function in Siemens mode. For G12.1 the machine data of the 2nd transformation record must be parameterized.

Properties of G12.1 and G13.1

The polar coordinates interpolation mode is activated/deactivated with the following G functions:

G function	Function	G group
G12.1	Operation with polar coordination interpolation ON	21
G13.1	Operation with polar coordination interpolation OFF	21

The commands G12.1 and G13.1 may not be programmed together with other commands in a block.

The commands G12.1 and G13.1 act modally and belong to the G group 21. In case of G12.1 the polar coordinates interpolation remains active during the process that G13.1 is programmed. Closed position or after NC RESET G13.1 is active (polar coordinates interpolation is deactivated).

Restrictions during the selection

- An intermediate traversing block is not inserted (chamfers/radii).
- A spline block sequence must be concluded.
- An active tool-length offset must be deselected.
- An activated tool radius compensation is incorporated into the transformation in the geometry axis by the control system.
- The frame that was active before TRANSMIT is deselected by the control system (corresponds to programmed frame reset G500 in Siemens mode).
- An active working area limitation is deselected by the control system for the axes affected by the transformation (corresponds to programmed WALIMOF in Siemens mode).
- Continuous path mode and rounding are interrupted.
- Possibly active DRF offsets in transformed axes must have been deleted by the operator.
- No geometry axis interchange may be active (parallel axes with G17 (G18, G19)).

Restrictions for the polar coordinates interpolation

- Tool change:

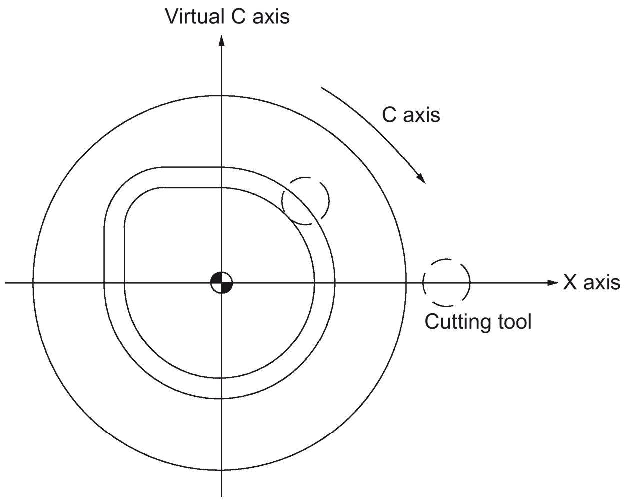
  Before a tool change, a tool radius compensation must be deselected!
- Work offset:

  All statements, which refer exclusively to the basic coordinate system, are permissible (FRAME, tool radius compensation). Unlike the procedure for inactive transformation, however, a frame change with G91 (incremental dimension) is not specially treated. The increment to be traversed is evaluated in the workpiece coordinate system of the new frame – regardless of which frame was effective in the previous block.
- Rotary axis:

  The rotary axis cannot be programmed as it is occupied by a geometry axis and thus cannot be programmed directly as channel axis.

Programming example

Coordinates system for the polar coordinates interpolation:



Programming example:

```
G291;
G94;
T0101;
G00 X120.0 C0;
G12.1;
G01 G42 X40.0 I-20.0;
G01 X-25.0;
G03 X.40.0 C25.0 K-15.0;
G01 C0;
G03 X20.0 120.0;
G01 G40 X120.0;
G13.1;
M30;
```

- T0101; ← Positioning at the start point of the cut machining
- G00 X120.0 C0; ← Polar coordinates interpolation mode ON
- G01 G42 X40.0 I-20.0; … G01 G40 X120.0; } Machining program with polar coordinates interpolation
- M30; ← Polar coordinates interpolation mode OFF

References:

SINUMERIK 808D/SINUMERIK 808D ADVANCED Function Manual, Chapter "Kinematic Transformation (M1)".

3.2.2 Reference point approach with G functions

3.2.2.1 Reference point approach with intermediate point (G28)

Format

G28 X... Z... ;

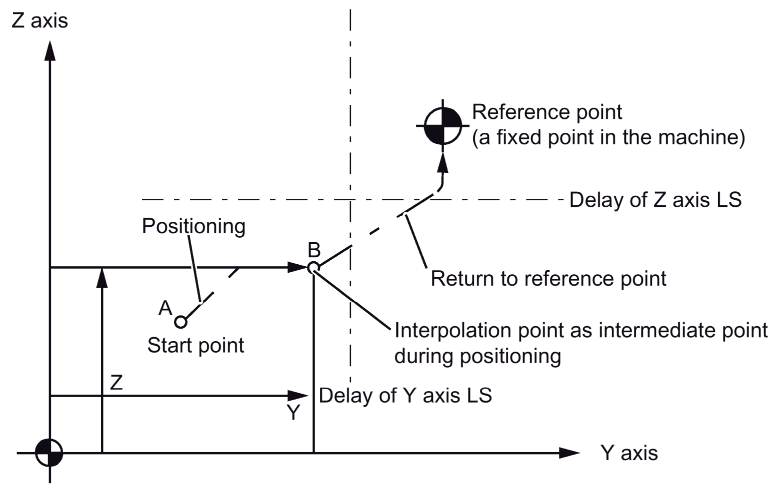
With the command "G28 X(U)...Z(W)...C(H)...Y(V);" the programmed axes can be traversed on their reference point. Thus, the programmed axes are first traversed with rapid traverse to the specified position and from there automatically to the reference point. The unprogrammed axes in the block with G28 are not traversed to their reference point.

Reference position

When the machine has been powered up (where incremental position measuring systems are used), all of the axes must approach their reference mark. Only then can traversing movements be programmed. The reference point can be approached in the NC program with G74. The reference point coordinates are set with the machine data 34100 $_MA_REFP_SET_POS[0] to [3]. A total of four reference positions can be determined.

Automatic reference point approach:

Programming example:
(G90/G91) G28 X ...Y ... Z ...



Note

The function G28 is implemented with the shell cycle cycle328.spf.

Before the reference point approach, a transformation must not be programmed for an axis which is to approach the reference point with G28. The transformation is deactivated in cycle328.spf.

3.2.2.2 Checking the reference position (G27)

Format

G27 X... Z... ;

This function is used to check whether the axes are on their reference point.

Test procedure

If the check with G27 is successful, the processing is continued with the next part program block. If one of the axes programmed with G27 is not on the reference point, Alarm 61816 "Axes not on reference point" is triggered and "AUTO" mode is interrupted.

Note

Function G27 is implemented with the cycle 328.spf as with G28.

To avoid a positioning error, the function "mirroring" should be deselected before executing G27.

3.2.2.3 Reference point approach with reference point selection (G30)

Format

G30 Pn X... Z... ;

For the commands "G30 Pn X... Z;", the axes are positioned on the specified intermediate point in the continuous-path mode, and finally traversed to the reference point selected with P2 - P4. With "G30 P3 X30.;", The X- axis returns to the third reference point. The second reference point is selected on omitting "P". Axes that are not programmed in a G30 block are also not traversed.

Reference point positions

The positions of all the reference points are always determined in relation to the first reference point. The distance of the first reference point from all subsequent reference points is set in the following machine data:

Element	MD
2. Reference point	$_MA_REFP_SET_POS[1]
3. Reference point	$_MA_REFP_SET_POS[2]
4. Reference point	$_MA_REFP_SET_POS[3]

Note

Further details of the points that were considered in the programming of G30 are available in Section "Reference point approach with intermediate point (G28) (Page 30)". Function G30 is implemented with the cycle 328.spf as with G28.

3.2.3 Use of the thread cutting function

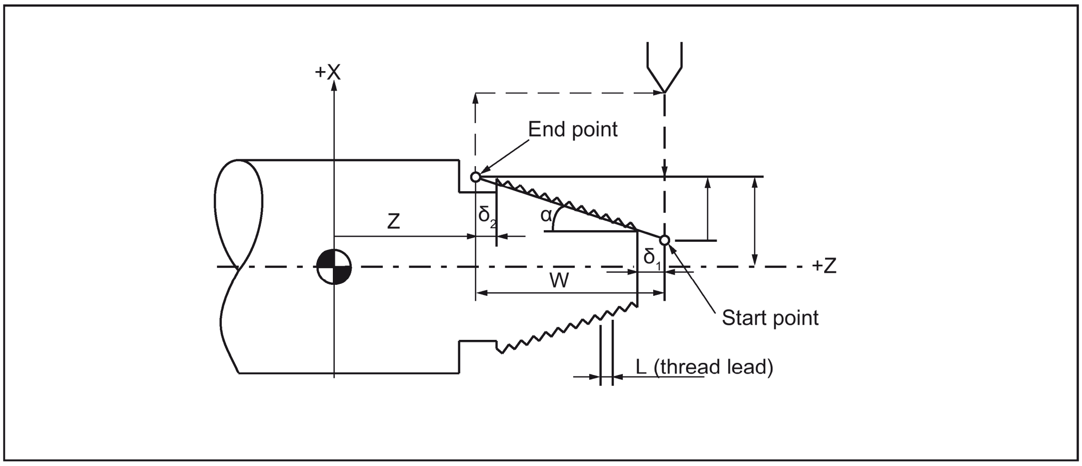
3.2.3.1 Thread cutting with constant lead (G32)

Format

With the commands "G32 X (U)... Z (W)... F... ;", the three thread types "Cylindrical thread", "Transversal thread", "Taper thread" can be prepared as right- or left-hand thread. The thread lead is defined with F. The end point coordinates are determined with X, Z (absolute) or U, W (incremental).

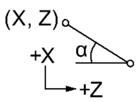
G Code System A	G Code System B	G Code System C
G32	G33	G33

Thread cutting:



Direction of thread lead

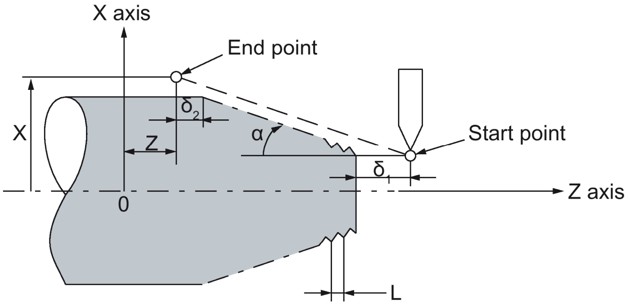
In case of taper threads, the direction in which the programmed lead is effective, depends on the taper angle. See the table below.

		Direction of thread lead
	α ≦ 45°	The programmed thread lead acts in the direction of the Z-axis.
	α > 45°	The programmed thread lead acts in the direction of the X-axis.

Example

Examples of programming:

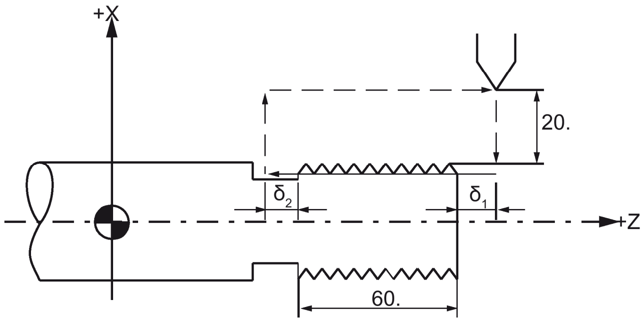
```
G33 X... Z... F...;
X, Z: End point
F_: Thread lead
(Radius axis programming)
```



Example of cutting a cylindrical thread (G code system A)

Programming example for cutting a cylindrical thread:

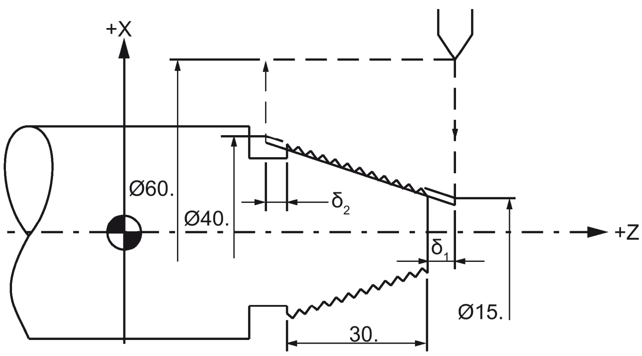
```
Thread lead L = 5.0 mm
δ1 = 5.0 mm
δ2 = 3.0 mm
Cutting depth per stroke = 1.0 mm
G00  U -42.;
G32  W -68. F5.0;
G00  U  42.;
     W 68.;
     U -44.;
G32  W -68.;
G00  U  44.;
     .
     .
     .
```



Example of cutting a taper thread (G code system A)

Programming example for cutting a taper thread:

```
Thread lead L = 4.0 mm
δ1 = 3.0 mm
δ2 = 2.0 mm
Cutting depth per stroke = 1.0 mm
G00   X13.
G32   X38. W -35. F 4.0;
G00   X60.;
      W35.;
       X11.;
G32 X36. W - 35.
G00 X60.;
      .
      .
      .
```



Precondition:

The technical prerequisite is a speed-controlled spindle with position measuring system

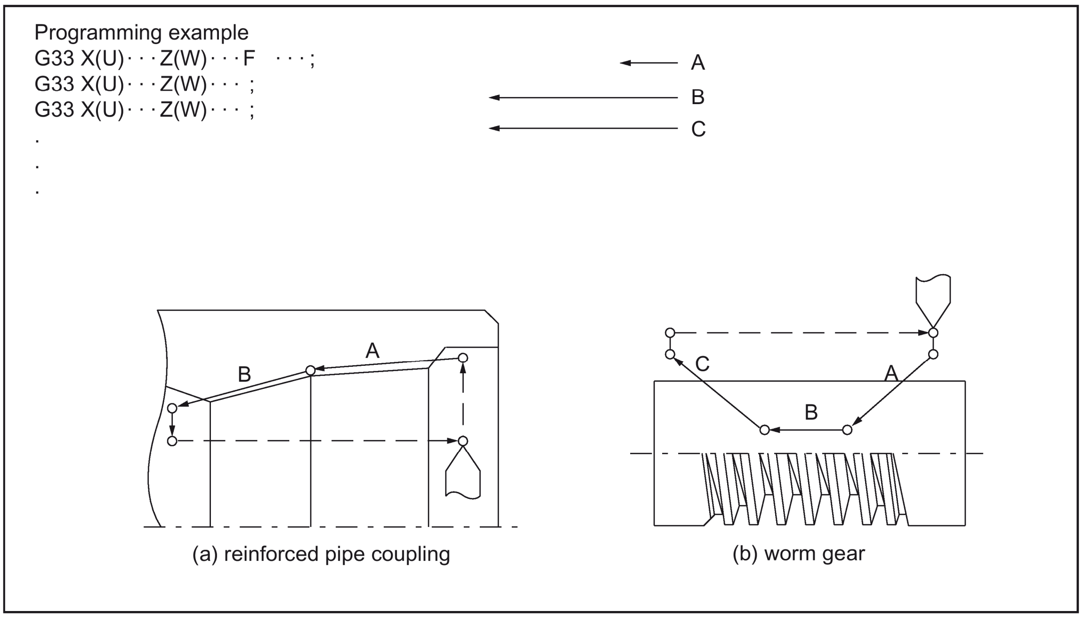
Procedure:

From the programmed spindle speed and the thread lead, the control system calculates the required feed with which the turning tool is traversed over the thread length in the longitudinal and/or transverse direction. The feed F is not taken into account for G32, the limitation to maximum axis velocity is monitored by the control system.

3.2.3.2 Interlinking of threads (G32)

The thread blocks can be arranged one after the other to make a chain via several G32 blocks programmed one after the other. With continuous-path mode G64, the blocks are linked by the look-ahead velocity control in such a way that there are no velocity jumps,

Cutting a continuous thread:



Note

As long as the thread is not completely cut, the spindle speed should not be changed! If the spindle speed is not kept constant, then there is a risk of losing the accuracy due to the servo lag.

Note

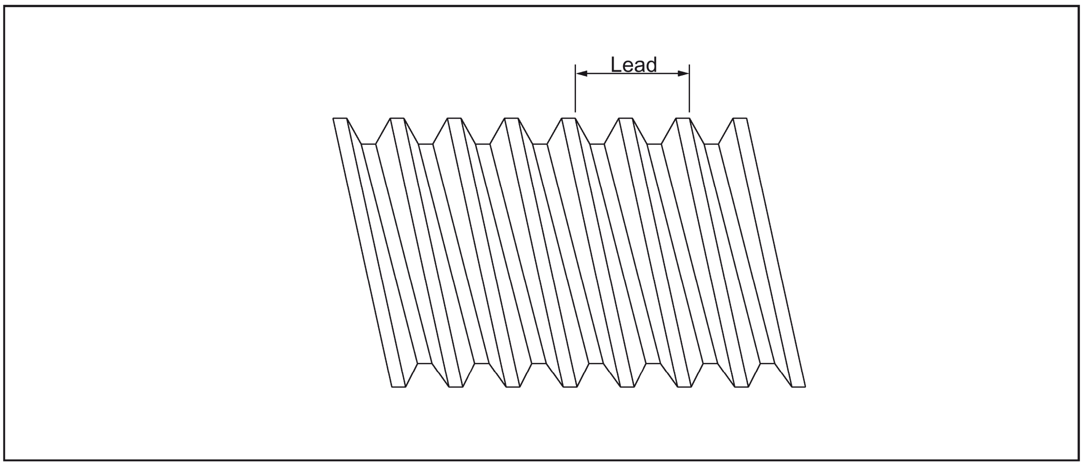
Feed control and feed stop are not taken into account during thread cutting!

If the command G32 is programmed when operating with G98 (feed per minute), an alarm is given off.

3.2.3.3 Cutting of multiple-start threads (G32)

The production of multiple-start threads is realized by the specification of start points offset to each other. The starting point offset is specified as the absolute angle position under the address Q. The related settings data 42000 ($SD_THREAD_START_ANGLE) is modified accordingly.

Double-start thread:



Format

With the commands "G32 X (U)... Z (W)... F... Q... ;", the spindle rotates by the angle specified with the address character Q after the output of the start-point pulse. Thereafter, the thread cutting starts in the direction of the end points specified with X (U) and Z (W) with the lead specified with F.

Specification of the address Q while cutting multiple-start threads:

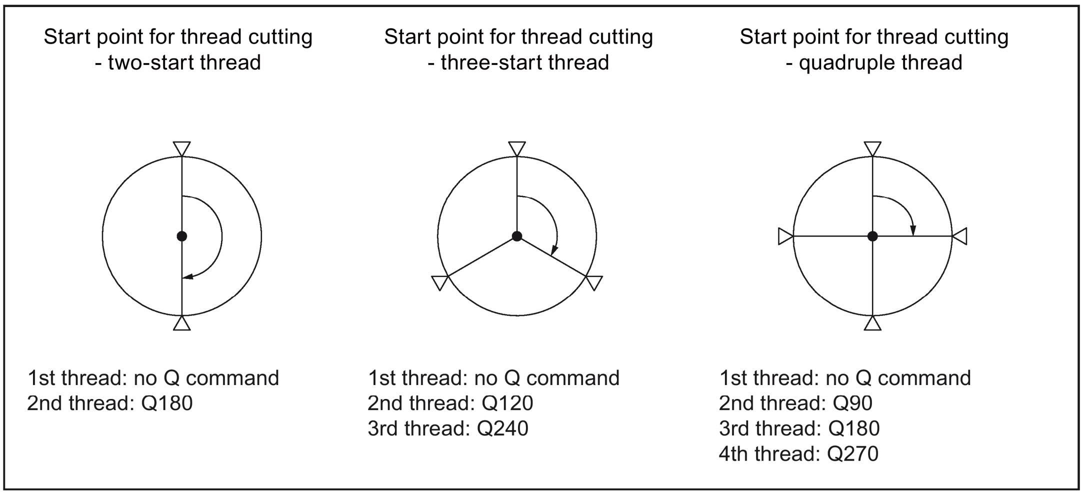
Smallest input increment: 0.001°

Programmed range: 0 ≦ B < 360.000

Calculation of the start angle in case of multiple-start threads

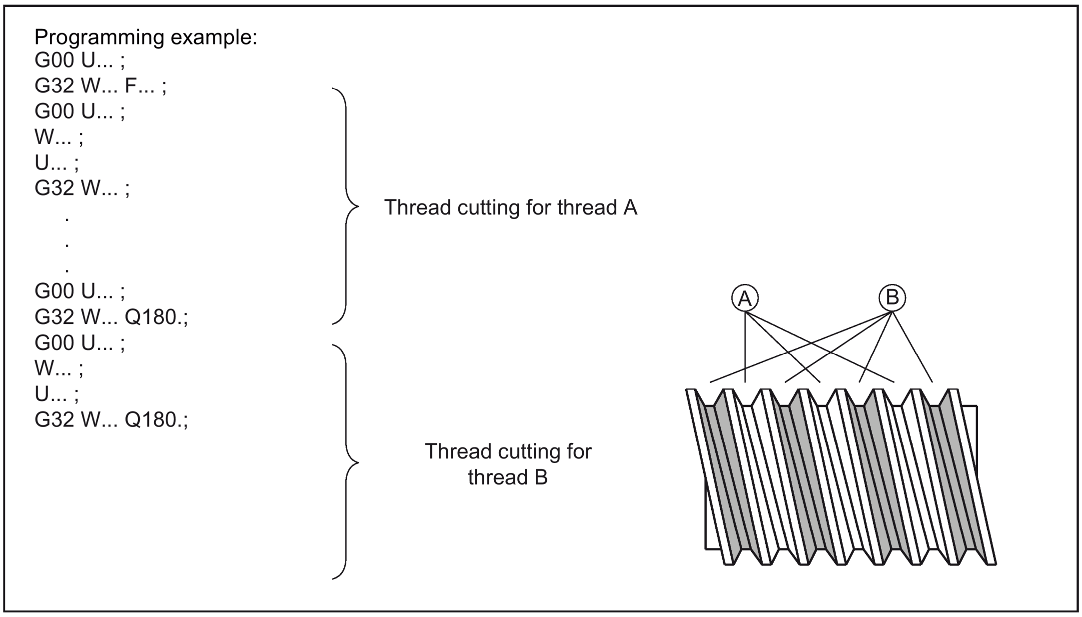
In general, the start point for the thread cutting is defined with the setting data $SD_THREAD_START_ANGLE. In case of multiple-start threads, the angular offset is calculated between the individual start points by dividing 360° by the number of threads. Examples for multiple-start threads (two-, three- and four-starts) can be seen in the following figure.

Calculation of the start angle in case of multiple-start threads:



Programming example for a multiple-start thread (G code system A)

Specification of the spindle revolution angles:



Note

If no starting point offset is specified (with Q), the "start angle for thread" defined in the setting data is used.

3.2.3.4 Thread cutting with variable lead (G34)

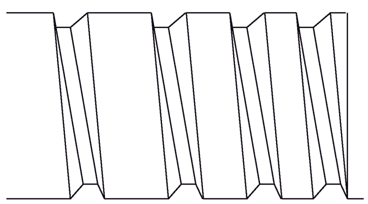
With the commands "G34 X (U)... Z (W)... F... K... ;", threads with a variable lead can be cut; the thread lead change for each spindle revolution is specified with the address K.

Format

G34 X... Z... F... K... ;

G Code System A	G Code System B	G Code System C
G34	G34	G34

Thread with variable thread lead:



Feedrate at the end point

The commands are to be given such that the feed at the end point does not have a negative value!

$$(F + \frac{K}{2})^2 + 2KW > 0$$

Calculation of the thread lead change

If you already know the starting and final lead of a thread, you can calculate the thread lead change to be programmed according to the following equation:

$$F = \frac{|k2e - k2a|}{2 * IG[mm/U]}$$

The identifiers have the following meanings:

K2e: Thread lead of axis target point coordinate in [mm/U]

K2a: Initial thread pitch (progr. under I,J and K) in [mm/U]

IG: Thread length in [mm]

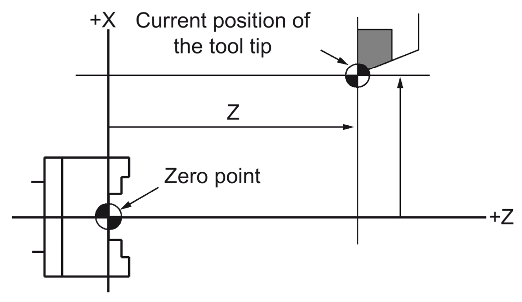
3.3 Measuring commands

3.3.1 The coordinate system

The position of a tool is defined uniquely through its coordinates in the coordinate system. These coordinates are defined by the axis positions. For instance, if the two axes involved are named as X and Z, then the coordinates are specified as follows:

X... Z...

Tool position specified with X... Z..:



The following coordinate systems are used for specifying the coordinates:

- Machine coordinate system (G53)
- Workpiece coordinate system (G50)
- Local coordinate system (G52)

3.3.1.1 Machine coordinate system (G53)

Defining the machine coordinate system

The machine coordinate system MCS is defined with the machine zero. All other reference points refer to the machine zero.

The machine zero is a fixed point of the machine tool, to which all (derived) measuring systems can be traced back.

Format

G53 X... Z... ;

X, Z: absolute dimension word

Selection of the machine coordinate system (G53)

G53 suppresses non-modal the programmable and the adjustable work offset. Traversing movements in the machine coordinate system on the basis of G53 are then programmed always when the tool is to be traversed to a machine-specific position.

Compensation deselection

If MD10760 $MN_G53_TOOLCORR = 0, the active tool length and tool nose radius compensation remain effective in a block with G53

If $MN_G53_TOOLCORR = 1, the active tool length and the tool radius compensation get suppressed in a block with G53.

3.3.1.2 Workpiece coordinate system (G50)

Before machining a coordinate system for the workpiece, the so-called workpiece coordinate system is to be created. This section describes different methods for setting, selecting and changing a workpiece coordinate system.

Setting a workpiece coordinate system

For setting a workpiece coordinate system, the following two methods can be used:

- with G50 (G92 in G code systems B and C) in the part program
- manually via the HMI operator panel

Format

G50 (G92) X... Z... ;

With G50 a coordinate transformation is programmed from the basic coordinate system (BCS) into basic zero-point system (BZS). G50 acts like an adjustable work offset.

3.3.1.3 Resetting the tool coordinate system (G50.3)

With G50.3 X.. (G Code Systems B and C with G92.1 P0), one can reset a shifted coordinate system before the shift. The tool coordinate system is reset to the coordinate system that is defined by the active adjustable work offsets (G54-G59). The tool coordinate system is set to the reference position if no adjustable work offset is active. G50.3 resets shifts carried out through G50 or G52. However, only the axes that are programmed, are reset.

Example 1:

```
N10 G0 X100 Y100          ;Display: WCS: X100 Y100          MCS: X100 Y100
N20 G50 X10 Y10           ;Display: WCS: X10 Y10            MCS: X100 Y100
N30 G0 X50 Y50            ;Display: WCS: X50 Y50            MCS: X140 Y140
N40 G50.3 X0 Y0           ;Display: WCS: X140 Y140          MCS: X140 Y140
```

Example 2:

```
N10 G10 L2 P1 X10 Y10
N20 G0 X100 Y100          ;Display: WCS: X100 Y100          MCS: X100 Y100
N30 G54 X100 Y100         ;Display: WCS: X100 Y100          MCS: X110 Y110
N40 G50 X50 Y50           ;Display: WCS: X50 Y50            MCS: X110 Y110
N50 G0 X100 Y100          ;Display: WCS: X100 Y100          MCS: X160 Y160
N60 G50.3 X0 Y0           ;Display: WCS: X150 Y150          MCS: X160 Y160
```

3.3.1.4 Selection of a workpiece coordinate system (G54 to G59)

As mentioned above, you can select one of the already set workpiece coordinate systems.

- G50

  Absolute commands function in connection with a workpiece coordinate system only if a workpiece coordinate system was selected earlier.
- Selection of a workpiece coordinate system from a selection of specified workpiece coordinate systems via the HMI operator panel

  A workpiece coordinate system can be selected by specifying a G function in the area G54 to G59.

Workpiece coordinate systems are setup after the reference point approach after Power On. The closed position of the coordinate system is set in MD20154[13].

3.3.1.5 Writing work offset/tool offsets (G10)

The workpiece coordinate systems defined through G54 to G59 or G54 P{1 ... 93} can be changed with the following two processes.

- Data inputting at HMI operator panel
- with the program commands G10 or G50 (setting actual value)

Format

Modified by G10:

G10 L2 Pp X (U)... Z(W)... ;

p=0:	External workpiece work offset
p=1 to 6:	The value of the workpiece work offset corresponds to the workpiece coordinate system G54 to G59 (1 = G54 to 6 = G59)
X, Z:	Absolute setting data of the workpiece coordinate system offset.
U, W:	Incremental setting data of the workpiece coordinate system offset.

Modified by G50:

G50 X... Z... ;

Explanations

Modified by G10:

G10 can be used to change each workpiece coordinate system individually. If the work offset with G10 is to be written only when the G10 block is executed on the machine (main run block), then MD20734 $MC_EXTERN_FUNCTION_MASK, Bit 13 must be set. An internal STOPRE is executed in that case with G10. The machine data bits affect all G10 commands in the ISO Dialect T and ISO Dialect M.

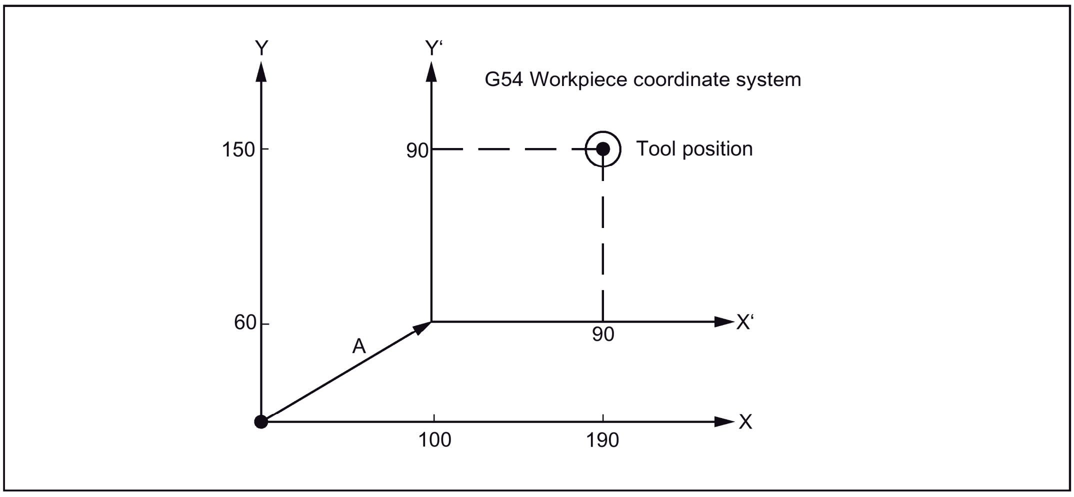
Modified by G50:

By specifying G50 X... Z..., a workpiece coordinate system that was selected earlier with a G command G54 to G59 or G54 P{1 ...93}, can be shifted and thus a new workpiece coordinate system can be set. If X and Z are programmed incrementally, the workpiece coordinate system is defined in such a way that the current tool position matches the total of the specified incremental value and the coordinates of the previous tool position (shift of coordinate system). Finally, the value of the coordinate system shift is added to each individual value of the workpiece work offset. To put it another way: All workpiece coordinate systems are shifted systematically by the same value.

Example

The tool in operation with G54 is positioned at (190, 150), and the workpiece coordinate system 1 (X' - Y') is created each time in G50X90Y90 with a shift of Vector A.

Example of setting coordinates:



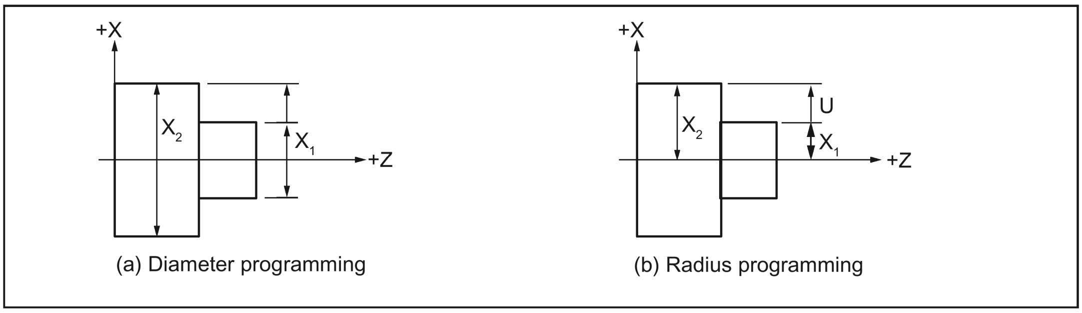
3.3.2 Defining the input modes for the coordinate values

3.3.2.1 Diameter and radius programming for the X-axis

The address X or U is used for programming the commands for the X-axis:

If the X-axis is defined as transverse axis with the machine data 20110 $MC_DIAMETER_AX_DEF = "X" and diameter programming (= Siemens G-code DIAMON) is activated with MD20150 $MC_GCODE_RESET_VALUES[28] = 2, then the programmed axis positions are interpreted as diameter values.

Coordinate values:



The diameter values apply for the following data:

- Actual value display of the transverse axis in the workpiece coordinate system
- "JOG" mode: Increments for incremental dimensions and handwheel travel
- Programming of end positions

3.3.2.2 Input inch/metric (G20, G21)

Depending upon the dimensioning in the production drawing, workpiece-related geometric axes can be programmed alternately in metric or inch dimensions. The input unit is selected with the following G functions:

G function	Function	G group
G20 (G70, G code system C)	Input in "inch"	06
G21 (G71, G code system C)	Input in "mm"	06

Format

G20 and G21 are always to be programmed at the start of a block and may not be present in the block together with other commands.

Supplements to the inch/metric-changeover

You can instruct the control system to convert the following geometrical dimensions (with necessary deviations) into the measuring system which is not set and enter them directly:

Examples

- Positional data X, Z
- Interpolation parameters I, J, K and circle radius R of the circle radius programming
- Thread lead (G32, G34)
- programmable work offset

> **Note**
>
> All other parameters such as feed rates, tool offsets or adjustable work offsets are interpreted (when using G20/G21) in the default measuring system (MD10240 SCALING_SYSTEM_IS_METRIC).
>
> Similarly, the display of system variables and machine data is also dependent on the G20G21 context. If the feed in the G20/G21 is to be activated, a new F value must be programmed explicitly.

For tool offset amounts while working with G20 or G21, see the table below:

Stored tool offset amount	when working with G20 (unit of measurement "inch")	when working with G21 (unit of measurement "mm")
150000	1.5000 inch	15.000 mm

3.3.3 Time-controlled commands (G04)

With G04 one can stop the workpiece machining between two NC blocks for a programmed time or number of spindle revolutions, e.g. for back off.

With MD20734 $MC_EXTERN_FUNCTION_MASK, for bit 2 it can be set, whether the dwell time is to be interpreted as time (s or ms) or alternatively in spindle revolutions. If for $MC_EXTERN_FUNCTION_MASK, bit 2=1 is set, then for active G98 the dwell time is interpreted in seconds; if G99 is selected, the dwell time is specified as spindle revolutions [U].

Format

G04 X...; or G04 P...;

X_: Time display (decimal points possible)

P_: Time display (no decimal points possible)

- The dwell time (G04 ..) must be programmed alone in a block.

There are two methods for executing the programmed dwell time:

MD $MC_EXTERN_FUNCTION_MASK

Bit2 = 0: Specification of the dwell time always in seconds [s]

Bit2 = 1: Specification of the dwell time either in seconds (G98 active) or in spindle revolutions (G99 active)

The processing of the next block is delayed in case of G98 (feed per minute) by a certain time (in seconds) and in case of G99 (revolutional feedrate) a certain number of spindle revolutions are awaited.

G04 must be programmed alone in a block.

Example

G98 G04 X1000 ;

Standard notation: 1000 * 0.001 = 1 second

Pocket calculator notation: 1000 seconds

G99 G04 X1000 ;

Standard notation: 1000 * 0.001 = 1 spindle revolution

Pocket calculator notation: 1000 spindle revolutions

3.3.4 Tool offset functions

While writing the program, you need not take cutting edge radius, length of cutting edge of the turning tool and the tool length into account.

You program the workpiece dimensions directly, for example following the production drawing.

While producing a workpiece, the tool geometry is taken into account automatically, so that with each tool used the programmed contour can be manufactured.

3.3.4.1 Tool offset data memory

The tool data for each tool are entered separately in the tool offset data memory of the control system. In the program you call only the required tool with its compensation data.

Contents

Geometrical dimensions: Length, radius

They consist of several components (geometry, wear). The control system computes the components to a certain dimension (e.g., overall length 1, total radius). The respective overall dimension becomes effective when the compensation memory is activated.

How these values are calculated in the axes is determined by the tool type and the commands G17, G18, G19 for the plane selection.

Tool type

The tool type (drill, turning tool or milling cutter) determines which geometry data are required and how they will be computed.

Length of cutting edge

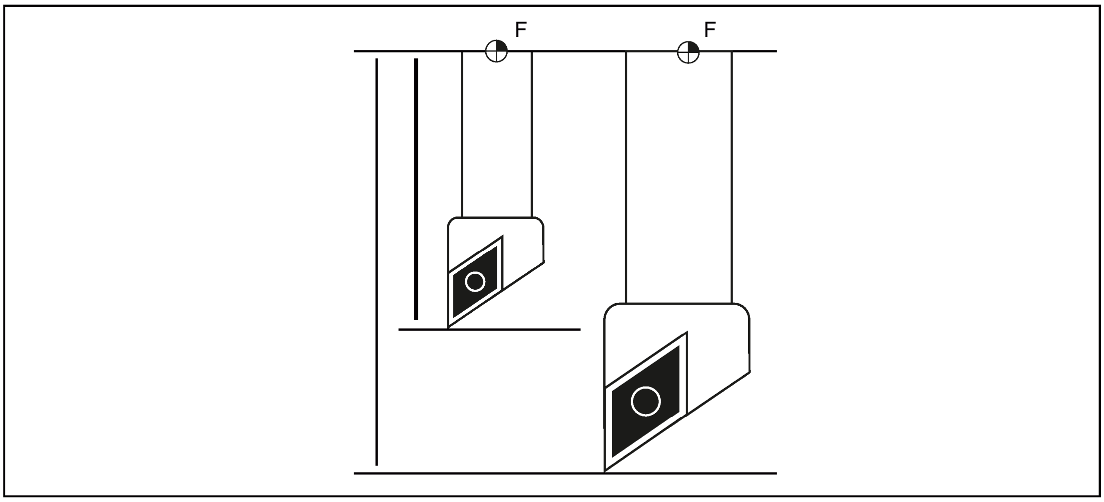
For the "turning tool" tool type, you must also enter the length of the cutting edge. The following images provide information on the necessary tool parameters.

3.3.4.2 Tool length compensation

This value compensates for the differences in length between the tools used.

The tool length is the distance between the toolholder reference point and the tool tip.

Tool length:

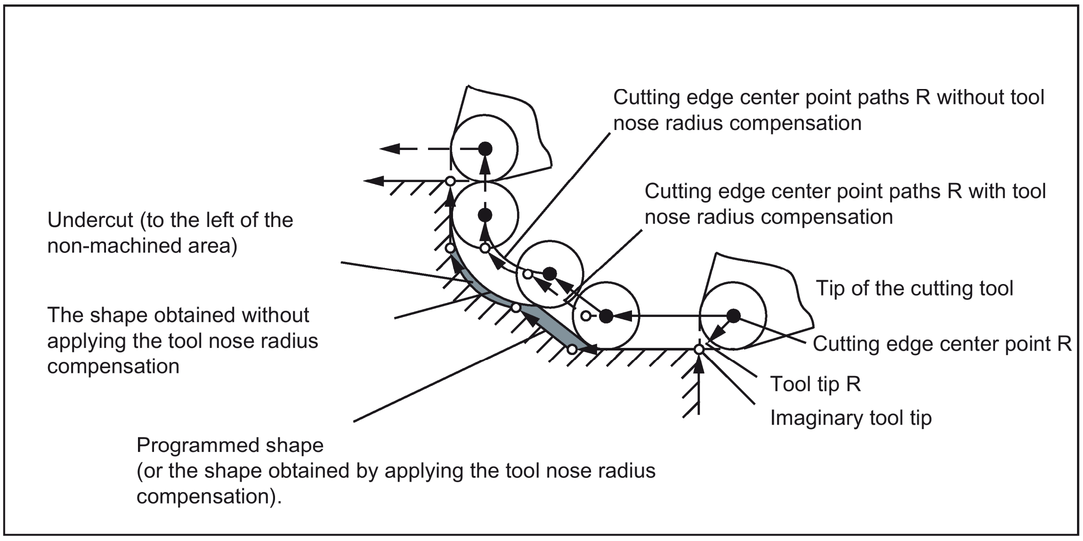


These lengths are measured and entered in the tool offset data memory together with the wear values. From this data, the control system calculates the traversing movements in the infeed direction.

3.3.4.3 Tool nose radius compensation (G40, G41/G42)

Since the tip of a cutting tool is always rounded, there are contour inaccuracies during taper turning or while processing the arcs, if the cutting edge radius is not taken into account. The diagram below illustrates how these problems arise. The tool nose radius compensation, which compensates such contour inaccuracies, is activated via G41 or G42.

Machining without tool nose radius compensation:



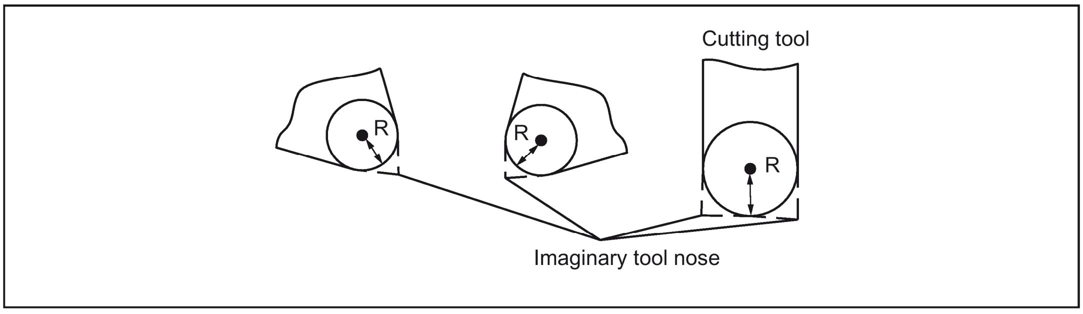
Amount of the tool nose radius compensation

The term "Amount of the tool nose radius compensation" defines the distance from the tool tip to the cutting edge center point R.

Defining the amount of the tool nose radius compensation

The amount of the tool nose radius compensation is specified via the circle radius of the tool tip without the sign.

Defining the amount of the tool nose radius compensation and an imaginary tool tip:

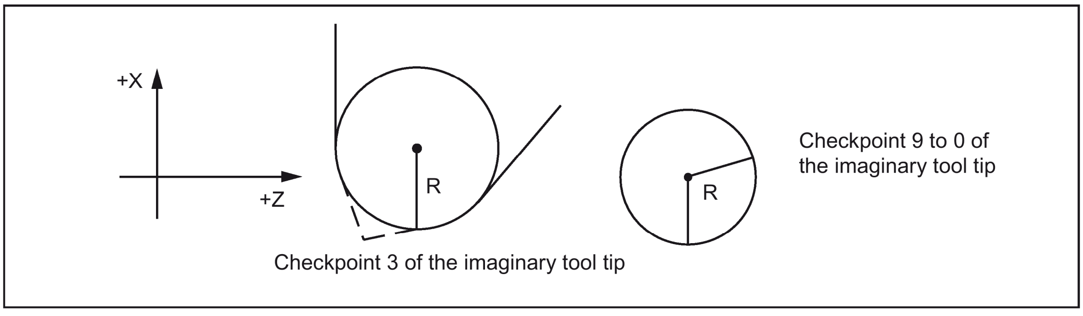


Defining an imaginary tool tip position (checkpoint)

Checkpoint memory

The position of the imaginary tool tip, observed from the center point of the tool tip R, is specified with a single-digit number from 0 to 9. This is the checkpoint. The checkpoint should be entered in the storage of the NC before saving the tool data.

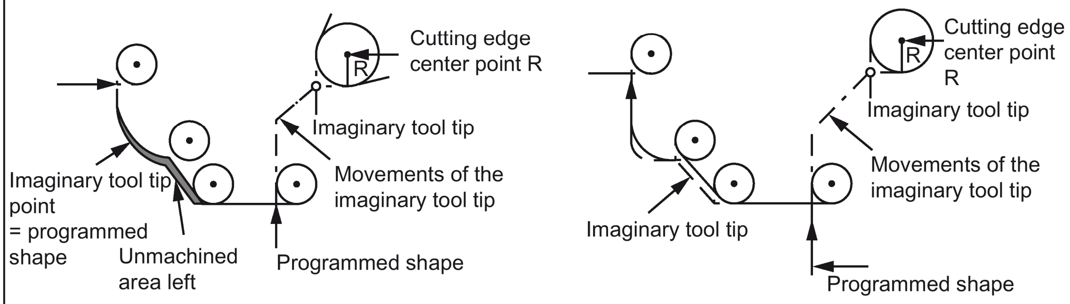
Example for defining a checkpoint:



Checkpoints and programs

When using the checkpoints 1 to 8 the imaginary length of the tool tip is to be used as reference while writing the program. The program should be written only after defining the coordinate systems.

Program and tool movements for the checkpoints 1 to 8:



(a) Program without tool nose radius compensation R

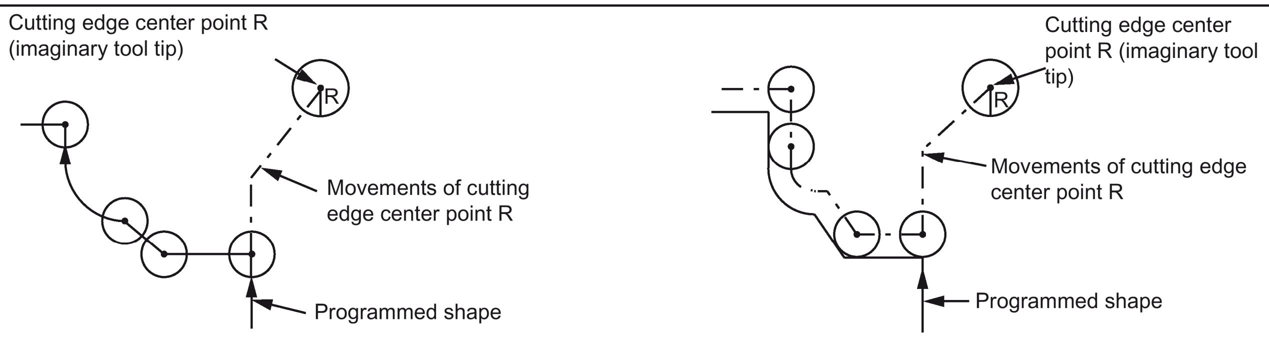
The imaginary tool tip follows the programmed shape, causing overcuts at cones and arcs.

(b) Program with tool nose radius compensation

Through the tool nose radius compensation R the path of the tool deviates from the programmed shape, so that the overcuts and the undercuts are eliminated.

When using the checkpoints 0 to 9 the cutting edge center point R is to be used as reference while writing the program. The program should be written only after defining the coordinate systems. If no tool nose radius compensation is used, the programmed shape may not differ from the processed one.

Program and tool movements for the checkpoints 0 to 9:



(a) Program without tool nose radius compensation R

The cutting edge center point R follows the programmed shape. Therefore, if the coordinate system is defined with respect to the cutting edge center point R, the programmed shape must be different from the machined one.

(b) Program with tool nose radius compensation

In the figure above, a corresponding offset is done as in case of program (b), in order to get a good surface quality without overcuts and undercuts in the finishing.

Selection/deselection of the tool nose radius compensation

- Selection of the tool offset
  The tool offset is selected through a T command.
- Activating the tool nose radius compensation

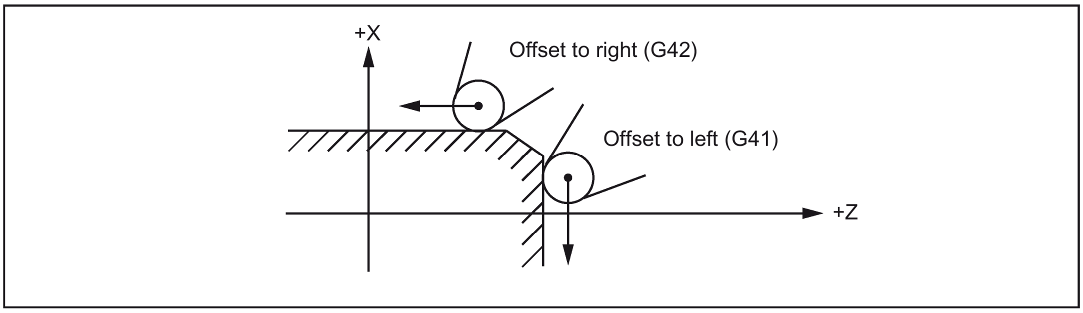
The following G functions are used for activating/deactivating the tool nose radius compensation:

G function	Function	G group
G40	Deselect the tool radius compensation	07
G41	Tool radius compensation (tool operates in machining direction to the left of the contour)	07
G42	Tool radius compensation (tool operates in machining direction to the right of the contour)	07

The commands G40 and G41/G42 are modal G functions of the G group 07. These remain active for as long as till another function of this G group is programmed. Closed position after POWER ON or NCK-RESET is G40.

The tool nose radius compensation is called either with G41 or with G42 and a T command.

Defining the tool nose radius compensation depending upon the machining direction:

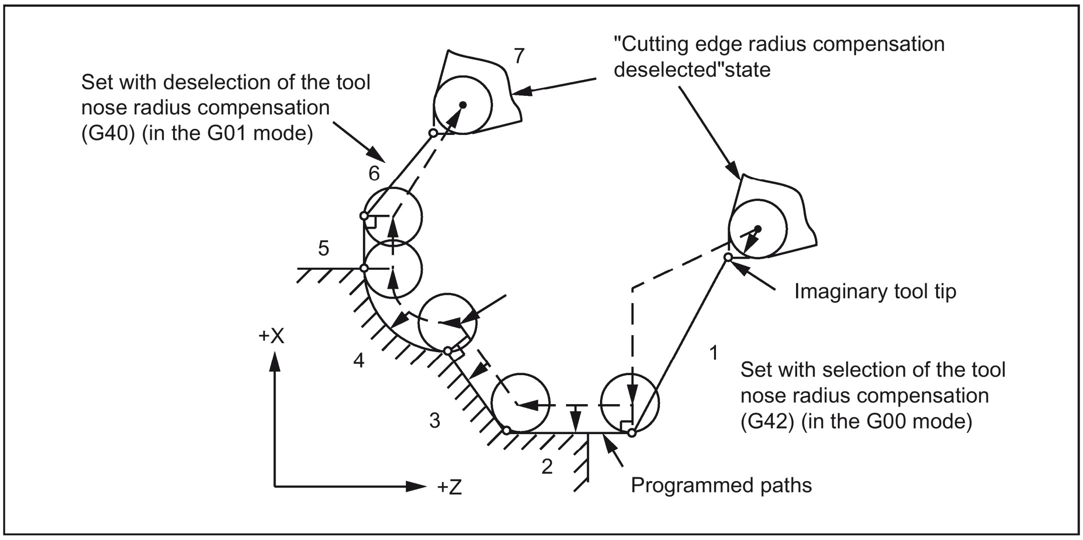


Change of the compensation direction

The compensation direction can be switched between G41 and G42 without deselecting G40. The last block with the old compensation direction ends with the normal position of the compensation vector at the end point. The new compensation direction is executed as a compensation start (default setting at starting point).

Contour of the motions in case of tool radius compensation

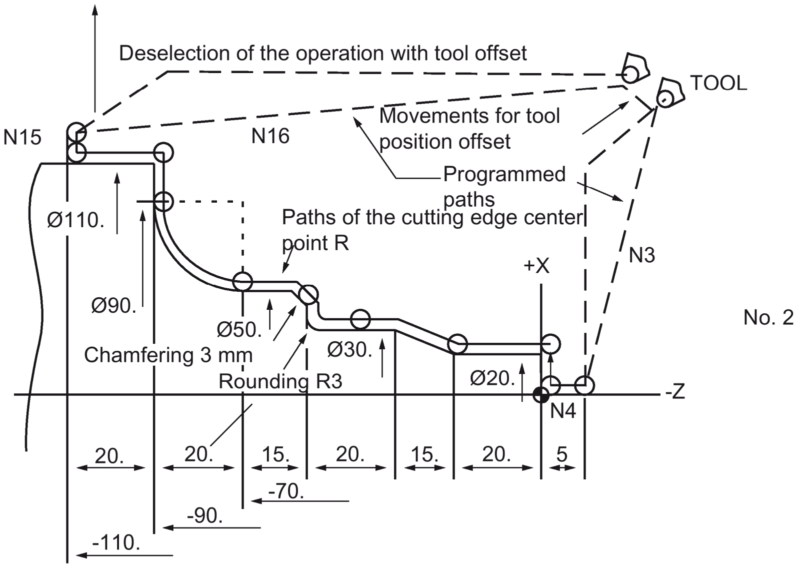
The following figure shows the execution of the tool radius compensation:



Upon selection (block 1) and deselection (block 6) of the tool nose radius compensation the compensating movements are carried out. Therefore, while selecting or deselecting the tool offset, be very careful that no collisions take place.

Programming example:

```
Programming example:
N1 G92 X140. Z20.;
N2 G00 S1700 M03 T0202;
N3 (G00) G42 X0 Z5.;          ——— Set for selecting the tool nose radius compensation R
N4 G01 Z0 F0.2;
N5 X20.;
N6 Z-20.;
N7 X30. W-15. S1100;
N8 G1 W-20. R3.;              ——— (Rounding)
N9 G1 X50. K-3. S700;         ——— (Chamfering)
N10 G01 Z-70.;
N11 G02 X90. Z-90. R20. S360; ——— Arc specification via R command
N12 G01 X110. S300;
N13 G04 U0;                   ——— Dwell time for reaching a sharp edge
N14 (G01) Z-110.;
N15 X120.;
N16 G00 X140. Z30.;
N17 G40;                      ——— Set with deselection of the tool nose radius compensation R
...
```



3.3.5 S-, T-, M- and B functions

3.3.5.1 Spindle function (S function)

The spindle speed is specified in rpm in Address S. The direction of spindle rotation is selected with M3 and M4. M3 = right direction of spindle rotation, M4 = left direction of spindle rotation. The spindle stops with M5. Details are available in the documentation of your machine manufacturer.

- S commands are modal, i.e., they remain active up to the next S command once they are programmed. The S command is maintained if the spindle is stopped with M05. If M03 or M04 is programmed thereafter without specifying an S command, then the spindle starts at the originally programmed speed.
- If the spindle speed is changed, please pay attention to which gear stage is currently set for the spindle. Details are available in the documentation of your machine manufacturer.
- The lower limit for the S command (S0 or an S command near S0) depends on the drive motor and the drive system of the spindle and is different from machine to machine. Negative values are not permitted for S! Details are available in the documentation of your machine manufacturer.

3.3.5.2 Constant cutting rate (G96, G97)

A constant cutting rate is selected and deselected with the following G functions. The commands G96 and G97 act globally and belong to the G group 02.

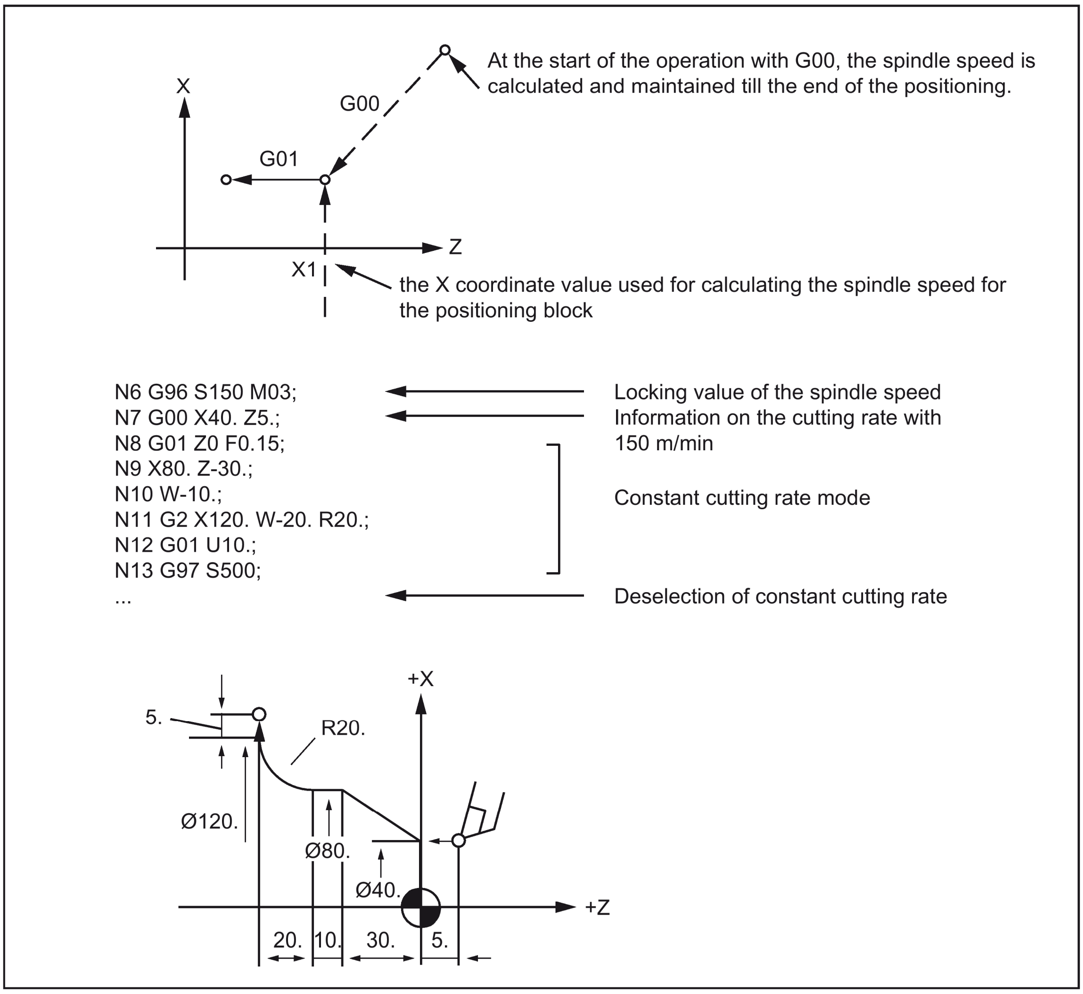
G function	Function	G group
G96	Constant cutting rate ON	02
G97	Deselection of constant cutting rate	02

Constant cutting rate ON (G96)

With "G96 S..." the spindle speed – depending on the respective workpiece diameter – is modified such that the cutting rate S in m/min or ft/min remains constant at the tool edge.

After activating with G96, the value of the X-axis is used as the diameter for monitoring the current cutting rate. If the position of the X-axis is changed, the spindle speed also changes in such a way that the programmed cutting rate is maintained.

Constant cutting rate:



Deselection of the constant cutting rate (G97)

According to G97, the control system interprets an S-word as spindle revolution in RPM. In case no new spindle revolution is specified, the last speed implemented by G96 is maintained.

Selection of the spindle revolution gear stage

In case of machines, in which the gear stage can be switched with an M command, the M command is to be written for selecting the corresponding gear stage before specifying G96. Details can be found in the documentation of your machine manufacturer.

Programming example

```
⋮
N8 Mxx;              ← M function for selecting the gear stage
N9 G96 S100 M03;       (Example: Gear stage no. 4)
⋮
```

3.3.5.3 Tool change with T functions (T function)

There is a direct tool change when the T word is programmed.

The effect of the T function is defined via the machine data. See machine manufacturer's configuration.

3.3.5.4 Additional function (M function)

The M functions initiate switching operations, such as "Coolant ON/OFF" and other functions on the machine. Various M functions have already been assigned a fixed functionality by the CNC manufacturer (see the following section).

Programming

M... Possible values: 0 to 9999 9999 (maximum INT value), integer

All free M function numbers can be assigned by the machine manufacturer, e.g. for switching functions to control the clamping devices or for switching on/off of further machine functions. See data of the machine manufacturer.

The NC-specific M functions are described below.

M functions to stop operations (M00, M01, M02, M30)

A program stop is triggered with this M function and the machining is interrupted or ended. Whether the spindle is also stopped depends on the specification of the machine manufacturer. Details are available in the documentation of your machine manufacturer.

M00 (program stop)

The machining is stopped in the NC block with M00. One can now, e.g., remove chips, re-measure, etc. A signal is output to the PLC. The program can be continued with the following key:

M01 (optional stop)

M01 can be set via the following methods:

- HMI/dialog box "Program control" or the
- VDI interface

The program processing of the NC is maintained with M01 only if the corresponding signal of the VDI interface is set or "Program control" was selected in the HMI/dialog box.

M30 or M02 (end of program)

A program is ended with M30 or M02.

Note

A signal is output to the PLC with M00, M01, M02 or M30.

Note

Data on whether spindle is stopped with the commands M00, M01, M02 or M30 or the coolant supply is interrupted is available in the documentation of your machine manufacturer.

3.3.5.5 M functions of spindle control (M19, M29)

M function	Function
M19	Positioning the spindle
M29	Changeover of spindle to the axis/open-loop control mode

The spindle is traversed to the spindle position defined in the setting data 43240 $SA_M19_SPOS[spindle number] with M19. The positioning mode is stored in $SA_M19_SPOS.

The M function number for the changeover of the spindle mode (M29) can also be set over a machine data variable. MD20095 $MC_EXTERN_RIGID_TAPPING_N_NR is used to pre-set the M function number. Only the M function numbers that are not used as standard M functions can be assigned. For example, M0, M5, M30, M98, M99 etc are not allowed.

In ISO mode, the spindle is switched to the axis mode with M29.

3.3.5.6 M functions for subroutine calls (M98, M99)

M function	Function
M98	Subprogram call
M99	Subprogram end

3.3.5.7 Macro call via M function

Via M numbers, one can call a subroutine (macro) similar to G65.

The configuration of a maximum of 10 M functions replacements is undertaken via machine data 10814 $MN_EXTERN_M_NO_MAC_CYCLE and machine data 10815 $MN_EXTERN_M_NO_MAC_CYCLE_NAME.

Programming takes place identical to G65. Repetitions can be programmed with the L address.

Restrictions

Only one M function replacement (or only one subroutine call) can be executed per part program line. Conflicts with other subroutine calls are signaled by alarm 12722. There is no further M function replacement in the replaced subroutine.

Otherwise, the same restrictions are valid as in G65.

Conflicts with pre-defined and other defined M numbers are rejected with an alarm.

Configuration example

Call of subroutine M101_MAKRO via the M101 M function:

$MN_EXTERN_M_NO_MAC_CYCLE[0] = 101

$MN_EXTERN_M_NO_MAC_CYCLE_NAME[0] = "M101_MAKRO"

Call of subroutine M6_MAKRO via the M6 M function:

$MN_EXTERN_M_NO_MAC_CYCLE[1] = 6

$MN_EXTERN_M_NO_MAC_CYCLE_NAME[1] = "M6_MAKRO"

Programming example for tool change with M function:

```
PROC MAIN
...
N10                 M6 X10 V20                          ;Call of M6_MAKRO program
...
N90                 M30
PROC M6_MAKRO
...
N0010               R10 = R10 + 11.11
N0020               IF $C_X_PROG == 1 GOTOF N40         ;($C_X_PROG)
N0030               SETAL(61000)                        ;programmed variable not
                                                        ;transferred correctly
N0040               IF $C_V == 20 GTOF N60              ;($C_V)
N0050               SETAL(61001)
N0060               M17
```

3.3.5.8 M functions (M08, M09)

General M functions

The non-specific M functions are defined by the machine manufacturer. A representative example of the use of general M functions is available under. Details are available in the documentation of your machine manufacturer. If an M command is programmed with an axis motion in the same block, whether the M function is to be executed at the start or end of the block on reaching the axis position depends on the machine data setting of the machine manufacturer. Details are available in the documentation of your machine manufacturer.

M function	Function	Remarks
M08	Coolant ON	These M functions are defined by the machine manufacturer.
M09	Coolant OFF	

Specification of several M functions in one block

A maximum of five M functions can be programmed in on block. Possible combinations of M functions and possible restrictions are specified in the documentation of your machine manufacturer.

Additional auxiliary functions (B functions)

If B is not used as axis identifier, B can be used as extended auxiliary function. B functions are output to the PLC as auxiliary functions (H functions with the address extension H1=).

Example: B1234 is output as H1=1234.

3.4 Additional functions

3.4.1 Program-support functions

3.4.1.1 Fixed cycles (G90, G92, G94)

Fixed cycles simplify it for the programmer to create new programs. Frequently occurring machining steps can be executed with a G function; without fixed cycles several NC blocks must be programmed. In this way, with fixed cycles, the machining program can be shortened and the memory space can be saved.

In the ISO Dialect a shell cycle is called, which uses the functionality of the standard Siemens cycle. Thus the addresses programmed in the NC block are transferred to the shell cycle via the system variable. The shell cycle customizes these data and calls a standard Siemens cycle.

A fixed cycle could be canceled only with G80 or a G code of G code Group 1 before the program can be continued with a blockwise cycle.

Longitudinal turning cycle

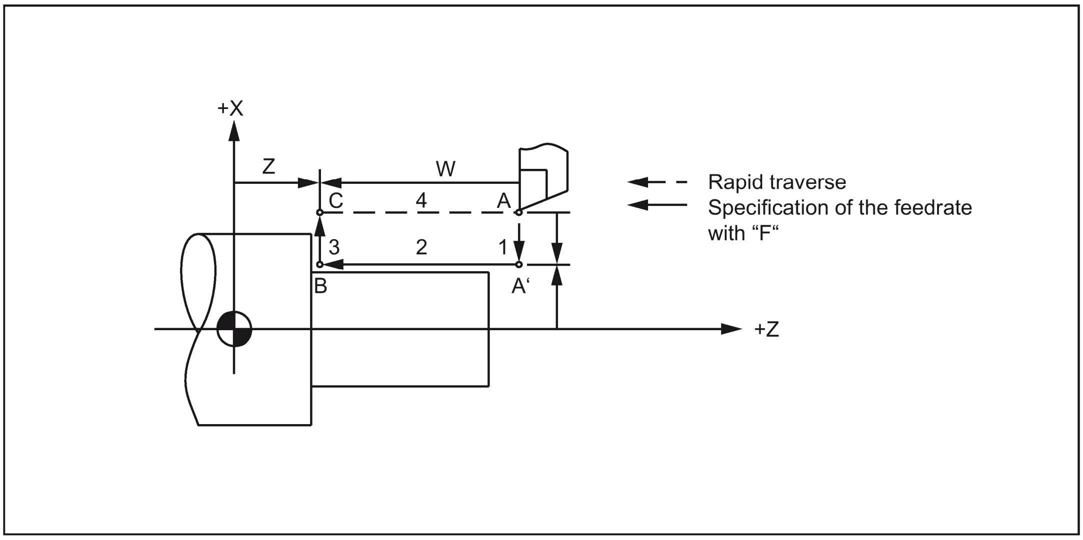
Format

G.. X... Z... F... ;

G code system A	G code system B	G code system C
G90	G77	G20

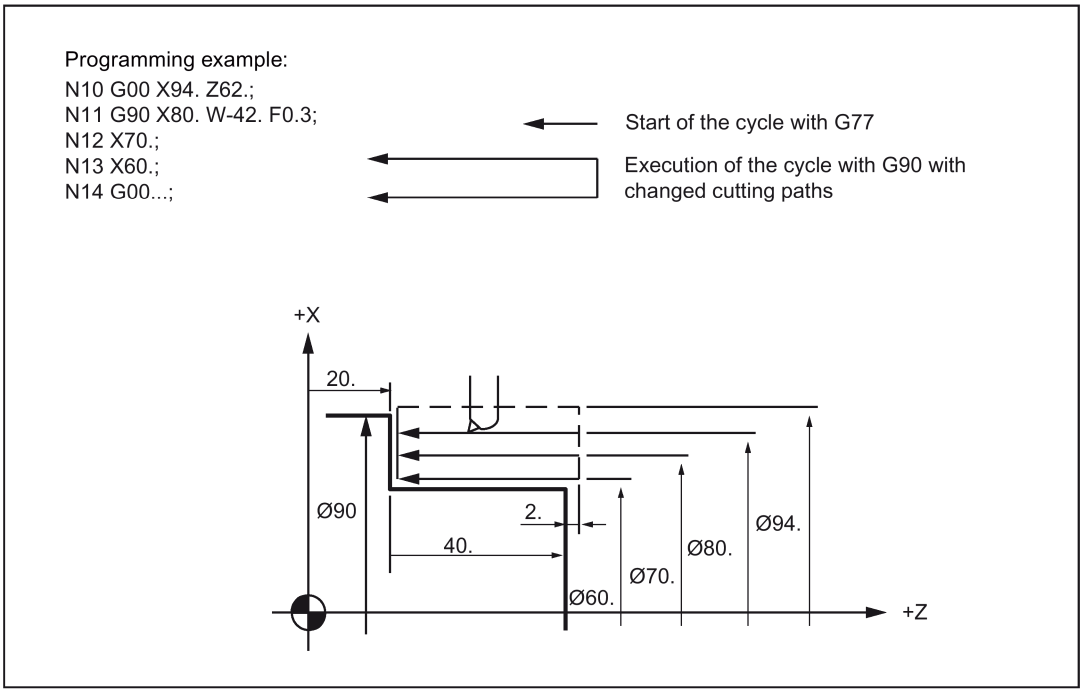
With the commands "G... X(U)... Z(W)... F... ;", a longitudinal turning cycle is executed as per the sequence 1-4.

Longitudinal turning cycle:



Since G90 (G77, G20) is a modal G function, the machining is executed within the cycle by specifying only the infeed motion in the direction of X axis in the subsequent blocks.

Longitudinal turning cycle (G code system A):



Straight cutting cycle

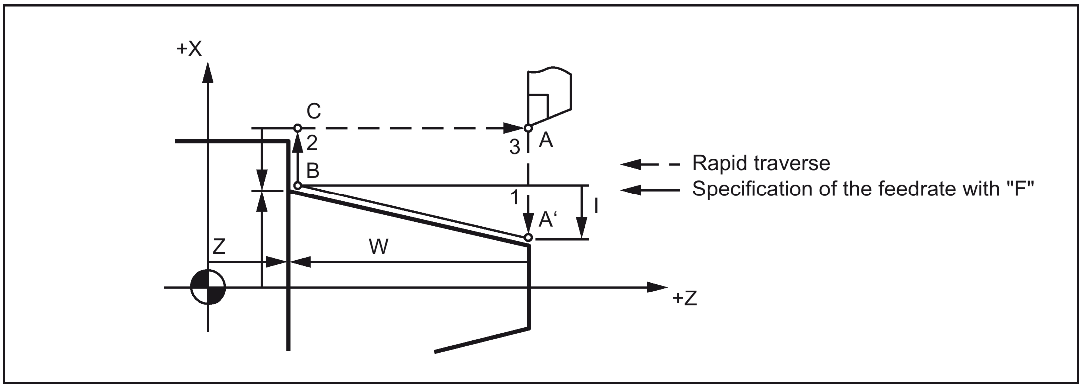
Format

G... X... Z... R... F... ;

G code system A	G code system B	G code system C
G90	G77	G20

With the commands "G... X(U)... Z(W)... R... F... ;", a straight cutting cycle is executed as per the sequence 1-4 given in the figure below.

Straight cutting cycle:

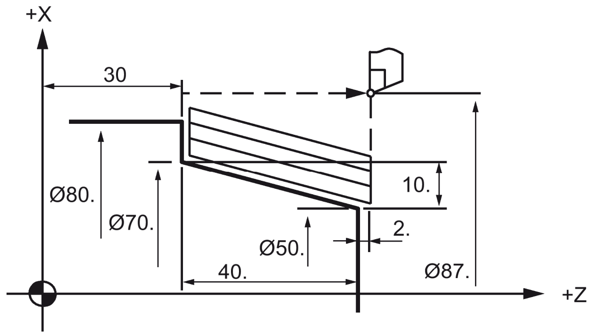


The sign before the address character R depends on point A' of the viewing direction from point B.

Straight cutting cycle (G code system A):

Programming example:

```
N20 G00 X87. Z72.;
N21 G90 X85. W-42. R-10.5 F0.25;
N22 X80.;
N23 X75.;
N24 X70.;
N25 G00 ...;
```



- When the cycle with G90 (G77, G20) is executed with activated single-block mode, the cycle is not terminated in the middle, but stops after the end of the cycle, which comprises of the sequence 1-4.
- The functions S, T and M, which are used as cutting conditions for the execution of G90 (G77, G20), are to be specified in the blocks before the block G90 (G77, G20). When these functions are specified in a block with the travel of axes, then the functions act only when the block is specified in the range of the operation with G90 (G77, G20).

```
G77   X... Z... R... F...;            ┐
      X...;                           │  G77 valid range
      X...;                           │
      X... T0505 M05; ← Error         ┘
G00   X... Z...;

G77   X... Z... R... F...;            ┐
      X...;                           │  G77 valid range
      X...;                           │
G00   X... T0505 M05; ← Correct       ┘
      X... Z...;
```

The operation with G90 (G77, G20) then remains active till the block where a G function of the group 01 is specified.

Thread cutting cycle

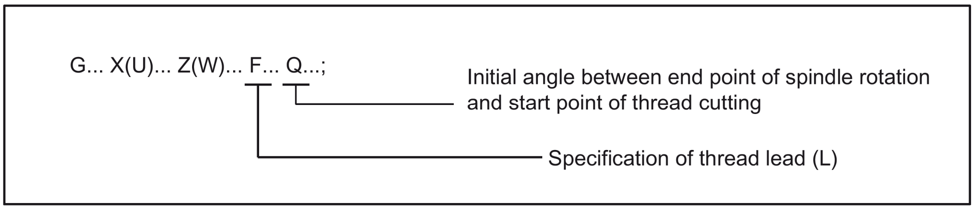
There are three kinds of thread cutting operations: cycles for cutting cylindrical threads, cycles for cutting taper threads, and transverse-taper turning cycles.

Format

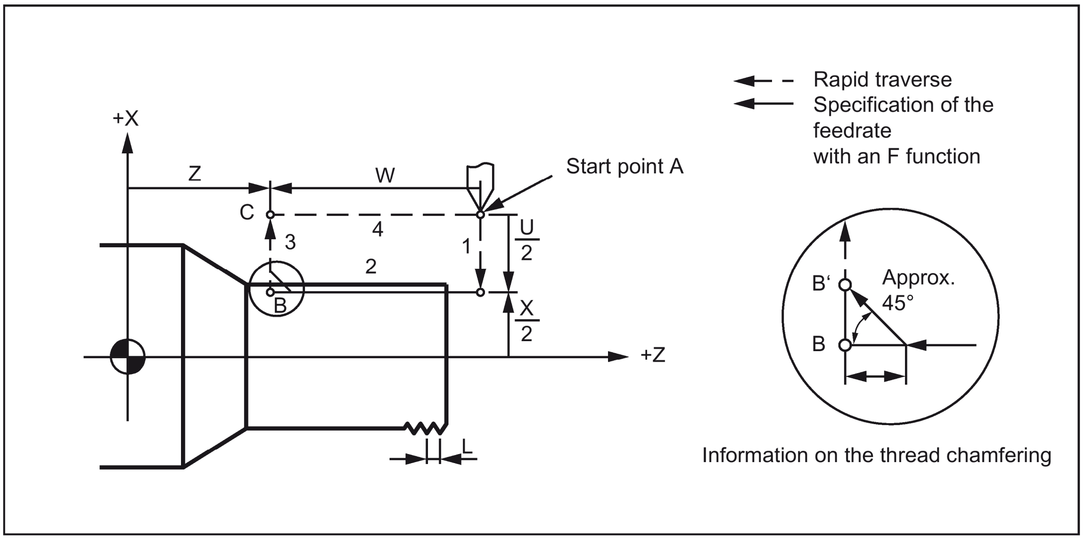
G... X... Z... F... Q... ;

G code system A	G code system B	G code system C
G92	G78	G21

Cycle for cutting cylindrical threads



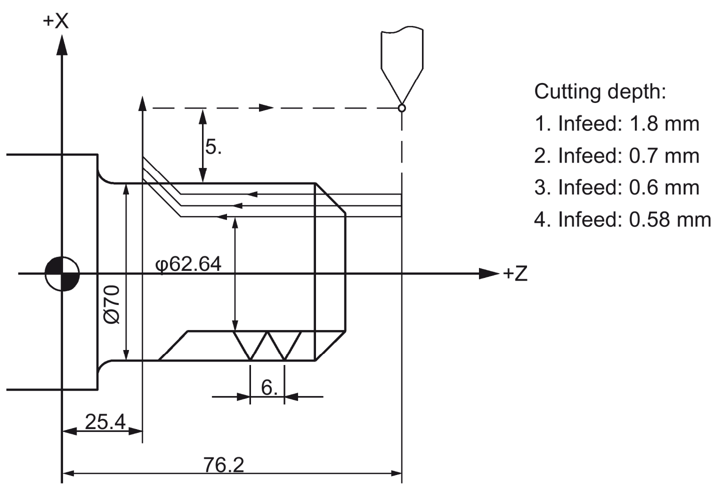
With the commands given above, the cycle for cutting cylindrical threads, sequence 1 - 4, are executed as shown in the figure below.



Since G92 (G78, G21) is a modal G function, the thread cutting cycle is executed within the cycle by specifying only the cutting depth in the direction of the X axis in the subsequent blocks. In these blocks G92 (G78, G21) need not be specified again.

Cycle for cutting a cylindrical thread (G code system B):

```
Programming example:
N30 G00 X80. Z76.2 Mxx;          ← Mxx; thread chamfering ON
N31 G78 X66.4 Z25.4 F6.;
N32 X65.;                          Thread cutting cycle with four infeeds
N33 X62.8;
N34 X62.64;                        Myy; thread chamfering OFF
N35 G00 X100. Z100. Myy;         ←
```



- When the cycle with G92 (G78, G21) is executed with activated single-block mode, the cycle does not wait on the halfway, but stops after the end of the cycle, which comprises of the sequence 1-4.
- Thread chamfering is possible within this thread cutting cycle. Thread chamfering is initiated by a machine signal. The size of the chamfer for the thread g can be specified in steps of 0.1*L in USER DATA, _ZSFI[26]. "L" is thus the specified thread lead.

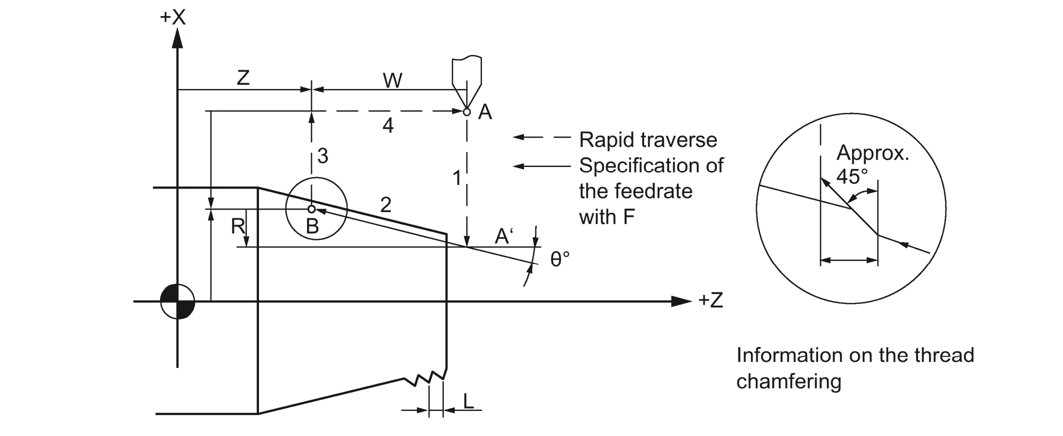
Cycle for cutting taper threads

Format

G... X... Z... R... F... ;

G code system A	G code system B	G code system C
G92	G78	G21

With the commands "G... X(U)... Z(W)... R... F... ;", a cycle for cutting taper threads is executed as per the sequence 1-4 given in the figure below.

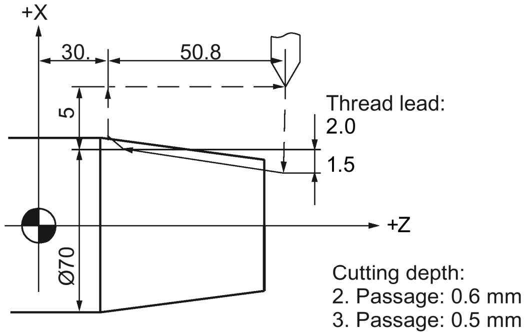


The sign before the address character R depends on point A' of the viewing direction from point B. Since G92 (G78, G21) is a modal G function, the thread cutting cycle is executed within the cycle by specifying only the cutting depth in the direction of the X axis in the subsequent blocks. In these blocks G92 (G78, G21) need not be specified again.

Cycle for cutting a taper thread (G code system A):

Programming example:

```
N50 G00 X80. Z80.8 Mxx;
N51 G92 X70. W-50.8 R-1.5 F2.;
N52 X68.8;
N53 X67.8;
N54 G00 X100. Z100. Myy;
```



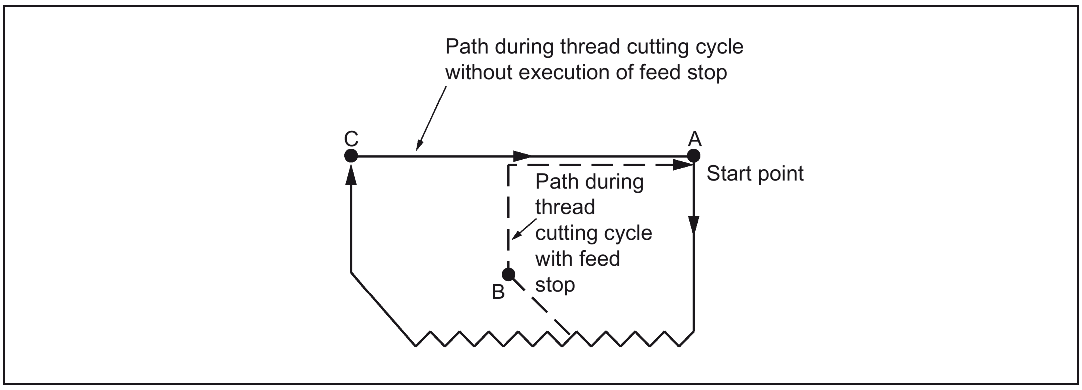
When the cycle with G92 (G78, G21) is executed with activated single-block mode, the cycle does not wait on the halfway, but stops after the end of the cycle, which comprises of the sequence 1-4.

The functions S, T and M, which are used as cutting conditions for the execution of G92 (G78, G21), are to be specified in the blocks before the block G92 (G78, G21). When these functions are specified in a block with the travel of axes, then the functions act only when the block is specified in the range of the operation with G92 (G78, G21).

If at the moment, when the cutting tool is at the starting point A or at the chamfering completion point B, the following key is pressed, then the suspended cycle is executed once again from the beginning.

If the option "thread cutting feedrate halt" is not selected, then the thread cutting cycle is continued when the following key is pressed during the execution of the thread cutting cycle. In this case the machining is halted till the tool is retracted again after the completion of the thread cutting cycle.

Feedrate halt during the execution of the thread-cutting cycle:



An alarm is output if the size of the chamfer is "0" during the use of G92 (G78, G21) in the cycle.

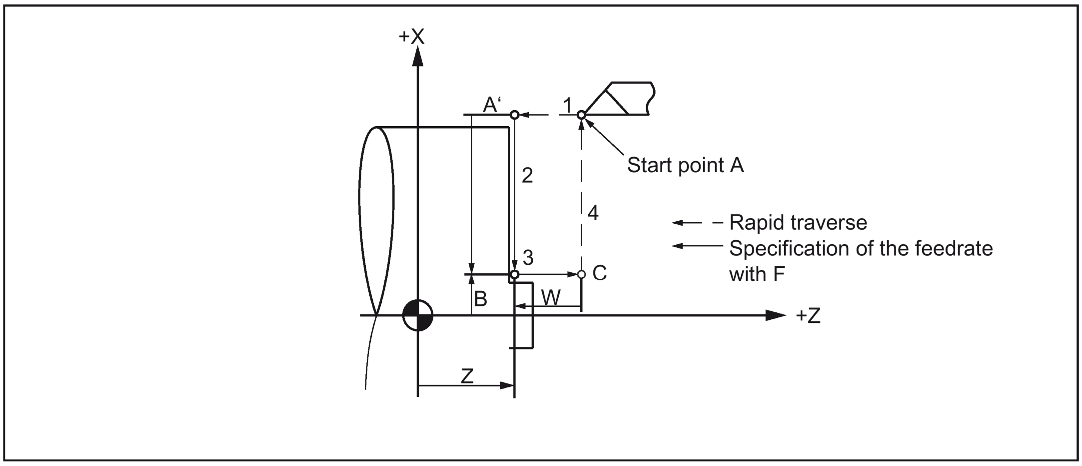
Radial cutting cycle

Format

G... X... Z... F... ;

G code system A	G code system B	G code system C
G94	G79	G24

With the commands "G... X(U)... Z(W)... F... ;", a straight facing cycle is executed as per the sequence 1-4 given in the figure below.



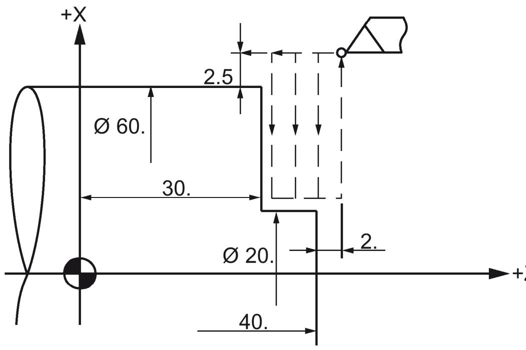
Since G94 (G79, G24) is a modal G function, the thread cutting cycle is executed within the cycle by specifying only the cutting depth in the direction of the Z axis in the subsequent blocks. In these blocks G94 (G79, G24) need not be specified again.

Straight facing cycle (G code system B):

Programming example:

```
N60 G00 X65. Z42.;
N61 G79 X20. Z38. F0.35;
N62 Z34.;
N63 Z30.;
N64 G00;
```

Machining in 3 cycles with G79



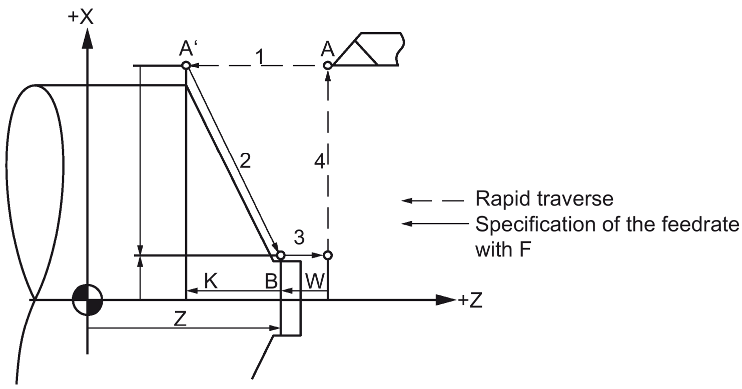
Transverse-taper turning cycle

Format

G... X... Z... R... F... ;

G code system A	G code system B	G code system C
G92	G78	G21

With the commands "G... X(U)... Z(W)... R... F... ;", a transverse-taper turning cycle is executed as per the sequence 1-4 given in the figure below.



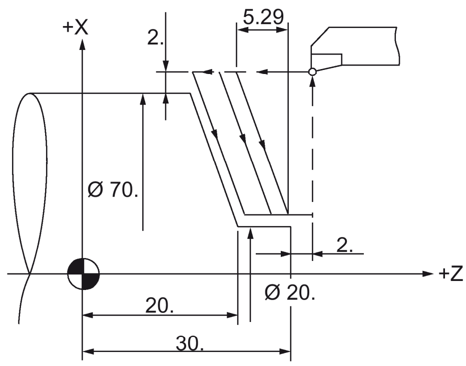
The sign before the address character R depends on point A' of the viewing direction from point B.

Transverse-taper turning cycle (G code system B):

Programming example:

```
N70N G00 X74. Z32.;
N71 G79 X20. Z30. R-5.29 F0.3;
N72 Z25.;
N73 Z20.;
N74 G00;
```

Machining in 3 cycles with G79



The functions S, T and M, which are used as cutting conditions for the execution of G94 (G79, G24), are to be specified in the blocks before the block G94 (G79, G24). When these functions are specified in a block with the travel of axes, then the functions act only when the block is specified in the range of the operation with G94 (G79, G24).

When the cycle with G94 (G79, G24) is executed with activated single-block mode, the cycle is not terminated in the middle, but stops after the end of the cycle, which comprises of the sequence 1-4.

3.4.1.2 Multiple repetitive cycles (G70 to G76)

Multiple repetitive cycles simplify the creation of new programs for the programmers. Frequently occurring machining steps can be executed with a G function; without multiple repetitive cycles several NC blocks must be programmed. In this way, the machining programs can be shortened and memory space can be saved by using multiple repetitive cycles.

In the ISO Dialect a shell cycle is called, which uses the functionality of the standard Siemens cycle. Thus the addresses programmed in the NC block are transferred to the shell cycle via the system variable. The shell cycle customizes these data and calls a standard Siemens cycle.

There are seven multiple repetitive cycles (G70 to G76) in the G code system A and B (see following table). Please note that all these G functions are not modal G functions.

G code	Description
G70	Finishing cycle
G71	Stock removal cycle, longitudinal axis
G72	Stock removal cycle, transverse axis
G73	Closed cutting cycle
G74	Multiple repetitive grooving cycles in the longitudinal axis
G75	Multiple repetitive grooving cycles in the transverse axis
G76	Multiple thread cutting cycle

These cycles are present in the G code system C also. However, other G functions are used. See the table below.

G code	Description
G72	Finishing cycle
G73	Stock removal cycle, longitudinal axis
G74	Stock removal cycle, transverse axis
G75	Contour repetition
G76	Multiple repetitive grooving cycles in the longitudinal axis
G77	Multiple repetitive grooving cycles in the transverse axis
G78	Multiple thread cutting cycle

Note

In the cycle descriptions given above, the G code system A and B are assumed.

Note

Consider the following USER DATA setting requirements when calling a cycle with G71/G72:

- _ZSFI[36] = 0: G71/G72 works with additional contour cut.
- _ZSFI[36] = 1 (default): G71/G72 works without additional contour cut.
- _ZSFI[35] = 0 (default): G71/G72 works with G41/G42 for roughing and finishing and traverses a two-cut distance in the cycle.
- _ZSFI[35] = 1: G71/G72 works with G40 for roughing and traverses a one-cut distance in the cycle.

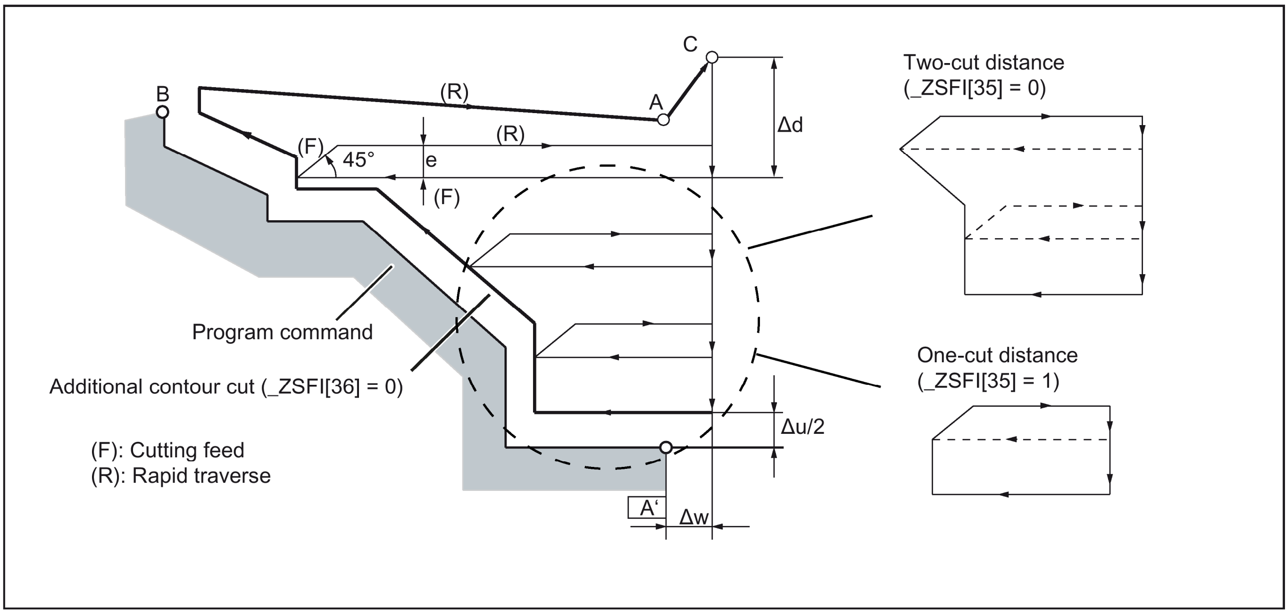
Stock removal cycle, longitudinal axis (G71)

By using the fixed cycles the number of steps in the programming can be reduced considerably owing to the fact that the roughing and the finishing cycles can be determined simply by determining the finishing cut form and the like. There are two different types of stock removal cycle.

Type I

The specified area is machined with finishing allowance through Δd (infeed depth during stock removal). Always when the contours A are written to A' to B by an NC Program, u/2 and Δw continue to be present.

Cutting path of a stock removal cycle, longitudinal axis:



Format

G71 U... R... ;

U: Infeed depth during stock removal (Δd), radius programming

This value is modal and remains effective till another value is programmed. The value can also be entered via USER DATA, _ZSFI[30], but this value is overwritten by the value of the program command.

R: (e), Retraction amount

This value is modal and remains effective till another value is programmed. The value can also be entered via USER DATA, _ZSFI[31], but this value is overwritten by the value of the program command.

G71 P... Q... U... W... F... S... T...

P: Starting block for determining the contour

Q: Last block for determining the contour

U: Finishing allowance in X direction (Δu) (diameter -/radius programming)

W: Finishing allowance in Z direction (Δw)

F: Machining feedrate

S: Spindle speed

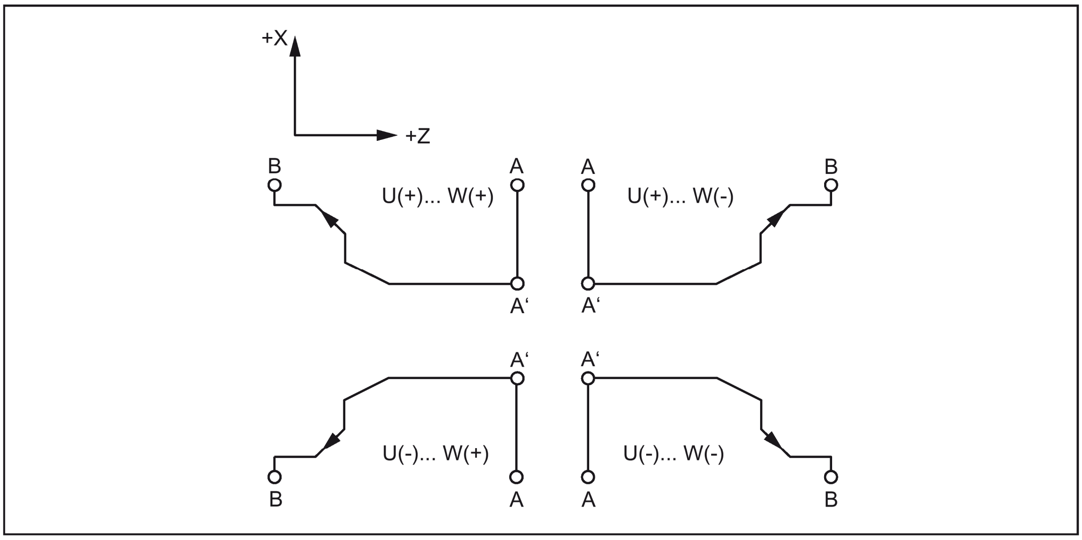
T: Select the tool

The F-, S- and T-functions printed within an NC program block and specified through the address characters P and Q are ignored. Only the F-, S- and the T-functions specified in the block with G71 are effective.

Note

Stock removal cycle, longitudinal axis

- Δd as well as Δu are specified with the address character U. When the address characters P and Q are specified, Δ"u" is the case.
- There are a total of four different cutting sectors. As shown in the figure below, Δ"u" and Δ "w" may have different signs:



Note

Stock removal cycle, longitudinal axis

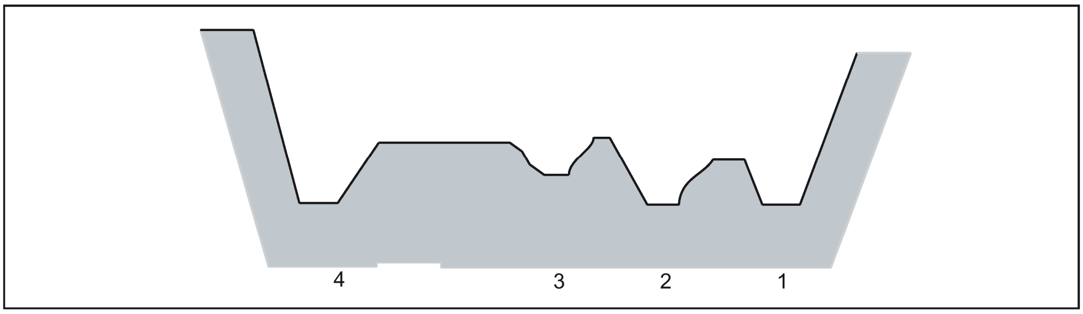
- In the block specified through the address P the contour is defined between the points A and A' (G00 or G01). No traversing command can be specified in this block in the Z axis.

  The contour defined between the points A' and B must either be a constantly rising or a constantly falling pattern on the X axis as well as on the Z axis.
- Within the range of the NC blocks specified with the address characters P and Q no subroutines can be called.

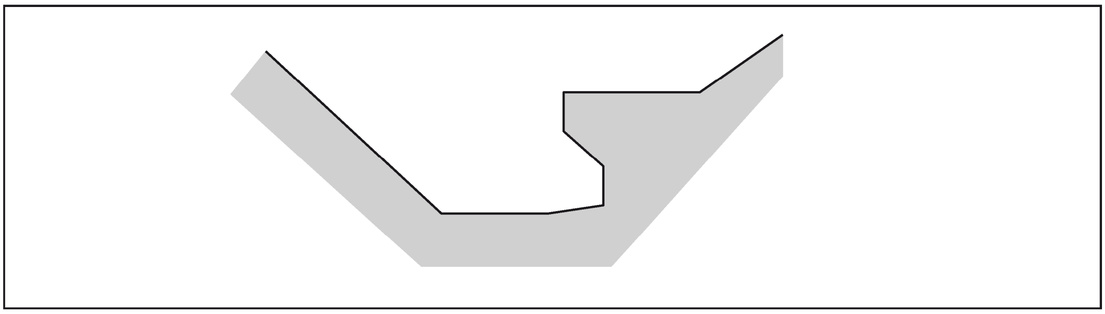
Type II

Contrary to Type I, a constant rise or a constant fall need not necessarily be specified for Type II i.e. pockets are also possible.

Pockets in case of a stock removal cycle (Type II):



Here, the profile of the Z axis must rise or fall uniformly. For example, the following profile cannot be machined:



Differentiation between Type I and Type II

Type I: Only an axis is specified in the first block in the contour description.

Type II: Two axes are specified in the first block of the contour description.

If the first block does not contain any movement in the Z axis and actually Type II should have been used, then W0 must be specified.

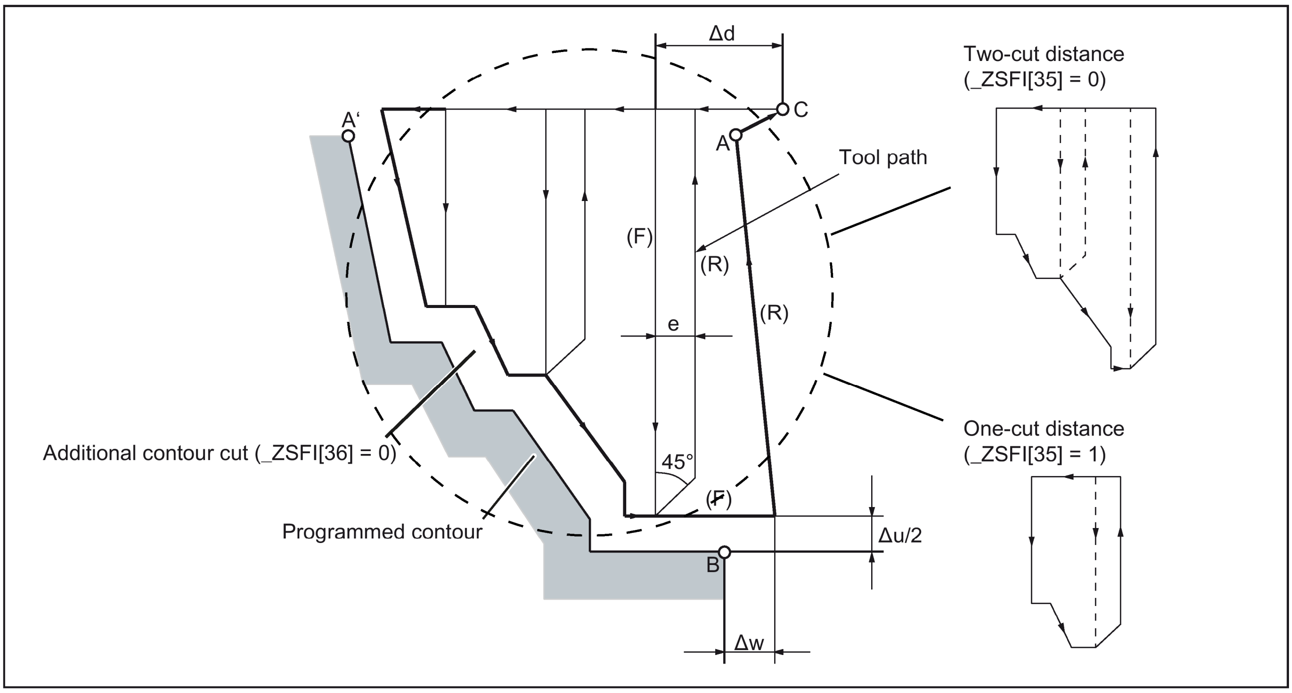
Example

Type I	Type II
G71 U10.0 R4.0 ; G71 P50 Q100 .... ; N50 X(U)... ; :: :: N100.............. ;	G71 U10.0 R4.0 ; G71 P50 Q100 .... ; N50 X(U)... Z(W)... ; :: :: N100........... ;

Stock removal cycle, transverse axis (G72)

With the command G72 a stock removal cycle can be programmed with finishing allowances on the face. As compared to the cycle called with G71, in which the machining is done through a movement parallel to the Z axis, in case of cycle G72 the machining is executed through movements parallel to the X axis. The cycle called with G72 thus executes the same machining as the one called with G71, but only in the other direction.

Cutting path of a stock removal cycle, transverse axis:



Format

G72 W... R... ;

The significance of the addresses W (Δd) and R (e) is the same as that of U and R.

G72 P... Q... U... W... F... S... T... ;

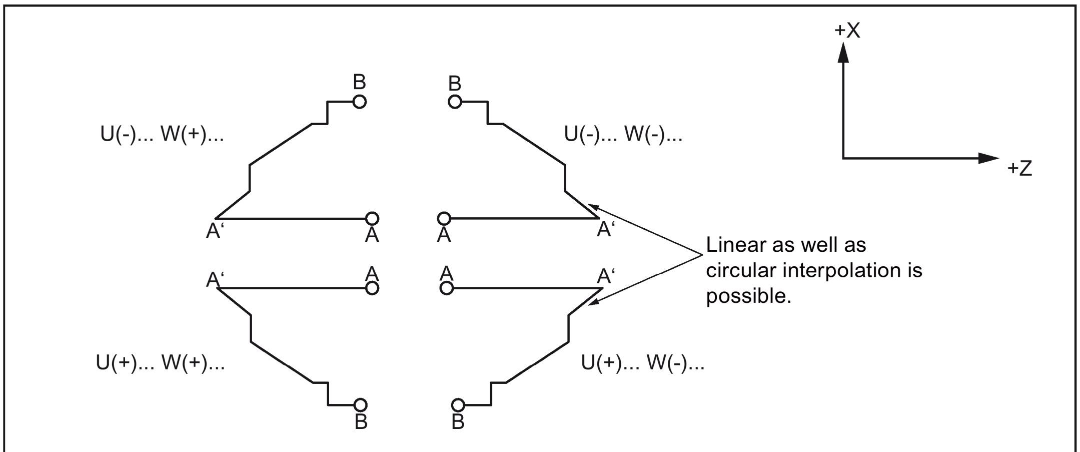
The addresses P, Q, U (Δu), W (Δw), F, S and T have the same significance as in the cycle G71.

Note

Stock removal cycle transverse axis

- The values Δ"i" and Δ"k" or Δ"u" and Δ"w" are defined with the addresses "U" or "W" respectively. Their significance, however, is defined by the address characters P and Q in the block with G73. The address characters U and W refer to Δ"i" or Δ"k", when P and Q are not specified in the same block. The address characters U and W refer to Δ"u" and Δ"w", when P and Q are not specified in the same block.
- There are a total of four different cutting sectors. As shown in the figure below, Δ"u" and Δ"w" may have different signs.

Signs of the numbers with U and W in the stock removal during face turning:



Note

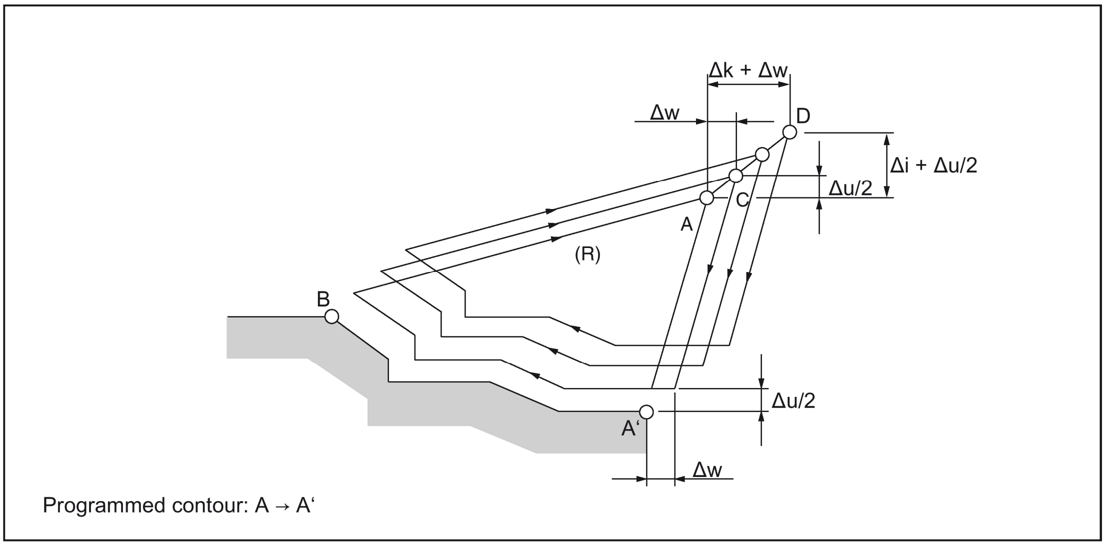
Stock removal cycle transverse axis

- The contour between the points A and A' is defined through the block specified with the address character P (G00 or G01). No traversing command can be specified in this block in the X axis. The contour defined between the points A' and B must either be a constantly rising or a constantly falling pattern on the X axis as well as on the Z axis.
- The machining is executed with the cycle with the command G73 and the specification of P and Q. The four cutting sectors will be discussed in more detail below. Please pay special attention to the signs of Δu, Δw, Δk and Δi. As soon as the execution cycle has ended, the tool returns to the point A.

Closed cutting cycle (G73)

The closed cutting cycle G73 is more effective, when a workpiece is machined, which has a form similar to that in the finishing cut i.e. cast-iron or forged workpieces.

Cutting path in the closed cutting cycle:



Format

G73 U... W... R... ;

U: Distance (Δi) from the starting point to the current tool position in the direction of the X axis (in radius programming).

This value is modal and remains effective till another value is programmed. The value can also be entered via USER DATA, _ZSFI[32], but this value is overwritten by the value of the program command.

W: Distance (Δk) from the starting point to the current tool position in the direction of the Z axis.

This value is modal and remains effective till another value is programmed. The value can also be entered via USER DATA, _ZSFI[33], but this value is overwritten by the value of the program command.

R: Number of cuts parallel to the contour (d).

This value is modal and remains effective till another value is programmed. The value can also be entered via USER DATA, _ZSFI[34], but this value is overwritten by the value of the program command.

G73 P... Q... U... W... F... S... T... ;

P: Starting block for determining the contour

Q: Last block for determining the contour

U: Finishing allowance in X direction of the X axis (Δu) (diameter -/radius programming)

W: Finishing allowance in the direction of Z axis (Δw)

F: Machining feedrate

S: Spindle speed

T: Select the tool

The F-, S- and T-functions printed within an NC program block and specified through the address characters P and Q are ignored. Only the F-, S- and the T-functions specified in the block with G73 are effective.

Finishing cycle (G70)

While the roughing is executed with G71, G72 or G73, the finishing is done with the following command.

Format

G70 P... Q... ;

P: Starting block for determining the contour

Q: Last block for determining the contour

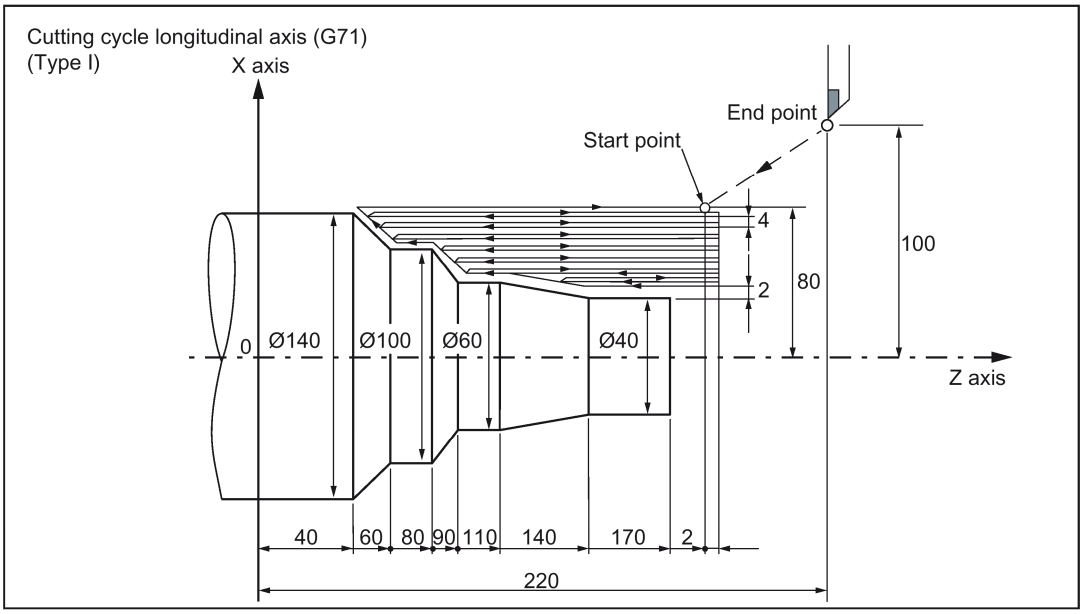
Note

Finishing cycle

- The functions specified between the blocks and defined with the address characters P and Q are effective in the cycle with G70, while the F-, S- and T-functions specified in the block with G71, G72 and G73 are not effective.
- The tool returns to the starting point and the next block is read, as soon as the execution cycle is completed with G70.
- Within the blocks defined with the address characters P and Q it is possible to call subroutines.

Examples

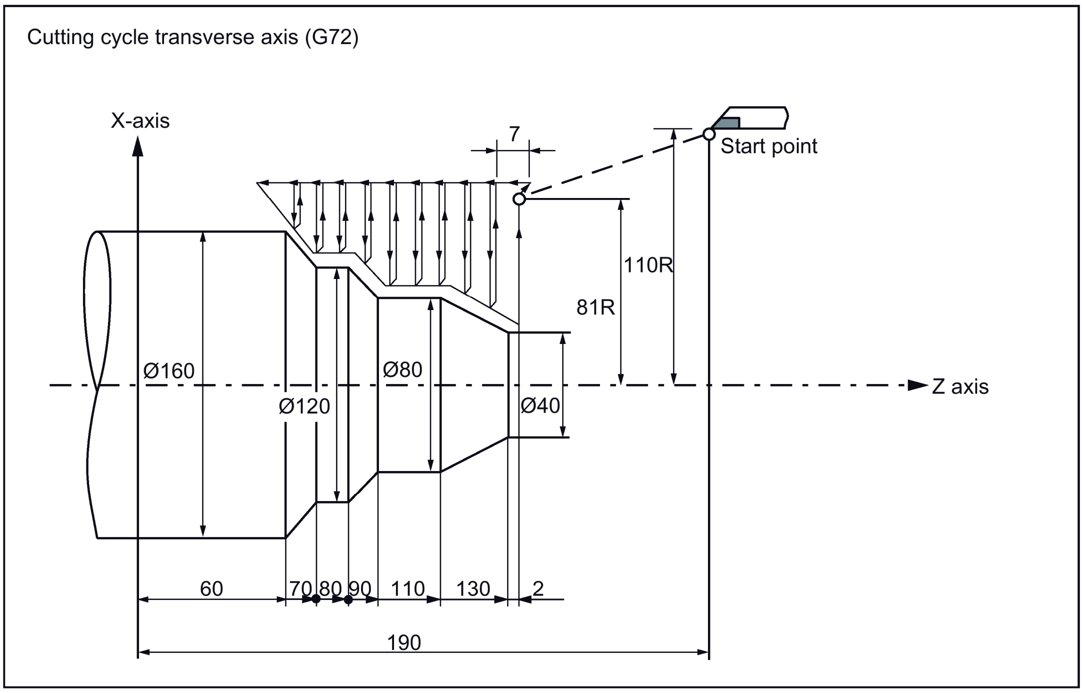
Stock removal cycle, longitudinal axis:



(Diameter programming, input metric)

```
N010 G00 X200.0 Z220.0
N011 X142.0 Z171.0
N012 G71 U4.0 R1.0
N013 G71 P014 Q020 U4.0 W2.0 F0.3 S550
N014 G00 X40.0 F0.15 S700
N015 G01 Z140.0
N016 X60.0 Z110.0
N017 Z90.0
N018 X100.0 Z80.0
N019 Z60.0
N020 X140.0 Z40.0
N021 G70 P014 Q020
N022 G00 X200 Z220
```

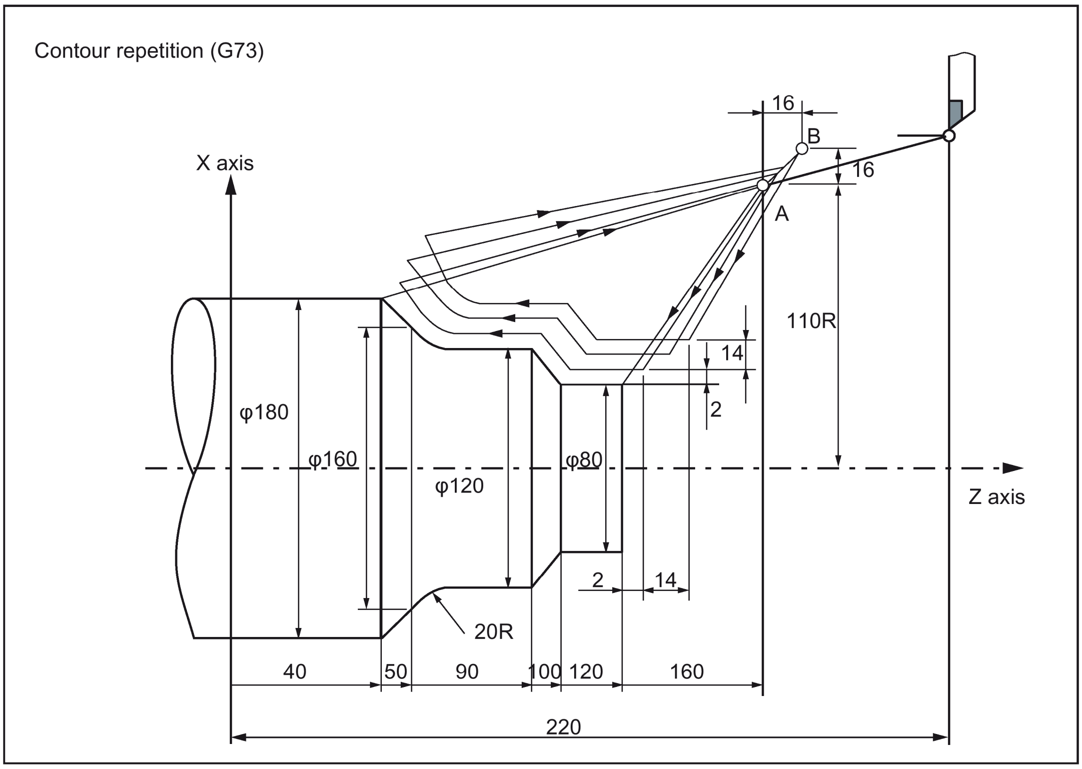
Stock removal cycle transverse axis:



(Diameter programming, input metric)

```
N010 G00 X220.0 Z190.0
N011 G00 X162.0 Z132.0
N012 G72 W7.0 R1.0
N013 G72 P014 Q019 U4.0 W2.0 F0.3
N014 G00 Z59.5 F0.15 S200
N015 G01 X120.0 Z70.0
N016 Z80.0
N017 X80.0 Z90.0
N018 Z110.0
N019 X36.0 Z132.0
N020 G70 P014 Q019
N021 X220.0 Z190.0
```

Contour repetition:



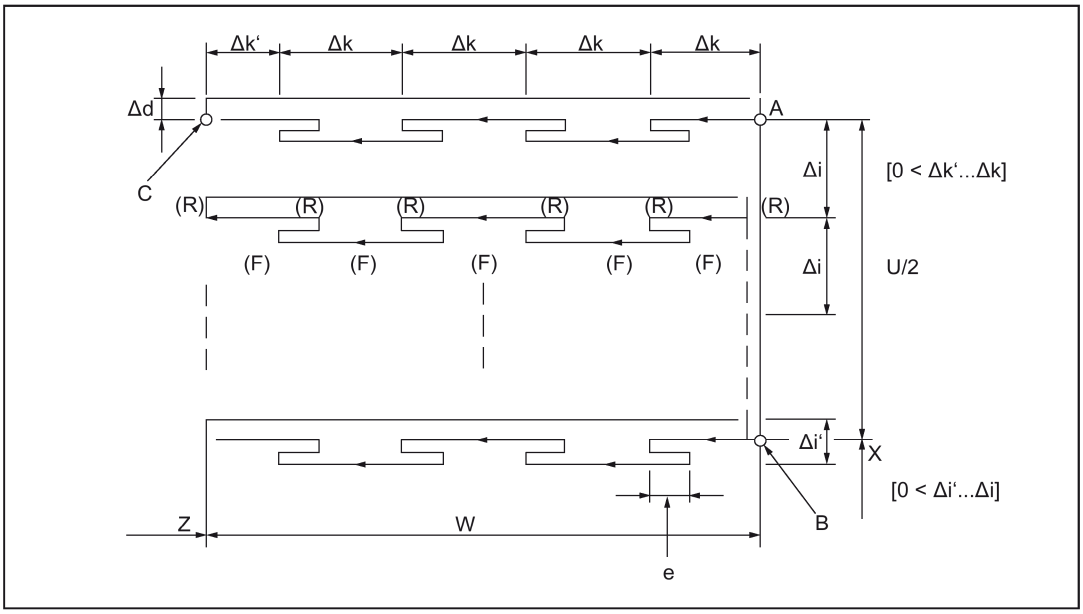
(Diameter programming, input metric)

```
N010 G00 X260.0 Z220.0
N011 G00 X220.0 Z160.0
N012 G73 U14.0 W14.0 R3
N013 G73 P014 Q020 U4.0 W2.0 F0.3 S0180
N014 G00 X80.0 Z120.0
N015 G01 Z100.0 F0.15
N017 X120 Z90.0
N018 Z70
N019 G02 X160.0 Z50.0 R20.0
N020 G01 X180.0 Z40.0 F0.25
N021 G70 P014 Q020
N022 G00 X260.0 Z220.0
```

Multiple repetitive grooving cycles in the longitudinal axis (G74)

In the cycle called with G74 a machining is executed parallel to the Z axis with a chip breakage.

Cutting path in case of a deep hole drilling cycle:



Format

G74 R... ;

R: (d), Retraction amount

This value is modal and remains effective till another value is programmed. The value can also be entered via USER DATA, _ZSFI[29], but this value is overwritten by the value of the program command.

G74 X(U)... Z(W)... P... Q... R... F...(f) ;

X: Starting point X (absolute positional data)

U: Starting point X (incremental positional data)

Z: Starting point Z (absolute positional data)

W: Starting point Z (incremental positional data)

P: Infeed amount (Δi) in X direction (without sign)

Q: Infeed amount (Δk) in Z direction (without sign)

R: Retraction amount (Δd) at the groove base

F: Feedrate

Note

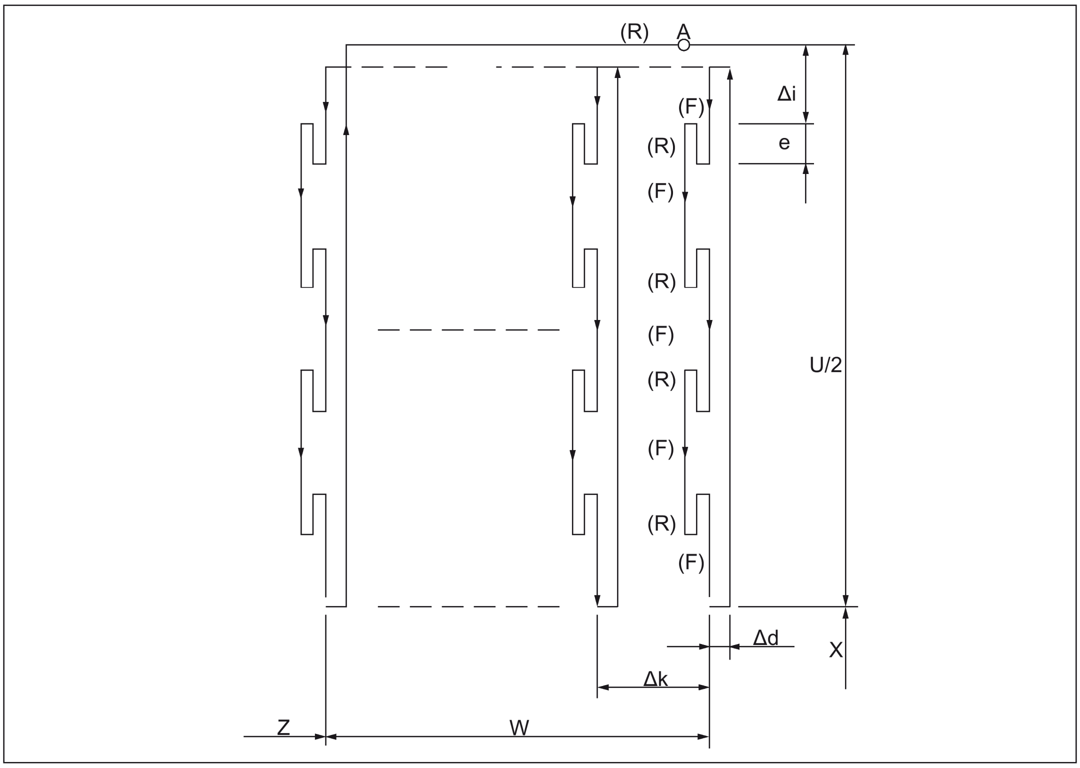
Multiple repetitive grooving cycles in the longitudinal axis

- While "e" and Δ"d" are determined via the address R, the significance of "e" and "d" is determined by the specification of the address X (U). Δ"d" is always used when X(U) is also specified.
- The execution cycle is executed through the command G74 with the specification of X (U).
- If the cycle is used for drilling, the addresses X(U) and P cannot be used.

Multiple repetitive grooving cycles in the transverse axis (G75)

In the cycle called with G75 a machining is executed parallel to the X axis with a chip breakage.

Cutting path in multiple repetitive grooving cycles in the transverse axis (G75):



Format

G75 R... ;

G75 X(U)... Z(W)... P... Q... R... F... ;

The addresses have the same significance here as in the cycle G74.

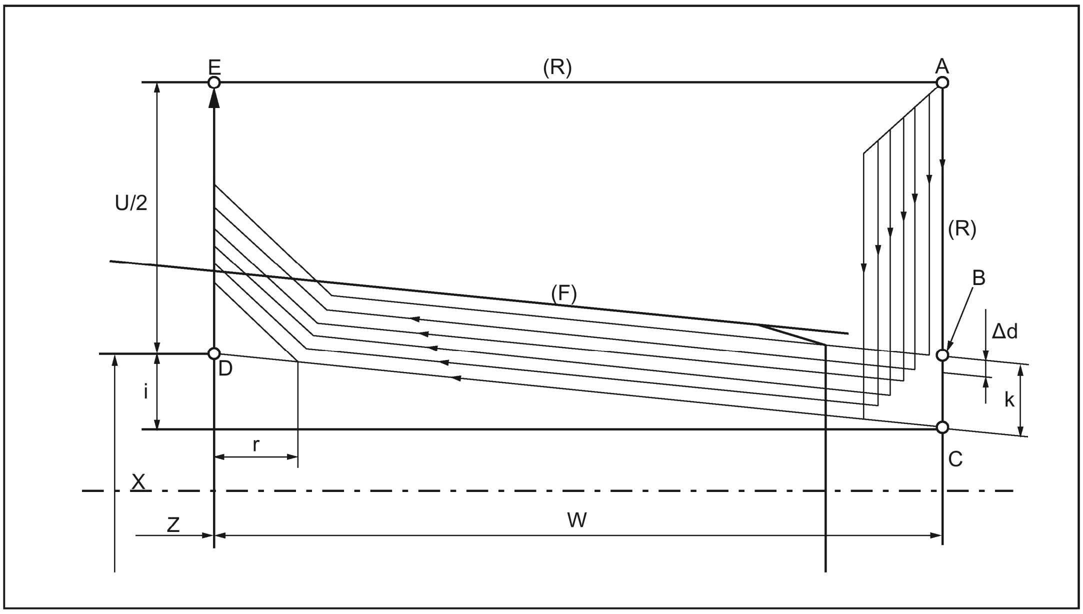
Note

If the cycle is used for drilling, the addresses Z(W) and Q cannot be used.

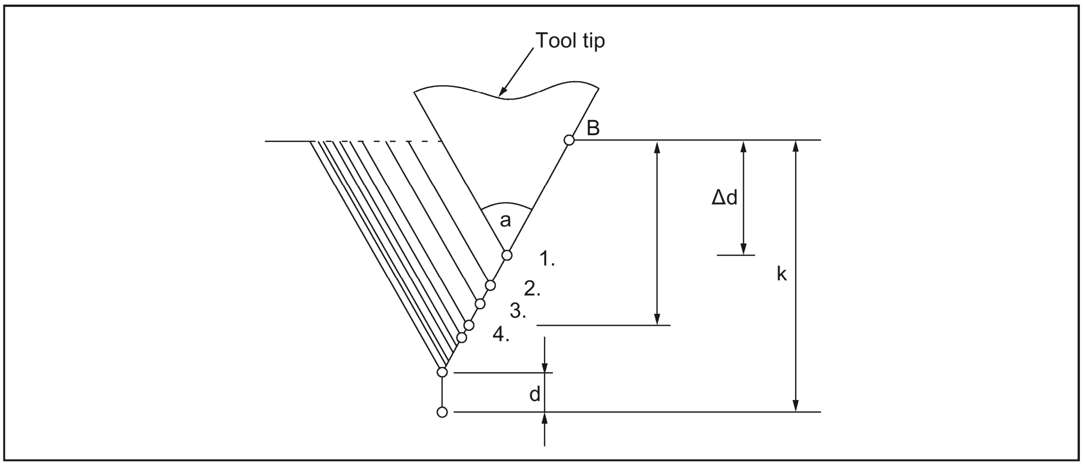
Multiple-thread cutting cycle (G76)

G76 calls an automatic thread-cutting cycle for cutting a cylindrical or a taper thread, in which the infeed takes place in a specific threaded bracket.

Cutting path in case of a cycle for cutting multiple-start threads:



Infeed during thread cutting:



Format

G76 P... (m, r, a) Q... R... ;

P:

m: Number of finishing cuts

This value is modal and remains effective till another value is programmed. The value can also be entered via USER DATA, _ZSFI[24], but this value is overwritten by the value of the program command.

r: Size of the chamfer at the end of the thread (1/10 * thread lead)

This value is modal and remains effective till another value is programmed. The value can also be entered via USER DATA, _ZSFI[26], but this value is overwritten by the value of the program command.

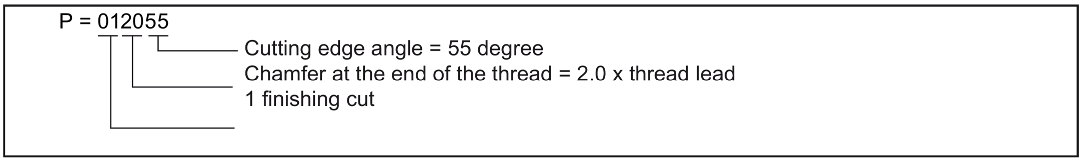
a: Facet angle

This value is modal and remains effective till another value is programmed. The value can also be entered via USER DATA, _ZSFI[25], but this value is overwritten by the value of the program command.

All the parameters specified above are specified simultaneously through the address P.

Example for an address with P:

G76 P012055 Q4 R0.5



Q: Minimum infeed depth (**Δdmin)**, radius value

Always when the cutting depth during a cycle operation (Δd - Δd-1) becomes less than this limiting value, then the cutting depth remains bound to the value specified with the address Q.

This value is modal and remains effective till another value is programmed. The value can also be entered via USER DATA, _ZSFI[27], but this value is overwritten by the value of the program command.

R: Finishing allowance

This value is modal and remains effective till another value is programmed. The value can also be entered via USER DATA, _ZSFI[28], but this value is overwritten by the value of the program command.

G76 X(U)... Z(W)... R... P... Q... F... ;

X, U: Endpoint of thread in the direction of the X axis (Positional data for (X) absolute, for (U) incremental)

Z, W: Endpoint of thread in the direction of Z axis

R: Radius difference for a taper thread **(i)**. i = 0 for simple cylindrical thread

P: Thread depth **(k)**, radius value

Q: Infeed amount for the 1st cut **(Δd)**, radius value

F: Thread lead **(L)**

Note

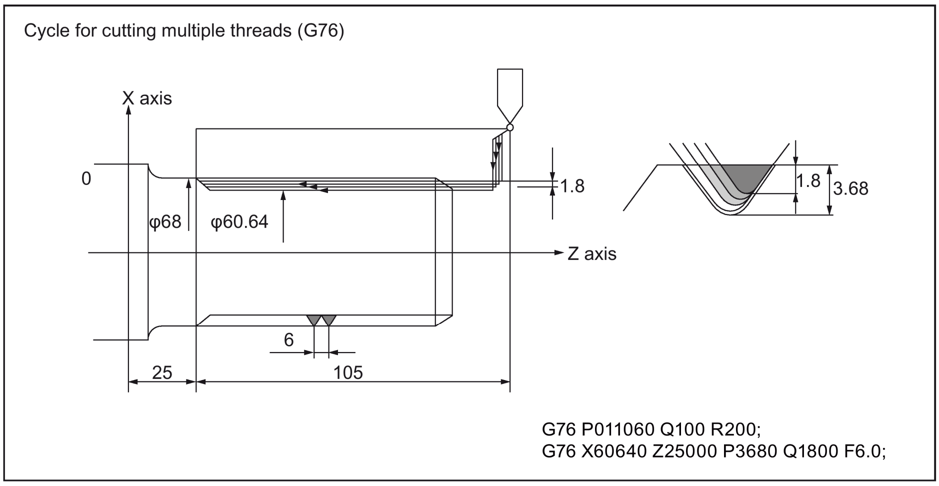
Multiple-thread cutting cycle

- The significance of the data specified with the address characters P, Q and R is determined by the appearance of X (U) and Z (W).
- The execution cycle is executed through the command G76 with the specification of X (U) and Z (W). While using this cycle a "single cut" is performed and the load on the tool tip is reduced.

  The amount of cutting per cycle is kept constant by assignment to the respective cutting depth. Δd on the first path and Δdn on the nth path. Corresponding to the respective sign of the address character, four symmetric sections are considered here.
- The same instructions as for thread cutting with G32 or for the thread cutting cycle with G76 are applicable here.

Examples

Cycle for cutting threads (G76):



Note

Supplementary conditions

- In "MDA" mode the commands G70, G71, G72 or G73 are not permitted; else an alarm 14011 is output. However, G74, G75 and G76 can be used in "MDA" mode.
- In the blocks with G70, G71, G72 or G73 as well as the sequence numbers specified through the addresses P and Q the programming of M98 (subroutine call) and M99 (end of subroutine) is not permitted.
- The following commands cannot be programmed in blocks having the sequence numbers specified through the address characters P and Q:
  - one shot G functions (with the exception of G04 dwell time)
  - G functions of the G group 01 (except G00, G01, G02 and G03)
  - G functions of the G group 06
  - M98/M99
- The programming should not be done in such a way that the final movement of the contour definition for G70, G71, G72 and G73 is concluded with a chamfering or corner rounding. Otherwise, an error message is output.
- In the cycles with G74, G75 and G76 the addresses P and Q use the smallest input increment for specifying the traverse path and the cutting depth.
- In the cycles G71, G72, G73, G74, G75, G76 and G78 no tool nose radius compensation may be performed.

3.4.1.3 Drilling cycle (G80 to G89)

With fixed cycles for hole-machining (G80 to G89) specific movements can be programmed for machining the drill holes, which normally require several command frames made of several single-block commands. The program called with the fixed cycle can be deselected again with G80.

The G functions used for calling the fixed cycles G80 to G89 are the same for all G code systems.

G functions for calling fixed cycles, axis motion pattern of fixed cycles

The G functions used for calling a fixed cycle are given in the table below.

G code	Drilling (- direction)	Machining at the bottom of a hole	Retraction (+ direction)	Applications
G80	-	-	-	Deselection
G83	interrupted cutting feedrate	-	Rapid traverse	Deep hole drilling front face
G84	Cutting feed	Dwell time -> left spindle run	Cutting feed	Tapping front face
G85	Cutting feed	Dwell time	Cutting feed	Drilling front face
G87	interrupted cutting feedrate	Dwell time	Rapid traverse	Deep hole drilling side surface
G88	Cutting feed	Dwell time -> left spindle run	Cutting feed	Tapping side surface
G89	Cutting feed	Dwell time	Cutting feed	Drilling side surface

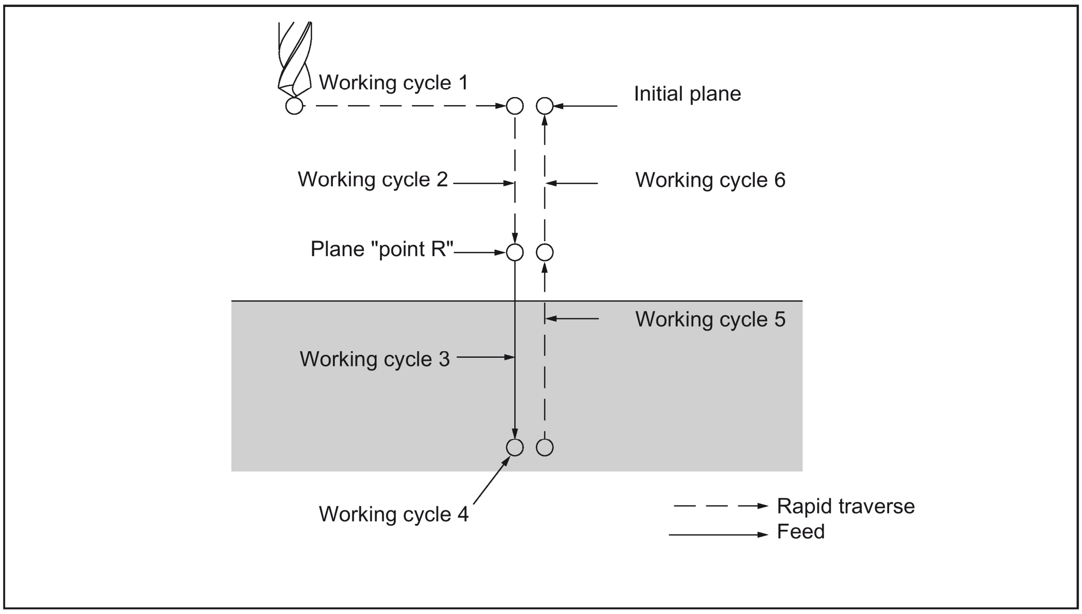
Explanations

When using the fixed cycles, the sequence of operation is generally as described below:

1. Working cycle 1
   Positioning the X, (Z) and C axis
2. Working cycle 2
   Rapid traversing movement to plane R
3. Working cycle 3
   Drilling
4. Working cycle 4
   Machining on drilling base

5. Working cycle 5
   Retraction till plane R
6. Working cycle 6
   Rapid retraction to positioning plane

Sequence of the working cycles in the drilling cycle:



Explanations: Positioning and drilling axis

As shown below, the positioning axis as well as the drilling axis are determined for the drilling through a G function. Thus the C axis and the X or the Z axis correspond to the positioning axis. The drilling axis is mapped through the X or the Z axis: These axes are not used as the positioning axis.

G function	Positioning plane	Drilling axis
G83, G84, G85	X axis, C axis	Z axis
G87, G88, G89	Z axis, C axis	X axis

G83 and G87, G84 and G88 as well as G85 and G89 have the same working cycle sequence except for the drilling axis.

Drilling mode

The G functions (G83-G85, G87-89) are modal and remain active till they are deselected again. As long as these G functions are selected, the drilling mode remains active. The data is retained till the drilling data in the drilling cycle is modified or deselected.

All the necessary drilling data must be specified at the start of a fixed cycle. Data may only be modified during the execution of a fixed cycle.

Repeat

If you wish to make several drilled holes equally spaced, you can specify the number of repetitions in the parameter "K". "K" is effective only in the block, in which it is specified.

The drilling data are stored; however, no drilling is done when K0 is programmed.

Deselection

For deselecting a fixed cycle either G80 is used or a function of the G group 01 (G00, G01, G02, G03).

Symbols and figures

Given below is an explanation of the individual fixed cycles. These symbols are used in the following figures:

Positioning (Rapid traverse G00)

Cutting feed (Linear interpolation G01)

Manual feed

P1 Dwell time

Mα M function for locking the C axis

M(α+1) M function for releasing the C axis

Note

In all fixed cycles the address character R (distance "initial plane - point R") is treated as radius.

However, Z or X (Distance "point R - bottom of a hole) is always treated as diameter or radius, depending upon the type of programming.

Deep hole drilling cycle (G83)/Side deep hole drilling cycle (G87)

Whether a deep hole drilling cycle (chip removal) or a high-speed deep hole drilling cycle (chip breaking) is executed, depends on the setting USER DATA, _ZSFI[20].

If no infeed is specified for the drilling cycle, a normal drilling cycle is executed.

High-speed deep hole drilling cycle (G83, G87) (USER DATA, _ZSFI[20]=0)

In case of the high-speed deep hole drilling cycle, the drill repeats the infeed with the cutting feedrate. This is retracted by a certain amount till the tool has reached the bottom of a hole.

Format

G83 X(U)... C(H)... Z(W)... R... Q... P... F... M... ;

or

G87 Z(W)... C(H)... X(U)... R... Q... P... F... M... ;

X, C or Z, C: Position of the hole

Z or X: Distance from the point R to the bottom of the hole

R_: Distance from the initial plane to the plane R

Q_: Infeed

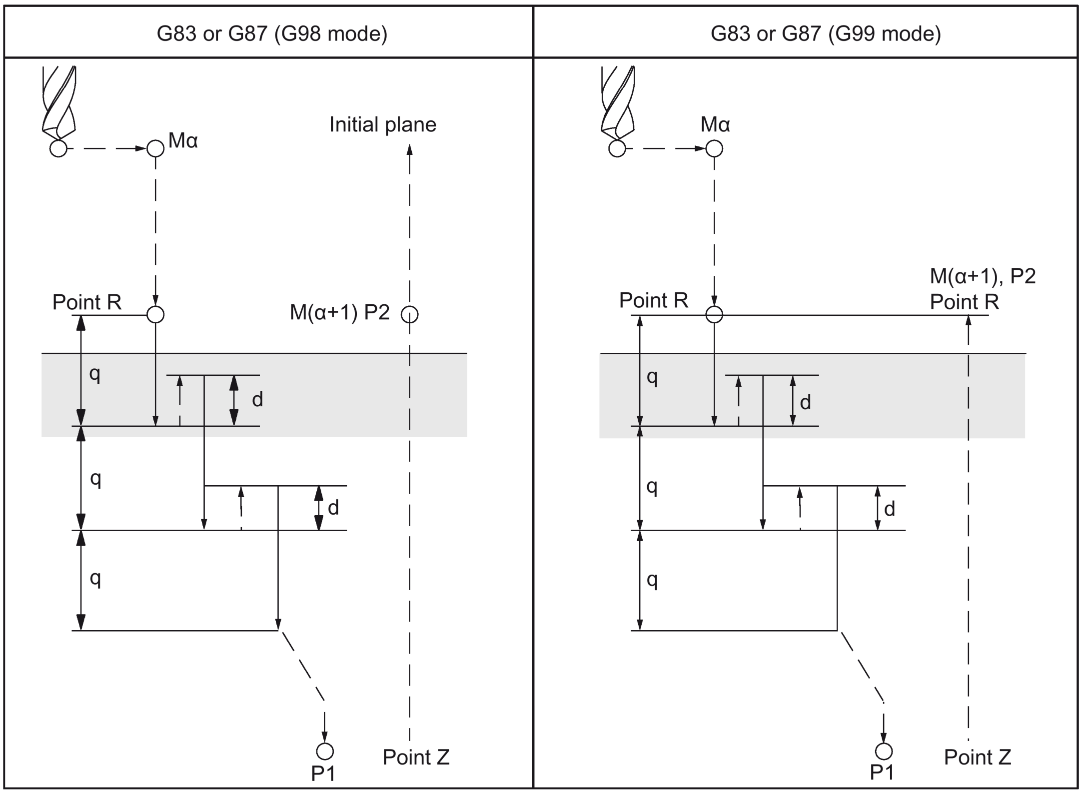
P_: Dwell time at the bottom of a hole

F_: Cutting feedrate

K_: Number of repetitions (if required)

M_: M function for locking the C axis (if needed)

Cycle "high-speed deep hole drilling":



Mα: M function for locking the C axis

M(α+1): M function for releasing the C axis

P1: Dwell time (Program)

P2: Specification of the dwell time in USER DATA, _ZSFR[22]

d: Specification of the retraction amount in USER DATA, _ZSFR[21]

Deep hole drilling cycle (G83, G87) (USER DATA, _ZSFI[20]=1)

In case of the deep hole drilling cycle, the drill repeats the infeed with the cutting feedrate. This is retracted to the plane R till the tool has reached the bottom of a hole.

Format

G83 X(U)... C(H)... Z(W)... R... Q... P... F... M... K... ;

or

G87 Z(W)... C(H)... X(U)... R... Q... P... F... M... K... ;

X, C or Z, C: Position of the hole

Z or X: Distance from the point R to the bottom of the hole

R_: Distance from the initial plane to the plane R

Q_: Infeed

P_: Dwell time at the bottom of a hole

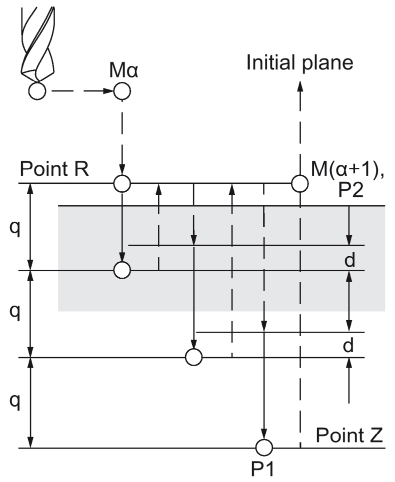
F_: Cutting feedrate

K_: Number of repetitions (if required)

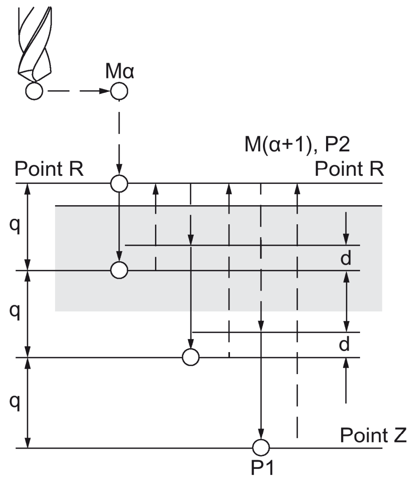
M_: M function for locking the C axis (if needed)

Deep hole drilling cycle:

G83 or G87 (G98 mode)



G83 or G87 (G99 mode)



Mα: M function for locking the C axis

M(α+1): M function for releasing the C axis

P1: Dwell time (Program)

P2: Specification of the dwell time in USER DATA, _ZSFR[22]

d: Specification of the retraction amount in USER DATA, _ZSFR[21]

Example

```
M3 S2500                                              ;Rotate the drilling tool
G00 X100.0 C0.0                                       ;Positioning X and C axis
G83 Z-35.0 R-5.0 Q5000 F5.0                           ;Machining of hole 1
C90.0                                                 ;Machining of hole 2
C180.0                                                ;Machining of hole 3
C270.0                                                ;Machining of hole 4
G80 M05                                               ;Deselection of the cycle and
                                                      ;Stopping of the drilling tool
```

Drilling cycle (G83 or G87)

If no value is programmed for the infeed (Q), then a normal drilling cycle is executed. In this case the tool is retracted from the bottom of a hole with rapid transverse.

Format

G83 X(U)... C(H)... Z(W)... R... P... F... M... K... ;

or

G87 Z(W)... C(H)... X(U)... R... P... F... M... K... ;

X, C or Z, C: Position of the hole

Z or X: Distance from the point R to the bottom of the hole

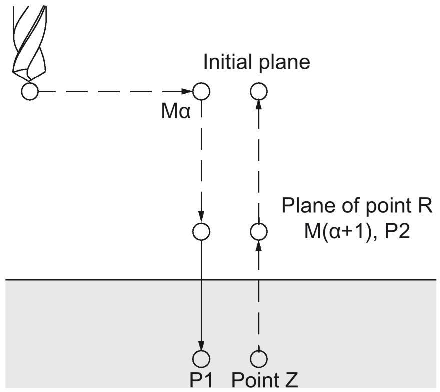
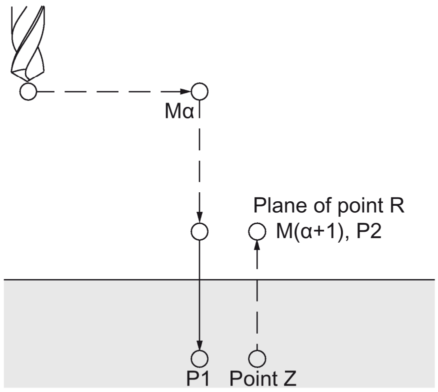
R_: Distance from the initial plane to the plane R

P_: Dwell time at the bottom of a hole

F_: Cutting feedrate

K_: Number of repetitions (if required)

M_: M function for locking the C axis (if needed)

G83 or G87 (G98 mode)	G83 or G87 (G99 mode)
	

Mα: M function for locking the C axis

M(α+1): M function for releasing the C axis

P1: Dwell time (Program)

P2: Specification of the dwell time in USER DATA, _ZSFR[22]

Example

```
M3 S2500                                      ;Rotate the drilling tool
G00 X100.0 C0.0                               ;Positioning X and C axis
G83 Z-35.0 R-5.0 P500 F5.0                    ;Machining of hole 1
C90.0                                         ;Machining of hole 2
C180.0                                        ;Machining of hole 3
C270.0                                        ;Machining of hole 4
G80 M05                                       ;Deselection of the cycle and
                                              ;Stopping of the drilling tool
```

After the programmed cutting depth has been reached for each cutting feedrate Q, the retraction is made to the reference plane R with rapid traverse. The approach motion for a new cut is also done again with rapid traverse, and that too by the path (d), which can be set in USER DATA, _ZSFR[10]. The path d and the cutting depth for each cutting feedrate Q is traversed with cutting feedrate. Q is to be specified incrementally without sign.

Note

If _ZSFR[10]

- > 0 = value is used for the derivative path "d" (minimum path 0.001)
- = 0 The limit point distance d is calculated internally in the cycles as follows:
  - If the drilling depth is 30 mm, then the value for the derivative path is always 0.6 mm.
  - For higher drilling depths the formula drilling depth/50 is used (maximum value 7 mm).

Tapping cycle front face (G84), side surface (G88)

In this cycle the direction of rotation of the spindle at the bottom of a hole is reversed.

Format

G84 X(U)... C(H)... Z(W)... R... P... F... M... K... ;

or

G88 Z(W)... C(H)... X(U)... R... P... F... M... K... ;

X, C or Z, C: Position of the hole

Z or X: Distance from the point R to the bottom of the hole

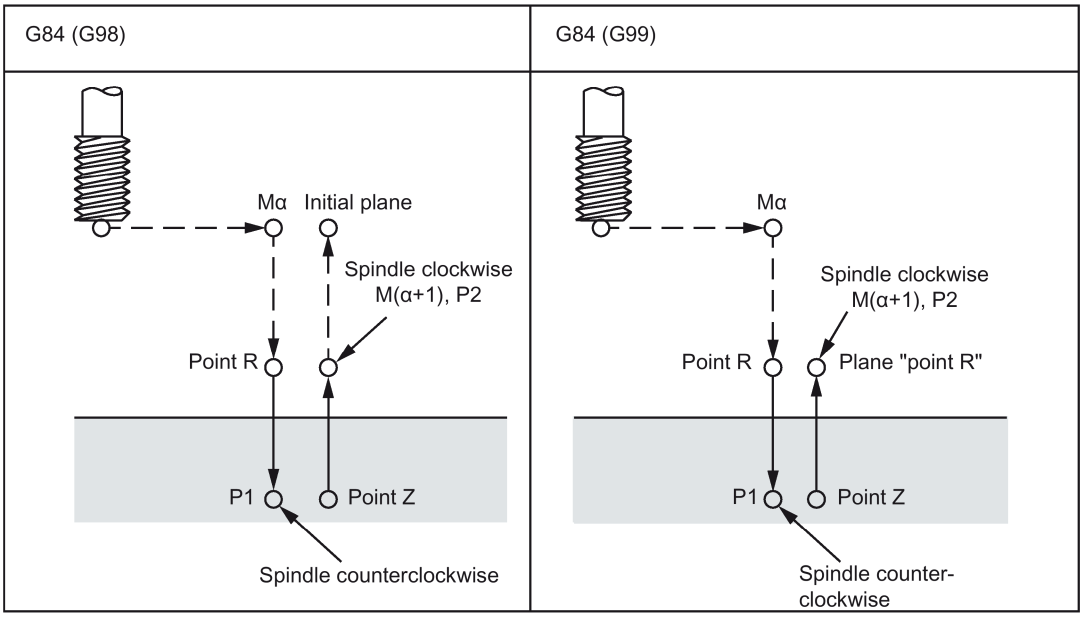
R_: Distance from the initial plane to the plane R

P_: Dwell time at the bottom of a hole

F_: Cutting feedrate

K_: Number of repetitions (if required)

M_: M function for locking the C axis (if needed)



P2: Specification of the dwell time in USER DATA, _ZSFR[22]

Explanations

During tapping the spindle rotates in clockwise direction in the direction of the bottom of a hole; thereafter, the direction of rotation is reversed for the return. The cycle is continued till the tool has been retracted completely.

Example

```
M3 S2500                                    ;Rotate the tapping tool
G00 X100.0 C0.0                             ;Positioning X and C axis
G84 Z-35.0 R-5.0 P500 F5.0                  ;Machining of hole 1
C90.0                                       ;Machining of hole 2
C180.0                                      ;Machining of hole 3
C270.0                                      ;Machining of hole 4
G80 M05                                     ;Deselection of the cycle and
                                            ;Stopping of the drilling tool
```

Drilling cycle front face (G85), side surface (G89)

Format

G85 X(U)... C(H)... Z(W)... R... P... F... K... M... ;

or

G89 Z(W)... C(H)... X(U)... R... P... F... K... M... ;

X, C or Z, C: Position of the hole

Z or X: Distance from the point R to the bottom of the hole

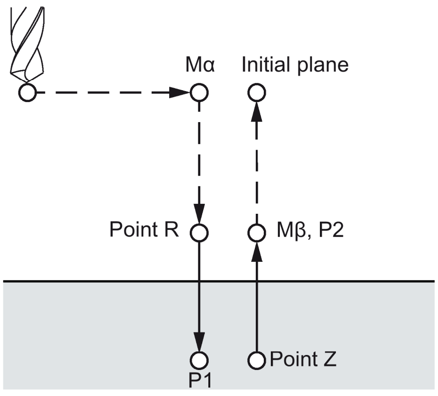
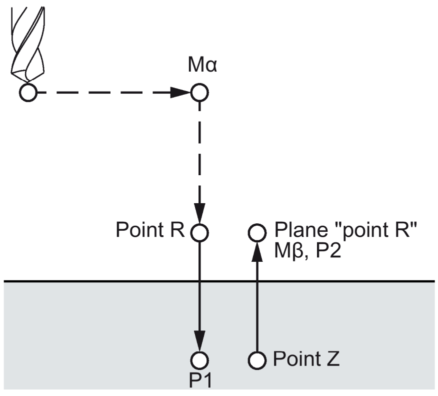
R: Distance from the initial plane to the plane R

P: Dwell time at the bottom of a hole

F: Cutting feedrate

K: Number of repetitions (if required)

M: M function for locking the C axis (if needed)

G85 or G89 (G98 mode)	G85 or G89 (G99 mode)
	

P2: Specification of the dwell time in USER DATA, _ZSFR[22]

Explanations

After the positioning at the bottom of a hole a traversing movement takes place to point R with rapid traverse. Thereafter, drilling is done from point R to point Z, and a return is done to point R.

Example

```
M3 S2500                                   ;Rotate the drilling tool
G00 X50.0 C0.0                             ;Positioning X and C axis
G85 Z-40.0 R-5.0 P500 M31                  ;Machining of hole 1
C90.0 M31                                  ;Machining of hole 2
C180.0 M31                                 ;Machining of hole 3
C270.0 M31                                 ;Machining of hole 4
G80 M05                                    ;Deselection of the cycle and
                                           ;Stopping of the drilling tool
```

Deselection of the fixed cycle for drilling (G80)

Fixed cycles can be deselected with G80.

Format

G80;

Explanations

The fixed cycle for drilling is deselected, and a transition is made to the normal operation again.

3.4.2 Programmable data input

3.4.2.1 Changing the tool offset value (G10)

With the command "G10 P ··· X(U) ··· Y(V) ··· Z(W) ··· R(C) ··· Q ;", the available tool offsets can be overwritten. However, it is not possible to create new tool offsets.

For description of the addresses, see the table below:

Address	Description
P	Tool offset number (see explanation below)
X	Tool offset for the X axis (absolute, incremental)
Y	Tool offset for the X axis (absolute, incremental)
Z	Tool offset for the Z axis (absolute, incremental)

Address	Description
U V W	Tool offset for the X axis (incremental) Tool offset for the X axis (incremental) Tool offset for the Z axis (incremental)
R	Tool nose radius compensation (absolute)
C	Tool nose radius compensation (incremental)
Q	Length of cutting edge

Address character P

With the address character P, the tool compensation number is specified and at the same time also, whether the offset value is to be modified for the tool geometry or for the wear. The value specified with the address character P depends on the setting in MD $MC_EXTERN_FUNCTION_MASK, bit 1:

$MC_EXTERN_FUNCTION_MASK, Bit1 = 0

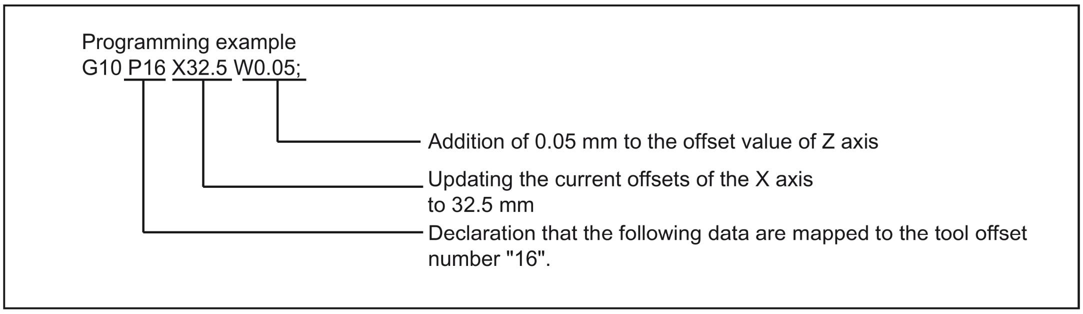
P1 to P99: Writing the tool offset

P100 + (1 to 1500): Writing the tool offset

$MC_EXTERN_FUNCTION_MASK, Bit1 = 1

P1 to P9999: Writing of tool wear

P10000 + (1 to 1500): Writing the tool geometry



Write work offsets

With the commands "G10 P00 X (U) ... Z (W) ... C (H) ... ;" the work offsets can be written and updated in a part program. The offset values remain unchanged for the non-programmed axes.

X, Z, C: Absolute or incremental (for G91) amount of offset in the workpiece coordinate system

U, W, H: Incremental amount of offset in the workpiece coordinate system

3.4.2.2 M function for calling subroutines (M98, M99)

This function can be used if subroutines are stored in the part program memory. Subroutines that are registered in the memory and whose program numbers are assigned can be called and executed any number of times.

Commands

The following M functions are used to call the subroutines.

M function	Function
M98	Subprogram call
M99	End of subroutine

Subroutine call (M98)

- M98 Pnnnnmmmm
  m: Program no. (max. 4 digits)
  n: No. of repetitions (max. 4 digits)

Before using program M98 Pnnnnmmmm to call a program, name the program correctly, that is, add the program number always to 4 digits with 0.

- If for example, M98 P21 is programmed, the part program memory is browsed by program name 21.mpf and the subroutine is executed once. To call the subroutine three times, one must program M98 P30021. An alarm is output if the specified program no. is not found.
- A nesting of subroutines is possible, up to 16 subroutines are allowed. An alarm is output if more subroutine levels are assigned than is allowed.

End of subroutine (M99)

A subroutine is ended with the command M99 Pxxxx and program processing is continued in Block No. Nxxxx. The control system first searches forward for the block number (from the subroutine call up to the end of the program). If no matching block number is found, the part program is eventually searched in the reverse direction (in the direction of the part start of program).

If M99 is without a block number (Pxxxx) in a main program, the control goes to the start of the main program and the main program is processed afresh. In case of M99 with navigation to the block number in the main program (M99xxxx), the block number is always searched from the start of program.

3.4.3 Eight-digit program number (M98, G65/G66)

An eight-digit program number selection is activated with the machine data 20734 $MC_EXTERN_FUNCTION_MASK, Bit 6=1. This function affects M98 and G65/66.

y: Number of program runs

x: Program number

Subprogram call

$MC_EXTERN_FUNCTION_MASK, Bit 6 = 0

M98 Pyyyyxxxx or

M98 Pxxxx Lyyyy

Max. four-digit program number

Addition of program number always to 4 digits with 0

Example:

M98 P20012: calls 0012.mpf 2 flows

M98 P123 L2: calls 0123.mpf 2 flows

$MC_EXTERN_FUNCTION_MASK, Bit 6 = 1

M98 Pxxxxxxxx Lyyyy

There is no extension with 0, even if the program number has less than 4 digits.

The programming of number of passes and program number in P (Pyyyyxxxxx) is not possible; the number of passes must always be programmed with L!

Example:

M98 P123: calls 123.mpf 1 Pass

M98 P20012: calls 20012.mpf 1 Pass

Caution: This is no longer compatible with ISO dialect original

M98 P12345 L2: calls 12345.mpf 2 Passes

Modal and blockwise Macro G65/G66

$MC_EXTERN_FUNCTION_MASK, Bit 6 = 0

G65 Pxxxx Lyyyy

Addition of program number to 4 digits with 0. Program number with more than 4 digits leads to an alarm.

$MC_EXTERN_FUNCTION_MASK, Bit 6 = 1

G65 Pxxxx Lyyyy

There is no extension with 0, even if the program number has less than 4 digits. A program number with more than 8 digits leads to an alarm.

3.4.4 Measuring functions (G31)

Measuring with "delete distance-to-go" (G31)

With "G31 X... Y... Z... F... ;" the measuring with "delete distance-to-go" is possible. If the measuring input of the 1st probe is present during the linear interpolation, the linear interpolation is interrupted and the distance-to-go of the axes is deleted. The program is continued with the next block.

Format

G31 X... Y... Z... F_;

G31: non-modal G function (active only in the block in which it is programmed)

PLC signal "measuring input = 1"

With the increasing edge of the measuring input 1, the current axis positions are stored in the axial system parameters or $AA_MM[<Axis>] $AA_MW[<Axis>]. These parameters can be read in Siemens mode.

$AA_MW[X]	Saving the coordinate values for the X axis in the workpiece coordinate system
$AA_MW[Z]	Saving the coordinate values for the Z axis in the workpiece coordinate system
$AA_MM[X]	Saving the coordinate values for the X axis in the machine coordinate system
$AA_MM[Z]	Saving the coordinate values for the Z axis in the machine coordinate system

---

Note

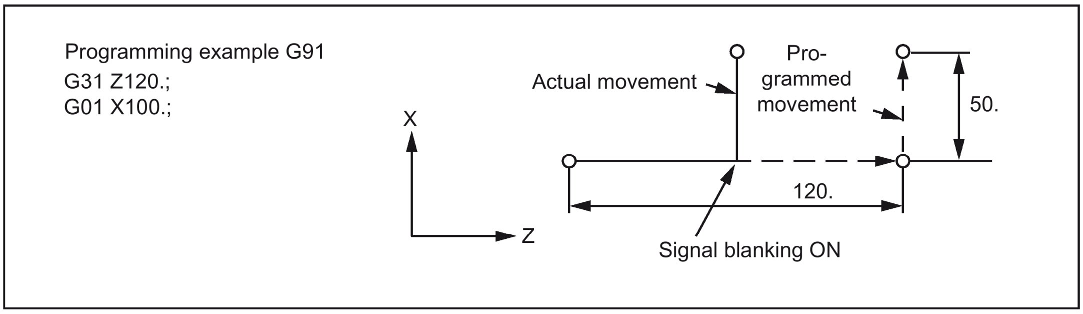
If G31 is activated, while the measuring signal is still active, the alarm 21700 is output.

Program continuation after the measuring signal

If in the next block incremental axis positions are programmed, then these axis positions refer to the measuring point. That is, the reference point for the incremental position is the axis position, at which the distance-to-go deletion is executed through the measuring signal.

If the axis positions are programmed absolute in the next block, then the programmed positions are traversed.

Programming example:



3.4.5 Macro programs

Macros comprise of several part program blocks and are completed with M99. In principle, macros are subroutines, which are called with G65 Pxx or G66 Pxx in the part program.

Macros called with G65 are non-modal. Macros called with G66 are modal and are deselected again with G67.

3.4.5.1 Differences with subroutines

Along with the call for macro programs (G65, G66) parameters can be specified, which can be evaluated in macro programs. On the other hand, no parameters can be specified in the subroutine calls (M98).

3.4.5.2 Macro program call (G65, G66, G67)

Macro programs are generally executed immediately after their call.

The procedure of calling a macro program is described in the following table.

Call method	Command code	Remarks
Simple call	G65	
Modal call (a)	G66	Deselection through G67

Simple call (G65):
Format

G65 P_ L_ ;

A macro program to which a program number was assigned with "P" is called and executed "L" times by specifying "G65 P ... L... <Argument>; ".

The required parameters must be programmed in the same block (with G65).

Explanation

In a part program block containing G65 or G66, the address Pxx is interpreted as program number of the subroutine in which the macro functionality is programmed. The number of passes of the macro can be defined with the address Lxx. All other addresses in this part program block are interpreted as transfer parameters and their programmed values are stored in the system variables $C_A to $C_Z. These system variables can be read in the subroutine and evaluated for the macro functionality. If other macros with parameter transfer are called in a macro (subroutine), then the transfer parameters in the subroutine must be saved in internal variable before the new macro call.

To enable internal variable definitions, one must switch automatically to Siemens mode during macro call. One can do this by inserting the instruction PROC<Program name> in the first line of the macro program. If another macro call is programmed in the subroutine, then the ISO-dialect-mode must be reselected in advance.

For the P and L commands, see the table below:

Address	Description	Number of digits
P	Program number	4 to 8 digits
L	Number of repetitions	

System variables for the addresses I, J, K

As the addresses I, J, and K can be programmed up to 10 times in a block containing macro call, the system variables of these addresses must be accessed with an array index. The syntax of these three system variables thus is $C_I[..], $C_J[..], $C_K[..]. The values remain in the programmed sequence in the array. The number of I, J, K addresses programmed in the block is given in the variables $C_I_NUM, $C_J_NUM, $C_K_NUM.

The transfer parameters I, J, K for macro calls are treated in each case as one block even if the individual addresses are not programmed. If a parameter is reprogrammed, or a following parameter based on the I, J, K sequence was programmed, it belongs to the next block.

The system variables $C_I_ORDER, $C_J_ORDER, $C_K_ORDER are set to detect the programming sequence in ISO mode. These are identical arrays of $C_I, $C_K and they contain the associated numbers of the parameters.

Note

The transfer parameters can be read only in the subroutine in Siemens mode.

Example:

```
N5 I10 J10 K30 J22 K55 I44 K33
  Block1  Block2  Block3
$C_I[0]=10
$C_I[1]=44
$C_I_ORDER[0]=1
```

```
$C_I_ORDER[1]=3

$C_J[0]=10
$C_J[1]=22
$C_J_ORDER[0]=1
$C_J_ORDER[1]=2

$C_K[0]=30
$C_K[1]=55
$C_K[2]=33
$C_K_ORDER[0]=1
$C_K_ORDER[1]=2
$C_K_ORDER[2]=3
```

Note

$C_I[0] is a DIN code. To use this code in ISO mode, machine data 20734 $MC_EXTERN_FUNCTION_MASK, Bit 3=1 must be set, with the default value being 800H.

Cycle parameter $C_x_PROG

In the ISO-dialect-0 mode, the programmed values can be evaluated in different ways, depending on the programming method (integer or actual value). The different evaluation is activated through a machine data.

If the MD is set, the control system responds as in the following example:

X100 ; X axis is traversed by 100 mm (100. with point) => actual value

Y200 ; Y-axis is traversed by 0.2 mm (200 without point) => integer value

If the addresses programmed in the block are used as transfer parameters of cycles, then the programmed values always exist as real values in the $C_x variables. For integer values, one cannot take recourse to the programming method (real/integer) in the cycles any more, and therefore there is no evaluation of the programmed values with the correct conversion factor.

There are two system variables $C_TYP_PROG. $C_TYP_PROG for information as to whether REAL or INTEGER programming was undertaken. The structure is the same as that of $C_ALL_PROG and $C_INC_PROG. If the value is programmed as INTEGER, then Bit is set to 0, for REAL it is set to 1. If the value is programmed over a variable $<Number>, then the corresponding bit is also set to 1.

Example:

P1234 A100. X100 -> $C_TYP_PROG == 1.

Only Bit 0 is present, because only A was programmed as REAL.

P1234 A100. C20. X100 -> $C_TYP_PROG == 5.

Bit 1 and Bit 3 (A and C) are present.

Restrictions:

A maximum of ten I, J, K parameters can be programmed in each block. Only one bit each is provided for I, J, K in the variable $C_TYP_PROG. Hence in $C_TYP_PROG the corresponding bit for I, J and K is always set to 0. Therefore it cannot be derived whether I, J or K is programmed as REAL or as INTEGER.

Modal call (G66, G67)

A modal macro program is called with G66. The specified macro program is executed only if the specified conditions are fulfilled.

- The modal macro program is activated on specifying "G66 P... L... <Parameters>;". The transfer parameters are handled as in G65.
- G66 is deselected by G67.

For modal call conditions, see the table below:

Call conditions	Function for mode selection	Function for mode deselection
after executing a traversing command	G66	G67

Specification of a parameter

The transfer parameters are defined by programming an Address A - Z.

Interrelation between addresses and system variables

Interrelation between addresses and variables	
Address	**System variable**
A	$C_A
B	$C_B
C	$C_C
D	$C_D
E	$C_E
F	$C_F
H	$C_H
I	$C_I[0]
J	$C_J[0]
K	$C_K[0]
M	$C_M
Q	$C_Q
R	$C_R
S	$C_S
T	$C_T
U	$C_U
V	$C_V
W	$C_W
X	$C_X
Y	$C_Y
Z	$C_Z

Interrelation between addresses and system variables

To be able to use I, J and K, these must be specified in the I, J, K sequence.

As the I, J and K addresses in a block containing a macro call can be programmed up to 10 times, access to the system variables within the macro program for these addresses must take place with an index. The syntax of these three system variables thus is $C_I[..], $C_J[..], $C_K[..]. The corresponding values are saved in the matrix in the sequence in which they were programmed. The number of I, J, K addresses programmed in the block is saved in the variables $C_I_NUM, $C_J_NUM and $C_K_NUM.

Unlike for the remaining variables, one index must always be specified while reading the three variables. The index "0" is always used for cycle calls (e.g. G81), for example, N100 R10 = $C_I[0]

Interrelation between addresses and variables	
Address	**System variable**
A	$C_A
B	$C_B
C	$C_C
I1	$C_I[0]
J1	$C_J[0]

Interrelation between addresses and variables	
K1	$C_K[0]
I2	$C_I[1]
J2	$C_J[1]
K2	$C_K[1]
I3	$C_I[2]
J3	$C_J[2]
K3	$C_K[2]
I4	$C_I[3]
J4	$C_J[3]
K4	$C_K[3]
I5	$C_I[4]
J5	$C_J[4]
K5	$C_K[4]
I6	$C_I[5]
J6	$C_J[5]
K6	$C_K[5]
I7	$C_I[6]
J7	$C_J[6]
K7	$C_K[6]
I8	$C_I[7]
J8	$C_J[7]
K8	$C_K[7]
I9	$C_I[8]
J9	$C_J[8]
K9	$C_K[8]
I10	$C_I[9]
J10	$C_J[9]
K10	$C_K[9]

Note

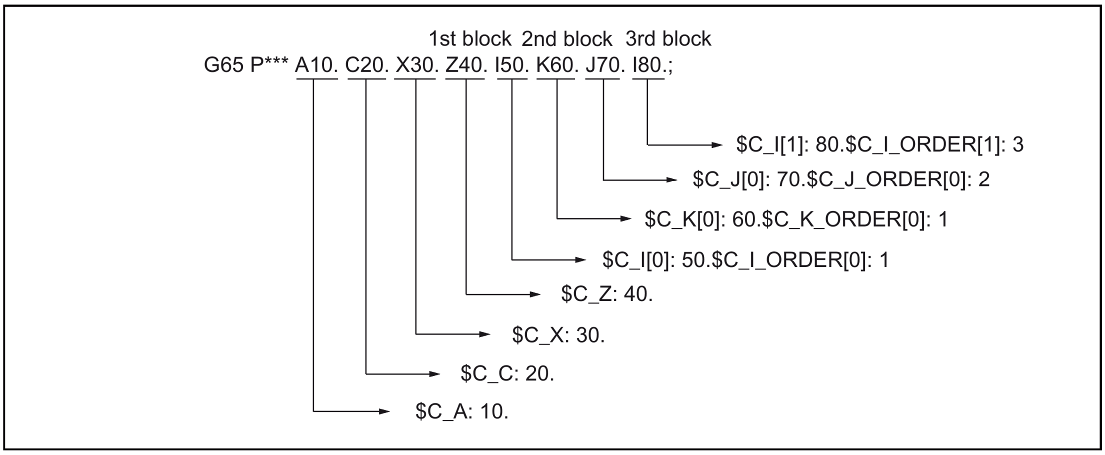
If more than one block of I, J or K addresses are specified, then the sequence of the addresses for each block of I/J/K is determined in such a way that the numbers of the variables are defined according to their sequence.

Example of entering a parameter

The value of the parameter contains a sign and a decimal point, independently of the address.

The value of the parameters is always saved as actual value.

Example of entering an argument:



Execution of macro programs in the Siemens and ISO modes

A called macro program can be called either in Siemens mode or in ISO mode. The language mode in which the program is executed is defined in the first block of the macro program.

If a PROC <Program name> instruction exists in the first block of a macro program, then an automatic changeover to Siemens mode is conducted. If this instruction is missing, the processing is done in ISO mode.

The transfer parameters can be saved in local variables by executing a program in Siemens mode. In ISO mode however, it is not possible to store transfer parameters in local variables.

To read transfer parameters in a macro program executed in ISO mode, one must first change over to Siemens mode with the G290 command.

Examples

Main program with macro call:

```
_N_M10_MPF:
N10 M3 S1000 F1000
N20 X100 Y50 Z33
N30 G65 P10 F55 X150 Y100 S2000
N40 X50
N50 ....
N200 M30
```

Macro program in Siemens mode:

```
_N_0010_SPF:
PROC 0010 ; Changeover to Siemens mode
N10 DEF REAL X_AXIS ,Y_AXIS, S_SPEED, FEED
N15 X_AXIS = $C_X Y_AXIS = $C_Y S_SPEED = $C_S FEED = $C_F
N20 G01 F=FEED G99 S=S_SPEED
...
N80 M17
```

Macro program in ISO mode:

```
_N_0010_SPF:
G290; Changeover to Siemens mode,
    ; to read the transfer parameters
N15 X_AXIS = $C_X Y_AXIS = $C_Y S_SPEED = $C_S FEED = $C_F
N20 G01 F=$C_F G99 S=$C_S
N10 G1 X=$C_X Y=$C_Y
G291; Changeover to ISO mode,
N15 M3 G54 T1
N20
...
N80 M99
```

3.4.6 Special functions

3.4.6.1 G05

The command G05 can call any subroutine, similar to the subroutine call "M98 Pxx". In order to accelerate the processing of the program, the subroutine called with G05 can be pre-compiled.

Format

G05 Pxxxxx Lxxx ;

Pxxxxx: Number of the calling program

Lxxx: Number of repetitions

(If "Lxxx" is not specified, L1 is applicable automatically.)

Example

G05 P10123 L3 ;

With this block the program 10123.mpf is called and executed thrice.

Restrictions

- Upon the call of a subroutine with G05 no switchover to Siemens mode is made. The command G05 has the same effect as a subroutine call with "M98 P_".
- Blocks, which contain G05 without the address character P, are ignored and no alarm is output.
- Blocks with G05.1 - regardless of whether with or without the address character P - as well as blocks with G05 P0 or G05 P01, are also ignored without the output of an alarm.

3.4.6.2 Switchover modes for DryRun and skip levels

Changeover of the skip levels (DB3200.DBB2) always represents an intervention in the program run, which has led to a short-term drop in velocity on the path. The same is true of the changeover of the DryRun mode (DryRun = dry run feedrate DB3200.DBX0.6) from DryRunOff to DryRunOn or vice-versa.

All the drops in velocity can be avoided with a changeover mode that is limited in its function.

No drop in velocity is required with a setting machine data 10706 $MN_SLASH_MASK==2 while changing the skip levels (i.e., a new value in the PLC->NCK-Chan interface DB3200.DBB2).

Note

The NCK processes blocks in two steps, the preprocessing and main runs (also pre-travel and main run). The result of the premachining changes to the preprocessing memory. The main machining takes the relevant oldest block out of the preprocessing memory and traverses its geometry.

Note

The premachining is changeover with the setting machine data $MN_SLASH_MASK==2 during a change of the skip level! All blocks located in the preprocessing memory are traversed with the old skip level. The user normally does not have any control over the fill level of the preprocessing memory. The user can see the following effect: **A new skip level is effective "some time" after the changeover.**

Note

The part program command STOPRE vacates the preprocessing memory. If one switches the skip level before STOPRE, then all the blocks after STOPRE are changed over securely. The same is valid for an implicit STOPRE.

No drop in velocity is required while changing the DryRun mode with the setting machine data 10704 $MN_DRYRUN_MASK==2. Here too, only the premachining that leads to the above-mentioned restrictions, is switched. The following analogy is apparent from this: **This will also be active "sometime" after the changeover of the DryRun mode.**

4 Milling

4.1 G code table

SINUMERIK ISO dialect milling

G code		Description
Group 1		
G00 1)	1	Rapid traverse
G01	2	Linear movement
G02	3	Circle/helix in clockwise direction
G03	4	Circle/helix in the counterclockwise direction
Group 2		
G17 1)	1	XY plane
G18	2	ZX plane
G19	3	YZ plane
Group 3		
G90 1)	1	Absolute programming
G91	2	Incremental programming
Group 5		
G93	3	inverse-time feedrate (1/min)
G94 1)	1	Feedrate in [mm/min, inch/min]
G95	2	Revolutional feedrate in [mm/rev, inch/rev]
Group 6		
G20 1)	1	Inch input system
G21	2	Metric input system
Group 7		
G40 1)	1	Deselection of cutter radius compensation
G41	2	Compensation left of contour
G42	3	Compensation to right of contour
Group 8		
G43	1	Positive tool length compensation on
G44	2	Negative tool length compensation on
G49 1)	3	Tool length compensation off
Group 9		
G73	1	High-speed deep hole drilling cycle with chip breakage
G74	2	Left tapping cycle
G76	3	Fine drill cycle
G80 1)	4	Cycle off
G81	5	Drilling cycle counterboring
G82	6	Countersink drilling cycle
G83	7	Deep hole drilling cycle with chip removal
G84	8	Right tapping cycle
G85	9	Boring cycle, retraction with G01 after reaching the end in axis Z, without spindle stop
G86	10	Boring cycle, spindle stops and then retraction with G00 after reaching the end in axis Z
G87	11	Reverse countersinking

G code		Description
G89	12	Boring cycle, stay for a while and then retraction with G01, without spindle rotation direction change
Group 10		
G98 1)	1	Return to starting point in fixed cycles
G99	2	Return to point R in fixed cycles
Group 11		
G50 1)2)	1	Scaling off
G51 2)	2	Scaling on
Group 12		
G66 2)	1	Macro module call
G67 1)2)	2	Delete macro module call
Group 13		
G96	1	constant cutting rate on
G97 1)	2	constant cutting rate off
Group 14		
G54 1)	1	Selecting work offset
G55	2	Selecting work offset
G56	3	Selecting work offset
G57	4	Selecting work offset
G58	5	Selecting work offset
G59	6	Selecting work offset
G54 P0	1	external work offset
Group 15		
G61	1	Exact stop modal
G63	2	Tapping mode
G64 1)	3	Continuous-path mode
Group 16		
G68	1	Rotation ON, 2D/3D
G69 1)	2	Rotation OFF
Group 17		
G15 1)	1	Polar coordinates off
G16	2	Polar coordinates on
Group 18 (non-modal effective)		
G04	1	Dwell time in [s] or spindle revolutions
G05	18	High-speed cycle cutting
G05.1 2)	22	High-speed cycle -> Call CYCLE305
G07.1 2)	16	Cylindrical interpolation
G08	12	Pre-control ON/OFF
G09	2	Exact stop
G10 2)	3	Write work offset/tool offset
G11	4	End parameter entry
G27	13	Checking the reference position
G28	5	1. Approaching a reference point
G30	6	2./3./4. Approaching a reference point
G30.1	19	Reference point position
G31 7		Measuring with "delete distance-to-go"

G code		Description
G52	8	programmable work offset
G53	9	Approach position in machine coordinate system
G60	22	directed positioning
G65 [2)]	10	Macro call
G72.1 [2)]	14	Contour repetition with rotation
G72.2 [2)]	15	Linear contour repetition
G92	11	Setting actual value
G92.1	21	Delete actual value, reset the WKS
Group 22		
G50.1	1	Mirroring on programmed axis OFF
G51.1	2	Mirroring on programmed axis ON
Group 31		
G290 [1)]	1	Selection of Siemens mode
G291	2	Selection of ISO dialect mode

Note

In general, the G functions mentioned in [1)] are defined by the NC during activation of the control system or during RESET. Data about the actual settings can be found in the documentation of your machine manufacturer.

The G functions mentioned in [2)] are optional. Whether the relevant function is available on your control system can be found out from the documentation of your machine manufacturer.

4.2 Drive commands

4.2.1 Interpolation commands

4.2.1.1 Cylindrical interpolation (G07.1)

Randomly running grooves can be cut on cylindrical workpieces with Function G07.1 (cylindrical interpolation). The path of the grooves is programmed with reference to the unwinded, plane surface of the cylinder.

The G functions specified below can be used to switch the operation of cylindrical interpolation on or off.

G function	Function	G group
G07.1	Operation with cylindrical interpolation	16

Format

```
G07.1 A (B, C) r          ;Activation of operation with cylindrical interpolation
G07.1 A (B, C) 0          ;Deselection of operation with cylindrical interpolation
```

A, B, C: Address of the rotary axis

r: Radius of the cylinder

No other commands should be present in the block containing G07.1.

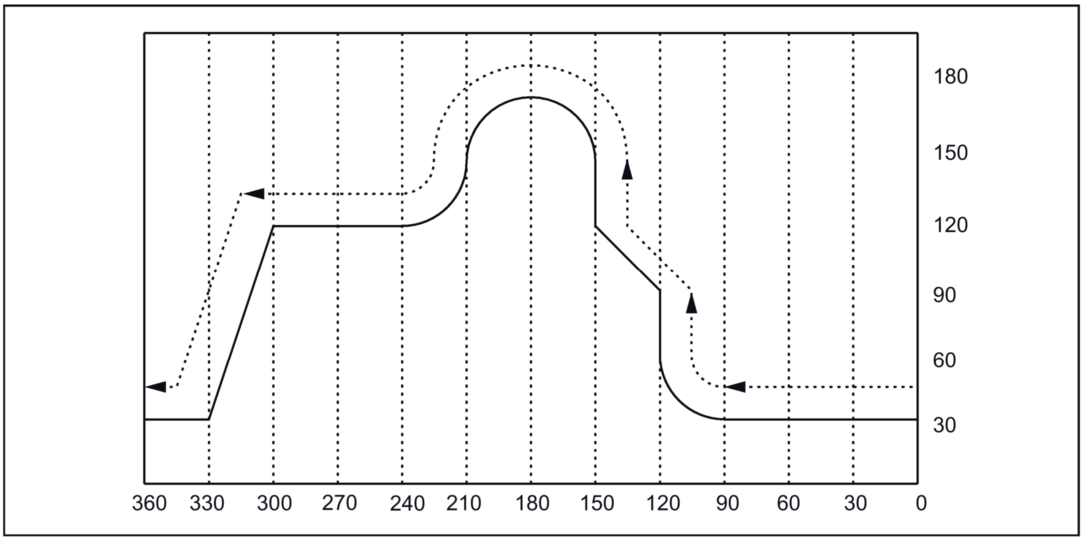
The G07.1 command is modal. Once G07.1 is specified, the cylindrical interpolation remains active till G07.1 A (B, C) is deselected. The cylindrical interpolation is deactivated in closed position or after NC RESET.

Note

G07.1 is based on the Siemens option TRACYL. For this, the appropriate machine data is to be set.
The corresponding data on this is available in the SINUMERIK 808D/SINUMERIK 808D ADVANCED Function Manual, Chapter "Kinematic Transformation (M1)".

Programming example

At the cylindrical plane (it arises because the circumference of a cylindrical workpiece is rolled off), in which the Z-axis is accepted as the linear axis and the A-axis as the rotary axis, the following program is written:



Program

```
M19
G40
G00 Z30. A-10.
G07.1 A57.296                       ;Operation with cylindrical interpolation ON
                                    ;(workpiece radius = 57.926)
G90
G42 G01 A0 F200
G00 X50.
G01 A90. F100
G02 A120. Z60. R30
G01 Z90.
Z120. A150.
Z150.
G03 Z150. A210. R30.
G02 Z120. A240. R30
G01 A300.
Z30. A330.
A360.
G00 X100.
G40 G01 A370.
G07.1 A0                            ;Operation with cylindrical interpolation OFF
G00 A0
```

Programming in operation with cylindrical interpolation

Only the following G functions can be used in cylindrical interpolation: G00, G01, G02, G03, G04, G40, G41, G42, G65, G66, G67, G90, G91 and G07.1. In operation with G00 only those axes can be used that are not involved at the cylindrical plane.

The following axes cannot be used as a positioning axis or a reciprocating axis:

1. The geometry axis in the peripheral direction of the surface of the cylinder (Y axis)
2. The additional linear axis for groove side offset (Z axis)

Relations between the cylindrical interpolation and operations with reference to the coordinate system

- The functions mentioned below should not be used in operation with cylindrical interpolation.
  - Mirroring
  - Scaling (G50, G51)
  - Rotation of the coordinate system (G68)
  - Setting the basic coordinate system
- The relevant overrides (rapid traverse, JOG, spindle speed) are effective.
- On deselecting this operation with cylindrical interpolation, the interpolation plane that was selected before the operation with the cylindrical interpolation was called becomes active again.
- To perform the tool length compensation, the command for the tool length compensation is to be written before specifying the G07.1 command.
- The work offset (G54 - G59) is also to be written before specifying the G07.1 command.

The positioning and interpolation commands, with which the tool path along the programmed contour, such as a straight line or a circular arc, is monitored, are described in the next Section.

4.2.1.2 Rapid traverse (G00)

You can use rapid traverse to position the tool rapidly, to traverse around the workpiece or to approach tool change points.

The following G functions can be used to call the positioning:

G function	Function	G group
G00	Rapid traverse	01
G01	Linear movement	01
G02	Circle/helix in the clockwise direction	01
G03	Circle/helix in the counterclockwise direction	01

Positioning (G00)

Format

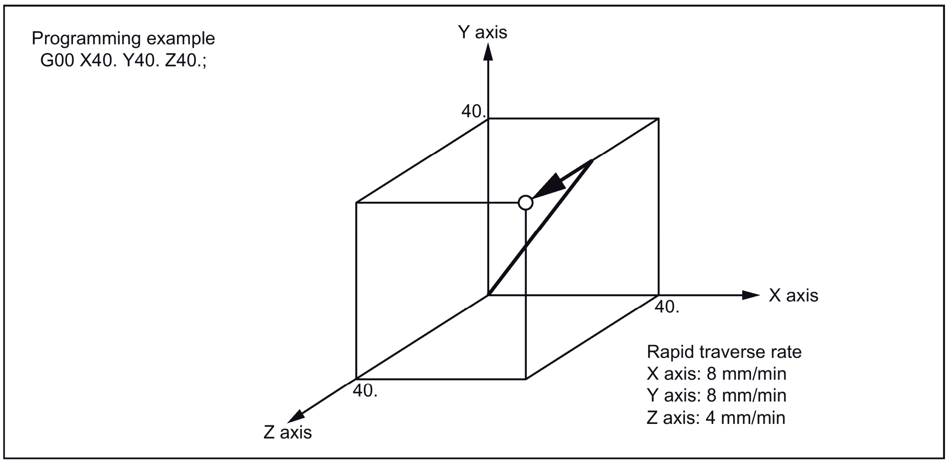
G00 X... Y... Z... ;

Explanation

The tool movement programmed with G00 is executed at the highest possible traversing speed (rapid traverse). The rapid traverse rate is defined separately for each axis in machine data. If the rapid traverse movement is executed simultaneously on several axes, the rapid traverse rate is determined by the axis which requires the most time for its section of the path.

Axes that are not programmed in a G00 block are not traversed. In positioning, the individual axes traverse independently of each other with the rapid traverse rate specified for each axis. The precise speeds of your machine can be consulted in the documentation of the manufacturer.

Positioning in the run state with 3 simultaneously controllable axes:



Note

As in positioning with G00, the axes traverse independently of each other (not interpolated), each axis reaches its end point at a different time. Hence, one must be very careful in positioning with several axes, so that a tool does not collide with a workpiece of the tool during the positioning.

Linear interpolation (G00)

Here, all programmed axes traverse in rapid traverse with linear interpolation and reach their target positions simultaneously.

4.2.1.3 Linear interpolation (G01)

With G01 the tool travels on paraxial, inclined or straight lines arbitrarily positioned in space. Linear interpolation permits machining of 3D surfaces, grooves, etc.

Format

G01 X... Y... Z... F... ;

In G01, the linear interpolation is executed with the path feed. The axes that are not specified in the block with G01 are not traversed. The linear interpolation is programmed as in the example given above.

Feed F for path axes

The feedrate is specified under the address F. Depending on the default setting in the machine data, the units of measurement specified with the G commands (G93, G94, G95) are either in mm or inch.

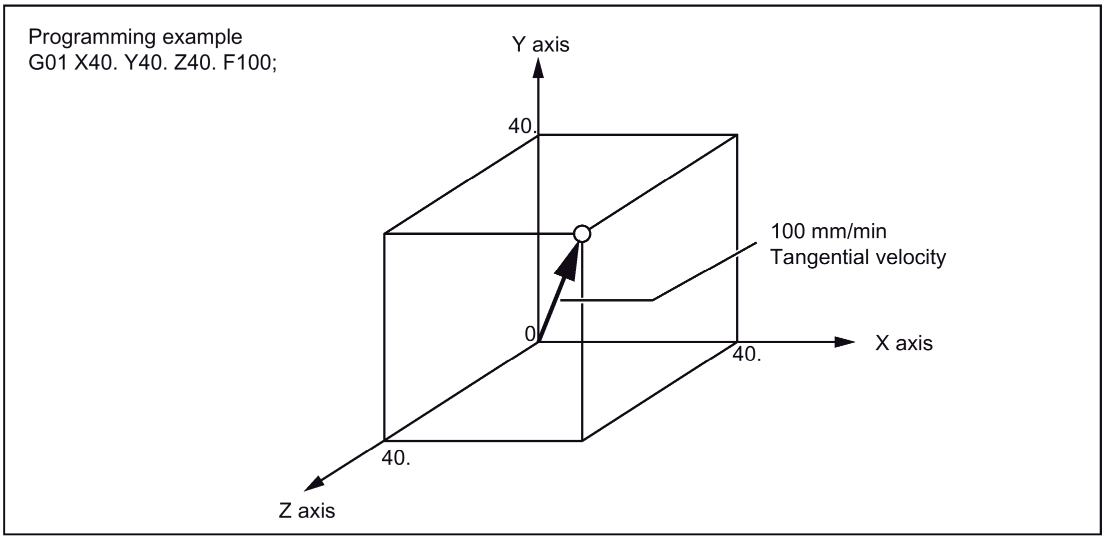
One F value can be programmed per NC block. The unit of feedrate is defined over one of the mentioned G commands. The feed F acts only on path axes and remains active until a new feed value is programmed. Separators are permitted after address F.

Note

An alarm is triggered while executing a G01 block if no feed was programmed in a block with G01 or in the previous blocks.

The end point can be specified either as absolute or as incremental. For more information, refer to Section "Absolute/incremental dimensioning (G90, G91) (Page 106)".

Linear interpolation:



4.2.1.4 Circular interpolation (G02, G03)

Format

To start the circular interpolation, please execute the commands specified in the following table.

Element	Command	Description
Designation of the plane	G17	Circular arc in Plane X-Y
	G18	Circular arc in Plane Z-X
	G19	Circular arc in Plane Y-Z
Direction of rotation	G02	clockwise
	G03	counterclockwise
End-point position	Two axes from X, Y or Z	End-point position in a workpiece coordinate system
	Two axes from X, Y or Z	Distance of start point - end point with sign
Distance between start point - center	Two axes from I, J or K	Distance start point - circle center with sign
Radius of circular arc	R	Radius of circular arc
Feed	F	Speed along the circular arc

Designation of the plane

With the commands specified below, a tool traverses along the specified circular arc in the plane X-Y, Z-X or Y-Z, so that the feed specified with "F" is maintained on the circular arc.

- in Plane X-Y:
  G17 G02 (or G03) X... Y... R... (or I... J...) F... ;
- in Plane Z-X:
  G18 G02 (or G03) Z... X... R... (or K... I...) F... ;
- in the Plane Y-Z:
  G19 G02 (or G03) Y... Z... R... (or J... K...) F... ;

Before the circle radius programming (with G02, G03), one must first select the desired interpolation plane with G17, G18 or G19. Circular interpolation is not allowed for the 4th and 5th axes, if these are linear axes.

Plane selection is also used to select the plane in which the tool radius compensation (G41/G42) is performed. The Plane X-Y (G17) is set automatically after activating the control system.

G17	X-Y plane
G18	Z-X plane
G19	Y-Z plane

The working planes should be specified, in general.

Circles can also be created outside the selected working plane. In this case, the axis addresses (specification of circle end positions) determine the circular plane.

Circular interpolation is possible in the Xβ, Zβ or Yβ plane while selecting an optional 5th linear axis, which also contains a 5th axis besides the X-Y, Y-Z and Z-X planes (β=U, V or W)

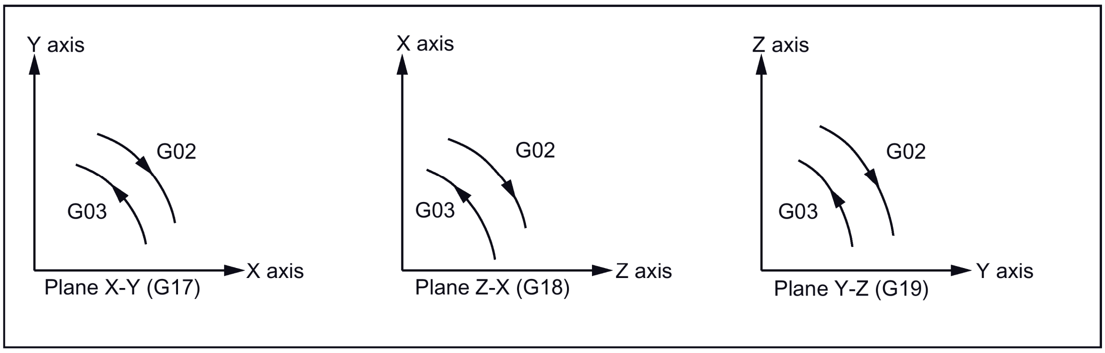
- Circular interpolation in the Xβ plane
  G17 G02 (or G03) X... β... R... (or I... J...) F... ;
- Circular interpolation in the Zβ plane
  G18 G02 (or G03) Z... β... R... (or K... I...) F... ;
- Circular interpolation in the Yβ plane
  G19 G02 (or G03) Y... β... R... (or J... K...) F... ;
- If the address characters for the 4th and 5th axes are omitted - such as in the commands "G17 G02 X... R... (or I... J...) F... ;", then the X-Y plane is selected automatically as the interpolation plane. Circular interpolation with the 4th and 5th axes is not possible if these additional axes are rotary axes.

Direction of rotation

The direction of rotation of the circular arc is to be specified as given in the following figure.

G02	clockwise
G03	counterclockwise

Direction of rotation of the circular arc:



End point

The end point can be specified corresponding to the definition with G90 or G91 as absolute or incremental.

If the specified end point does not lie on the circular arc, the system outputs Alarm 14040 "Error in end point of circle".

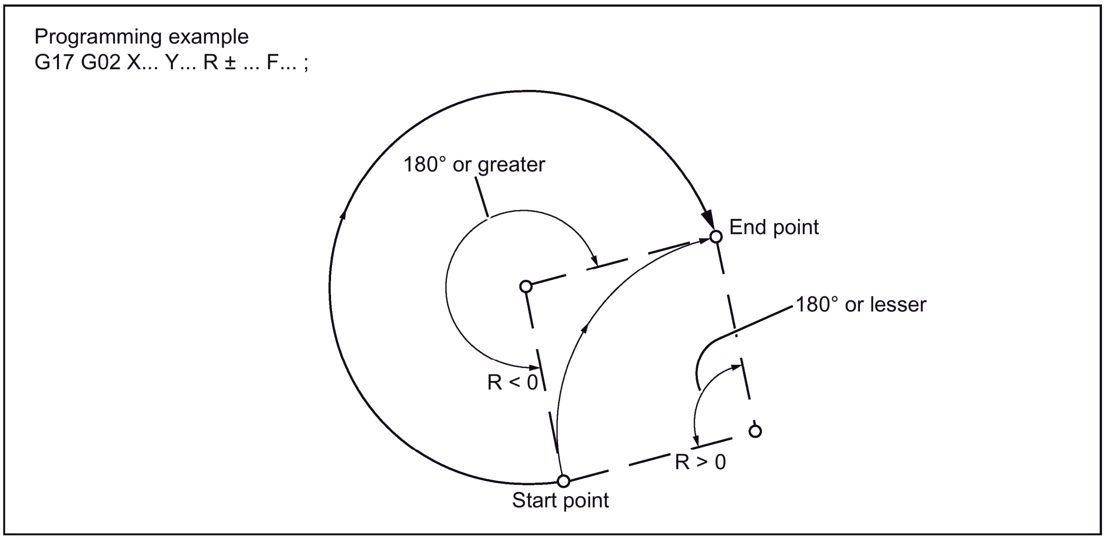
Possibilities of programming circular movements

The control system offers two options of programming circular movements.

The circular motion is described by the:

- Center point and end point in the absolute or incremental dimension (default)
- Radius and end point in Cartesian coordinates
  For a circular interpolation with a central angle <= 180 degree, the programming should be "R > 0" (positive).
  For a circular interpolation with a central angle > 180 degree, the programming should be "R < 0" (negative).

Circular interpolation with specification of Radius R:



Feed

During the circular interpolation, the feed can be specified exactly as during linear interpolation (see Section "Linear interpolation (G01) (Page 93)").

4.2.1.5 Contour definition programming and addition of chamfers or radiuses

Chamfers or radiuses can be added after each traversing block between linear and circular contours. For example, to grind sharp edges of workpieces.

The following combinations are possible during addition:

- between two straight lines
- between two circular arcs
- between a circular arc and a straight line
- between a straight line and a circular arc

Format

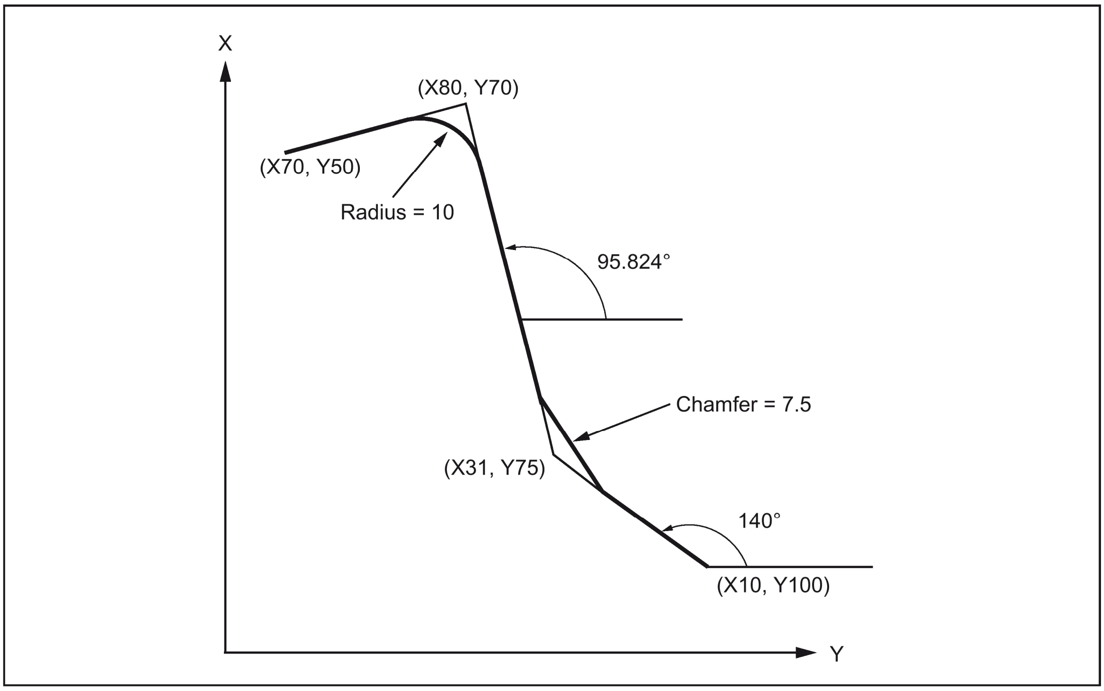
, C...; Chamfer

, R...; Rounding

Example

```
N10 G1 X10. Y100. F1000 G17
N20, A140, C7.5
N30 X80. Y70., A95.824, R10
```

Three straight lines:



ISO dialect mode

In the ISO dialect original, the address C can be used as axis name as well as for denoting a chamfer on the contour.

The address R can either be a cycle parameter or an identifier of the radius of a contour.

The address A is the angle in contour definition.

To differentiate between these two possibilities, a comma "," must be used while programming the contour definition before the address "A", "R" or "C".

Siemens mode

The identifiers of chamfer and radius are defined in Siemens mode using the machine data. Name conflicts can be avoided this way. There should be no comma before the identifier of the radius or chamfer.

Selection of plane

Chamfer or fillet is possible only in the plane specified through the plane selection (G17, G18 or G19). These functions cannot be used on parallel axes.

Note

No chamfer/rounding is inserted, if

- No straight- or circular contour is available in the plane,
- a movement takes place outside the plane,
- The plane is changed or a number of blocks specified in the machine data, that do not contain any information about traversing (e.g., only command outputs), is exceeded.

Coordinate system

After a block that changes the coordinate system (G92 or G52 to G59) or that contains a command of reference point approach (G28 to G30), should not contain any command for chamfering or rounding of corners.

Thread cutting

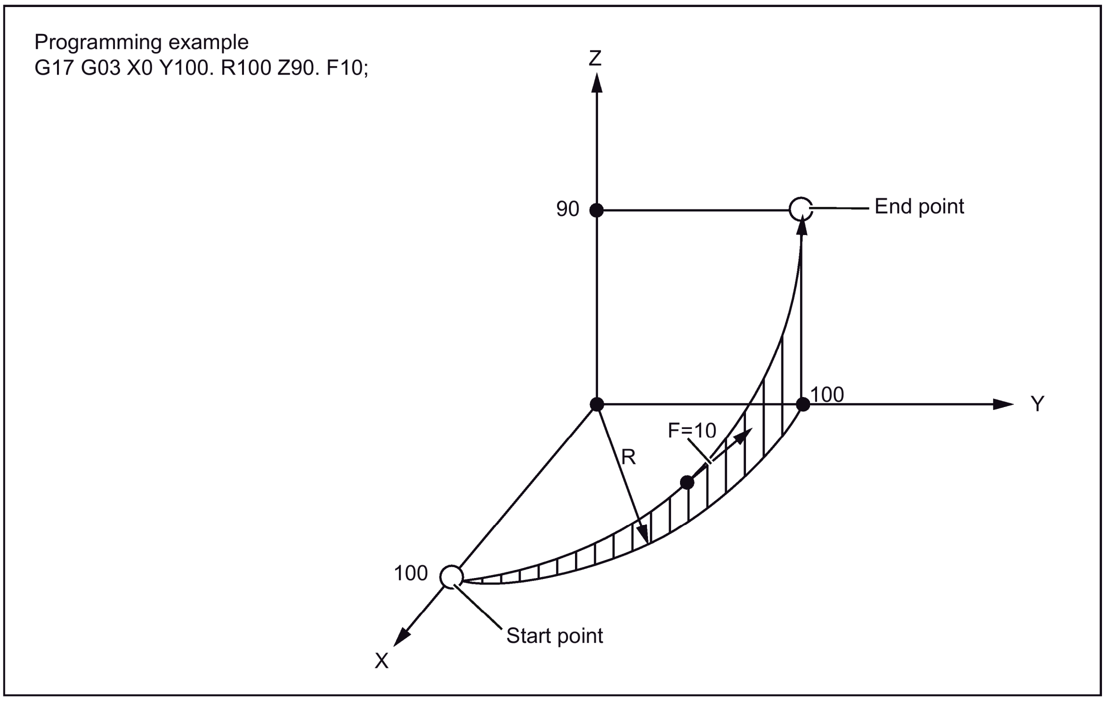
The specification of fillet in thread cutting blocks is not permissible.

4.2.1.6 Helical interpolation (G02, G03)

With helical interpolation, two motions are superimposed and executed in parallel:

- A plane circular motion on which
- A vertical linear motion is superimposed.

Helical interpolation:



Note

G02 and G03 are modal. The circular motion is performed in those axes that are defined by the specification of the working plane.

4.2.2 Reference point approach with G functions

4.2.2.1 Reference point approach with intermediate point (G28)

Format

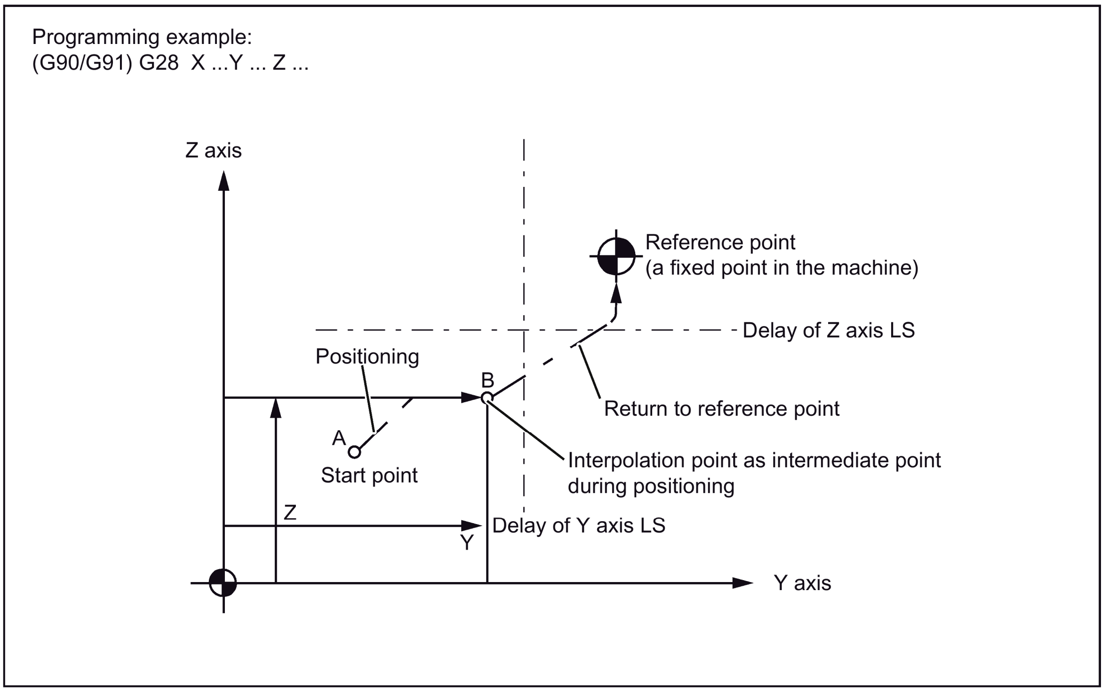
G28 X... Y... Z... ;

The commands "G28 X... Y... Z... ;" can be used to traverse the programmed axes to their reference point. Here, the axes are first traversed to the specified position with rapid traverse, and from there to the reference point automatically. The axes not programmed in the block with G28 are not traversed to their reference point.

Reference position

When the machine has been powered up (where incremental position measuring systems are used), all the axes must approach their reference mark. Only then can traversing movements be programmed. The reference point can be approached in the NC program with G28. The reference point coordinates are defined with the machine data 34100 $_MA_REFP_SET_POS[0] up to [3]). A total of four reference positions can be defined.

Automatic reference point approach:



Return to reference point

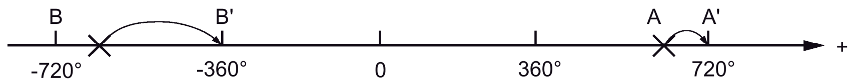
Note

The G28 function is implemented with the shell cycle cycle328.spf. A transformation must not be programmed for an axis which is to approach the reference point with G28 which must approach the reference mark. The transformation is deactivated in cycle328.spf.

Automatic reference point approach for rotary axes

Rotary axes can be used for automatic reference point approach exactly as linear axes. The approach direction of the reference traverse is defined with the machine data 34010 MD_$MA_REFP_CAM_DIR_IS_MINUS.

Return to reference point - rotary axes:



(Return to reference point: The positive traversing direction was selected as the direction of return to the reference point.)

Additions to the commands for automatic reference point approach:

Tool radius compensation and defined cycles

G28 should not be used in operation with tool radius compensation (G41, G42) or in a defined cycle!

Note

G28 is used to interrupt the tool radius compensation (G40) with eventual axis traverse movement to the reference point. Hence, tool radius compensation is to be deactivated before G28 is issued.

Tool offset in G28

In G28, the interpolation point is approached with the current tool offset. The tool offset is deselected when the reference point is finally approached.

4.2.2.2 Checking the reference position (G27)

Format

G27 X... Y... Z... ;

This function is used to check whether the axes are on their reference point.

Test procedure

If the check with G27 is successful, the processing is continued with the next part program block. If one of the axes programmed with G27 is not on the reference point, Alarm 61816 "Axes not on reference point" is triggered and "AUTO" mode is interrupted.

Note

Function G27 is implemented with the cycle 328.spf as with G28.

To avoid a positioning error, the function "mirroring" should be deselected before executing G27.

4.2.2.3 Reference point approach with reference point selection (G30)

Format

G30 Pn X... Y... Z... ;

For the commands "G30 Pn X... Y... Z;", the axes are positioned on the specified intermediate point in the continuous-path mode, and finally traversed to the reference point selected with P2 - P4. With "G30 P3 X30. Y50.;", The X- and Y-axes return to the third reference point. The second reference point is selected on omitting "P". Axes that are not programmed in a G30 block are also not traversed.

Reference point positions

The positions of all the reference points are always determined in relation to the first reference point. The distance of the first reference point from all subsequent reference points is set in the following machine data:

Element	MD
2. Reference point	$_MA_REFP_SET_POS[1]
3. Reference point	$_MA_REFP_SET_POS[2]
4. Reference point	$_MA_REFP_SET_POS[3]

Note

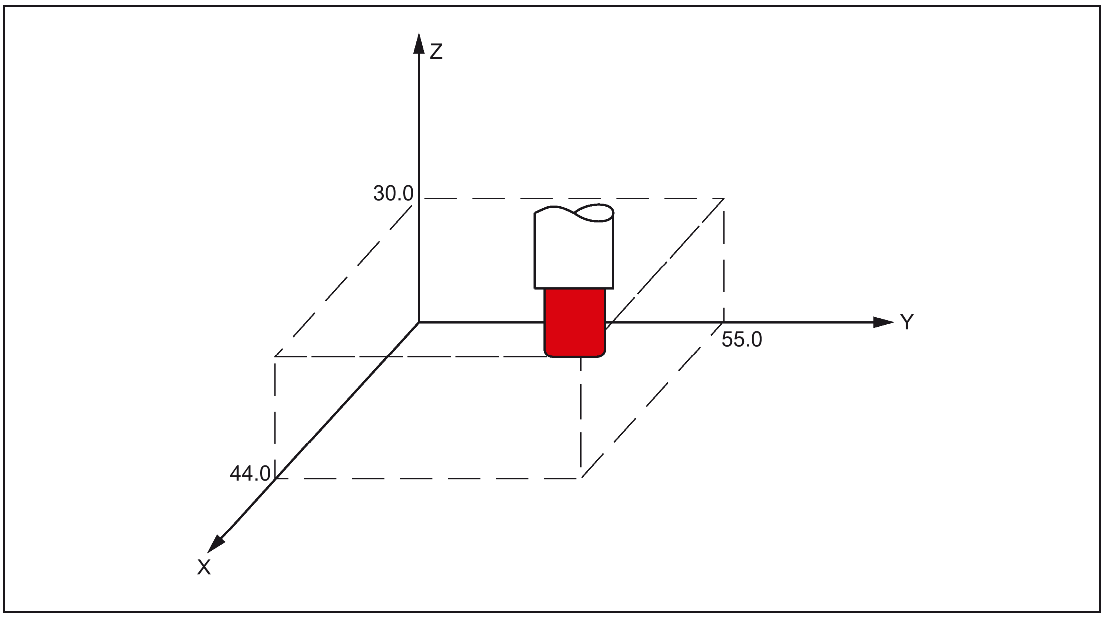
Further details of the points that were considered in the programming of G30 are available in the Chapter "Reference point approach with intermediate point (G28)". Function G30 is implemented with the cycle 328.spf as with G28.

4.3 Motion commands

4.3.1 The coordinate system

The position of a tool is defined uniquely by its coordinates in the coordinate system. These coordinates are defined through axis positions. If, for instance, the three involved Axes are denoted by X, Y and Z, the coordinates are specified as follows:

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



The following coordinate systems are used to specify the coordinates:

- Machine coordinate systems (G53)
- Workpiece coordinate system (G92)
- Local coordinate system (G52)

4.3.1.1 Machine coordinate systems (G53)

Defining the machine coordinate system

The machine zero defines the machine coordinate system MCS. All other reference points refer to the machine zero.

The machine zero is a fixed point on the machine tool which can be referenced by all (derived) measuring systems.

Format

(G90) G53 X... Y... Z... ;

X, Y, Z: absolute dimension word

Selection of machine coordinate system (G53)

G53 suppresses the programmable and adjustable work offset. Traversing in the machine coordinate system on the basis of G53 are always programmed if the tool is to traverse to a machine-specific position.

Compensation deselection

If MD10760 $MN_G53_TOOLCORR = 0, then the active tool length and tool radius compensation remains active in a block with G53

If MD10760 $MN_G53_TOOLCORR = 1, then the active tool length and tool radius compensations in a block are suppressed with G53.

4.3.1.2 Workpiece coordinate system (G92)

Before machining, you must create a coordinate system for the workpiece, the so-called work piece coordinate system. This section describes different methods of setting, selecting and changing a workpiece coordinate system.

Setting a tool coordinate system

The following two methods can be used to set a tool coordinate system:

- With G92 in the part program
- manually through the HMI operator panel

Format

(G90) G92 X... Y... Z... ;

The base point traverses to the specified position on outputting an absolute command. The difference between tool tips and the base point is compensated through the tool length compensation; this way the tool tip can traverse to the target position in any case.

4.3.1.3 Resetting the tool coordinate system (G92.1)

With G92.1, one can reset a shifted coordinate system before the shift. The tool coordinate system is reset to the coordinate system that is defined by the active adjustable work offsets (G54-G59). The tool coordinate system is set to the reference position if no adjustable work offset is active. G92.1 resets shifts carried out through G92 or G52. However, only the axes that are programmed, are reset.

Example 1:

```
N10 G0 X100 Y100          ;Display: WCS: X100 Y100          MCS: X100 Y100
N20 G92 X10 Y10           ;Display: WCS: X10 Y10            MCS: X100 Y100
N30 G0 X50 Y50            ;Display: WCS: X50 Y50            MCS: X140 Y140
N40 G92.1 X0 Y0           ;Display: WCS: X140 Y140          MCS: X140 Y140
```

Example 2:

```
N10 G10 L2 P1 X10 Y10
N20 G0 X100 Y100          ;Display: WCS: X100 Y100          MCS: X100 Y100
N30 G54 X100 Y100         ;Display: WCS: X100 Y100          MCS: X110 Y110
N40 G92 X50 Y50           ;Display: WCS: X50 Y50            MCS: X110 Y110
N50 G0 X100 Y100          ;Display: WCS: X100 Y100          MCS: X160 Y160
N60 G92.1 X0 Y0           ;Display: WCS: X150 Y150          MCS: X160 Y160
```

4.3.1.4 Selection of a workpiece coordinate system (G54 to G59)

As mentioned above, you can select one of the already set workpiece coordinate systems.

- G92

  Absolute commands function in connection with a workpiece coordinate system only if a workpiece coordinate system was selected earlier.

- Selection of a workpiece coordinate system from a selection of specified workpiece coordinate systems via the HMI operator panel

  A workpiece coordinate system can be selected by specifying a G function in the area G54 to G59.

  Workpiece coordinate systems are setup after the reference point approach after Power On. The closed position of the coordinate system is set in MD20154[13].

4.3.1.5 Writing work offset/tool offsets (G10)

The workpiece coordinate systems defined through G54 to G59 or G54 P{1 ... 93} can be changed with the following two processes.

- Data inputting at HMI operator panel
- with the program commands G10 or G92 (setting actual value)

Format

Modified by G10:

G10 L2 Pp X... Y... Z... ;

p=0:	External workpiece work offset
p=1 to 6:	The value of the workpiece work offset corresponds to the workpiece coordinate system G54 to G59 (1 = G54 to 6 = G59)
X, Y, Z:	Workpiece work offset for each axis during an absolute command (G90). Value that must be added during an incremental command (G91) for each axis to the specified workpiece work offset.

Modified by G92:

G92 X... Y... Z... ;

Explanations

Modified by G10:

G10 can be used to change each workpiece coordinate system individually. If the work offset with G10 is to be written only when the G10 block is executed on the machine (main run block), then MD20734 $MC_EXTERN_FUNCTION_MASK, Bit 13 must be set. An internal STOPRE is executed in that case with G10. The machine data bits affect all G10 commands in the ISO Dialect T and ISO Dialect M.

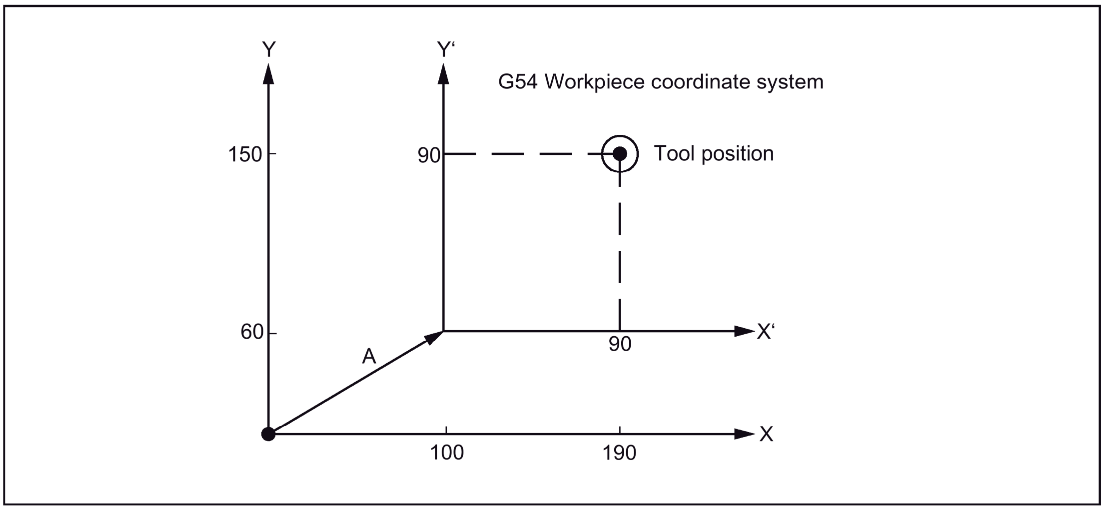
Modified by G92:

By specifying G92 X... Y... Z..., a workpiece coordinate system that was selected earlier with a G command G54 to G59 or G54 P{1 ...93}, can be shifted and thus a new workpiece coordinate system can be set. If X, Y and Z are programmed incrementally, the workpiece coordinate system is defined in such a way that the current tool position matches the total of the specified incremental value and the coordinates of the previous tool position (shift of coordinate system). Finally, the value of the coordinate system shift is added to each individual value of the workpiece work offset. To put it another way: All workpiece coordinate systems are shifted systematically by the same value.

Example

The tool in operation with G54 is positioned at (190, 150), and the workpiece coordinate system 1 (X' - Y') is created each time in G92X90Y90 with a shift of Vector A.

Example of setting coordinates:



4.3.1.6 Local coordinate system (G52)

For programming simplification, a type of workpiece coordinate system can be setup to create a program in the workpiece coordinate system. This part coordination system is also called local coordinate system.

Format

G52 X... Y... Z... ; Setting the local coordinate system

G52 X0 Y0 Z0 ; Deselection of the local coordinate system

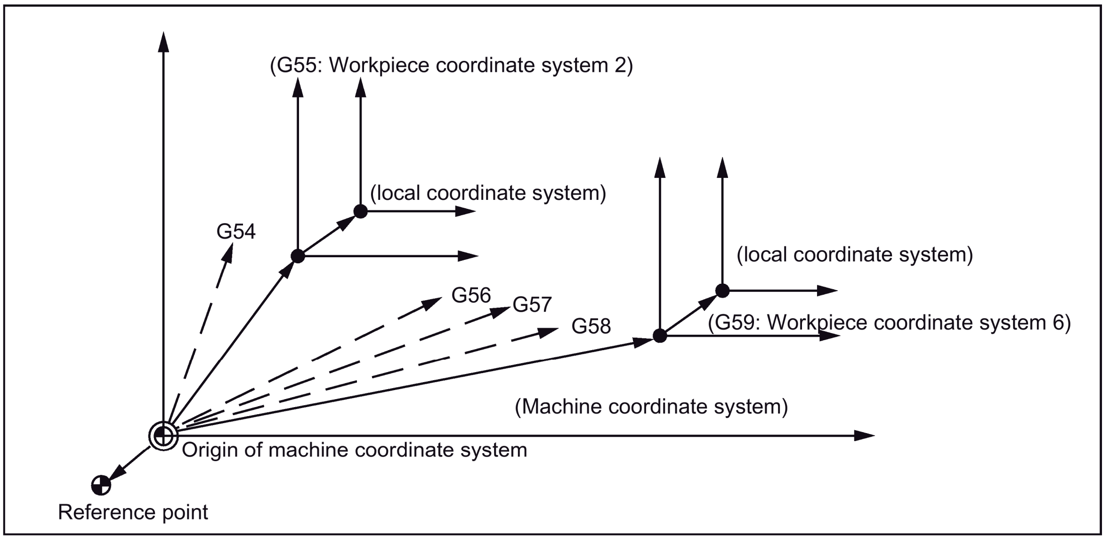
X, Y, Z: Origin of the local coordinate system

Explanations

G52 can be used to program work offsets for all path and positioning axes in the direction of the specified axis. In this way one can work with changing zero points, e.g. during repetitive machining operations at different workpiece positions.

G52 X... Y... Z... is a work offset around the offset values programmed in the relevant specified axis directions. The last specified adjustable work offset (G54 to G59, G54 P1 - P93) serves as reference.

Setting the local coordinate system:



4.3.1.7 Selection of the plane (G17, G18, G19)

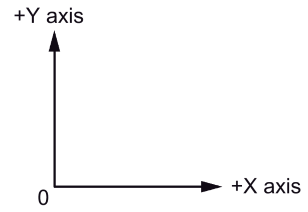
The selection of the plane in which the circular interpolation, tool radius compensation and rotation of the coordinate system took place is done by specifying the following G functions.

G function	Function	G group
G17	X-Y plane	02
G18	Z-X plane	02
G19	Y-Z plane	02

The plane is defined as described below (with a help of the example of Plane X-Y):

The horizontal axis in the first quadrant is the Axis +X, and the vertical axis in the same quadrant is Y+.

Selection of plane:



- The Plane X-Y (G17) is selected automatically after activating the control system.
- The command for moving an individual axis can be specified independently of the plane selection by G17, G18 or G19. Thus for instance, the Z axis can be shifted by specifying "G17 Z ....;".
- The plane in which the tool radius compensation is executed with G41 or G42 is defined by specifying G17, G18 or G19.

4.3.1.8 Parallel axes (G17, G18, G19)

An axis located parallel to one of the three main axes of the coordinate system can be activated by using the function G17 (G18, G19) <Axis name>.

The three main axes are, e.g., X, Y and Z.

Example

G17 U0 Y0

The parallel axis U is activated when the X axis in the G17 plane is replaced.

Explanations

- An associated parallel axis can be defined for each geometry axis with machine data $MC_EXTERN_PARALLEL_GEOAX[ ].
- Only geometry axes from a plane defined with (G17, G18, G19) can be replaced.
- On replacing the axes, normally all shifts (frames) - with the exception of the handwheel and external shifts - are deleted. The following machine data is to be set to prevent the values from being deleted:

  Shifts (frames)
  $MN_FRAME_GEOAX_CHANGE_MODE
- Details are available in the machine data description.
- Alarm 12726 "Impermissible plane selection with parallel axes" is output if a main axis is programmed along with the associated parallel axis with a command for selecting the plane.

4.3.1.9 Rotation of the coordinate system (G68, G69)

Properties of G68 and G69

A coordinate system can be rotated through the following G functions.

G function	Function	G group
G68	Rotation of the coordinate system	16
G69	Deselection of Rotation of the coordinate system	16

G68 and G69 are modal G functions of the G group 16. G69 is set automatically on activating the control system and resetting the NC.

The blocks containing G68 and G69 should not contain any other G functions.

The rotation of the coordinate system is called with G68 and deselected with G69.

Format

G68 X_ Y_ R_ ;

X_, Y_ :

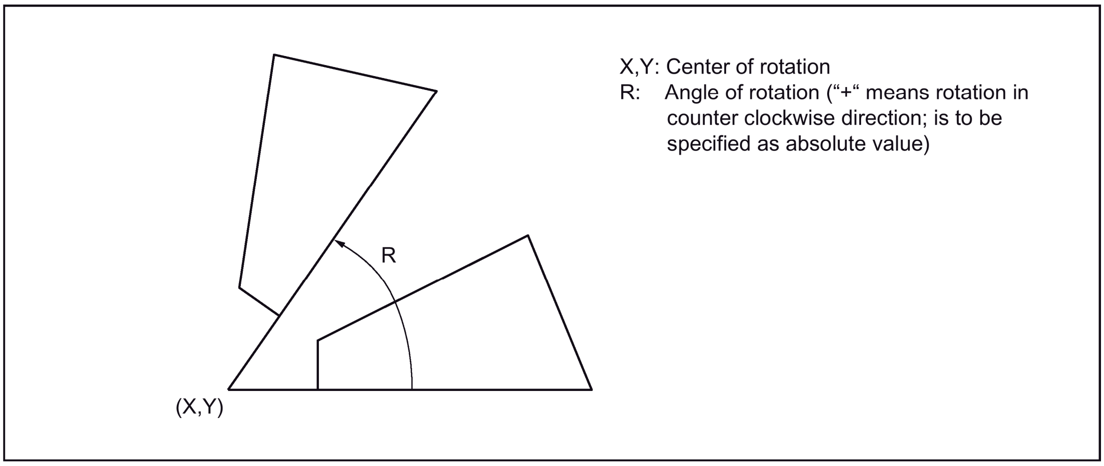
Absolute coordinate values of the rotation center. The actual position is accepted as the rotation center if these are omitted.

R_ :

Angle of rotation as a function of G90/G91 absolute or incremental. If R is not specified, the value of the channel-specific setting from the setting data 42150 $SC_DEFAULT_ROT_FACTOR_R is used as angle of rotation.

- By specifying G17 (or G18, G19) G68 X... Y... R... ; " the commands specified in the following blocks are rotated by the angle specified with R around the point (X, Y). The angle of rotation can be specified in units of 0.001 degree.

  Rotation of a coordinate system:



- The deselection of the coordinate system rotation takes place through G69.
- G68 is executed in the plane that was selected through G68. The 4th and 5th axes must be linear axes.

  G17: X-Y plane

  G18: Z-X plane

  G19: Y-Z plane

Additions to the commands for rotating the coordinate systems

- If "X" and "Y" are omitted, the current position is used as the rotation center for the coordinate rotation.
- The positional data for the rotation of a coordinate system are specified in the rotated coordinate system.
- If you program a change of plane (G17 to G19) after a rotation, the angles of rotation programmed for the axes are retained and continue to apply in the new working plane. It is therefore advisable to deactivate the rotation before a change of plane.

4.3.1.10 3D rotation (G68/G69)

G code G68 is extended for 3D rotation.

G68 must be programmed in a single block and the blocks containing G68 and G69 should not contain any other G functions.

Format

G68 X.. Y.. Z.. I.. J.. K.. R..

X.. Y.. Z..:	Coordinates of the pivot point related to the current workpiece zero. If no coordinate is programmed, the pivot point lies in the workpiece zero. The value is always interpreted as absolute. The coordinates of the pivot point act as a work offset. G90/G91 in the block does not affect the G68 command.
I.. J.. K..:	Vector in pivot point. The coordinate system is rotated around this vector at angle R.
R..:	Angle of rotation. The angle of rotation is always absolute.

The 2D or 3D rotation differentiation takes place only through the programming of the vector I, J, K. If there is no vector in the block, G68 2DRot is selected. If there is a vector in the block, G68 3DRot is selected. In the cases of both 2D rotation and 3D rotation, if no angle is programmed, the angle from the setting data 42150 $SC_DEFAULT_ROT_FACTOR_R is active.

If a vector is programmed with the length 0 (I0, Y0, K0), the Alarm 12560 "Programmed value outside the permissible limits" is triggered.

Two rotations can be switched one after the other with G68. If so far no G68 is active in a block containing G68, the rotation is written to the second ISO system frame. If G68 is already active, the rotation is written to the third ISO system frame. Thus, both rotations follow one another.

The 3D rotation is ended with G69. If two rotations are active, both are deselected with G69. G69 must not be alone in the block.

4.3.2 Defining the input modes of the coordinate values

4.3.2.1 Absolute/incremental dimensioning (G90, G91)

Whether the dimensions after an axis address should be absolute or relative (incremental) is specified with these G commands.

Properties of G90, G91

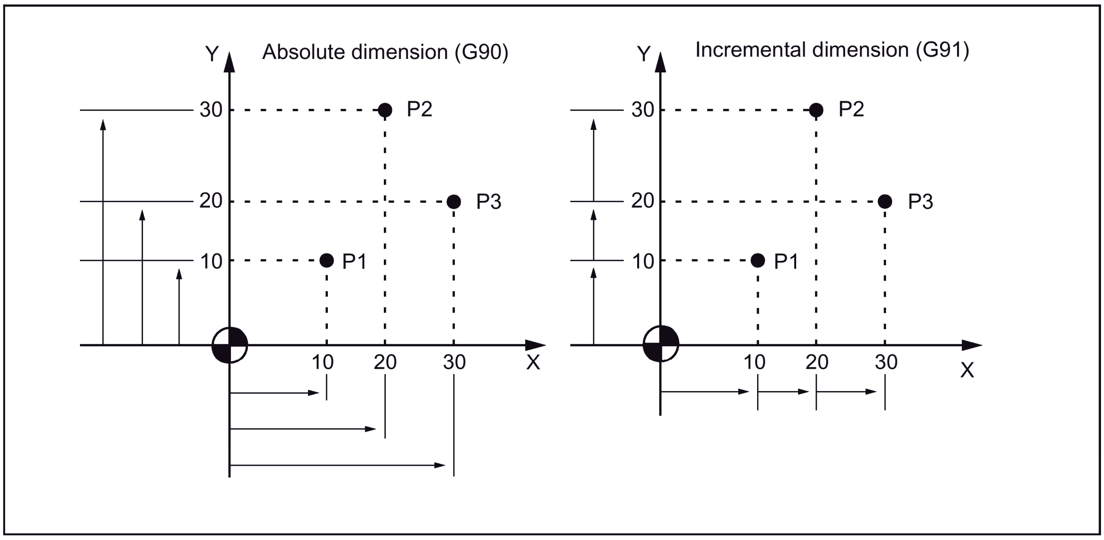
G command	Function	G group
G90	Absolute dimensioning	03
G91	Incremental dimensioning	03

- G90 and G91 are modal G functions of the G group 03. If G90 and G91 are programmed in the same block, the last G function in the block is effective.
- The closed position of G90 or G91 is set in machine data MD20154 $MC_EXTERN_GCODE_RESET_VALUES[2].

Format

- The programmed values are interpreted as absolute axis positions for all axis positions programmed according to G90, e.g. X, Y, Z.
- The programmed values are interpreted as incremental axis positions for all axis positions programmed according to G91, e.g. X, Y, Z.

Absolute and incremental dimensioning (G90, G91):



4.3.2.2 Inch/metric input (G20, G21)

Workpiece-related axes can be programmed in metric or inch dimensions alternately, depending on the dimensioning in the production drawing. The input unit is selected with the following G functions.

G command	Function	G group
G20	Input in "inch"	06
G21	Input in "mm"	06

Format

G20 and G21 are always to be programmed at the start of the block and should not exist along with other commands in a block. The following values are processed in the selected unit of measurement while executing the G function for selecting the unit of measurement: All the following programs, offset values, certain parameters as well as certain manual operation and readout data.

Programming example:

```
G291;
G20;          ———————  Defining the input format "inch"
.
.
.
```

Additions to the commands for defining the unit of measurement

- The closed position is defined via the machine data MD20154 $MC_EXTERN_GCODE_RESET_VALUES[5].
- During changeover, the values of the work offsets are changed completely.
- If the unit of measurement is changed over during program execution, the following must be executed in advance:
  While using a workpiece coordinate system (G54 to G59), this is to be traced back to the basic coordinate system.
  All tool offsets are to be deactivated (G41 to G44 and G49).
- The following is to be done after switching the measuring system from G20 to G21:
  G92 must be executed before specifying the traversing commands for the axes (to setup the coordinate system).
- G20 and G21 are not used to switch the hand wheel- and incremental weighting. This takes place through the PLC program. The machine data responsible for this is called $MA_JOG_INCR_WEIGHT.

4.3.2.3 Scaling (G50, G51)

Properties of G50, G51

The form defined by a part program can be enlarged or reduced according to the required scale. The desired scaling can be selected and deselected via the following functions.

G command	Function	G group
G50	Scaling OFF	11
G51	Scaling ON	11

The selection for scaling and mirroring takes place with G51. A distinction is made between two options in scaling:

- Axial scaling with the parameters I, J, K
  If I, J, K is not programmed in the G51 block, the relevant default value from the setting data 43120 $A_DEFAULT_SCALE_FACTOR_AXIS is effective.
  Negative axial scaling factors lead additionally to mirroring.
- Scaling in all axes with the scaling factor P
  If P is not written in the block G51, the default value from the setting data is effective.
  Negative P values are not possible.

Format

There are two different types of scaling.

Scaling along all axes with the same scaling factor

G51 X... Y... Z... P... ; Start scaling

G50; Deselection of scaling

X, Y, Z: Center coordinate value for the scaling (absolute command)

P: Scaling factor

Scaling along each individual axis with different scaling factors

G51 X... Y... Z... I... J... K... ; Start scaling
G50; Deselection of scaling

X, Y, Z: Reference point of scaling (absolute command)
I, J, K: Scaling factor for the X-, Y- and Z-axis

The type of the scaling factor depends on MD22914 $MC_AXES_SCALE_ENABLE.

$MC_AXES_SCALE_ENABLE = 0:
The scaling factor is specified with "P". If "I,J,K" is programmed in this setting, the setting data 42140 $SC_DEFAULT_SCALE_FACTOR_P is used for the scaling factor.

$MC_AXES_SCALE_ENABLE = 1:
The scaling factor is specified with "I,J,K". If only "P" is programmed in this MD setting, the setting data 43120 $SA_DEFAULT_SCALE_FACTOR_AXIS is used for the scaling factors.

Weighting of scaling factors

The scaling factors are multiplied either with 0.001 or 0.00001. The factors are selected with MD22910 $MC_WEIGHTING_FACTOR_FOR_SCALE=0, scaling factor 0.001, $MC_WEIGHTING_FACTOR_FOR_SCALE=1, scaling factor 0.00001.

The workpiece zero is always the reference point for the scaling. A reference point cannot be programmed.

Programmable mirroring (negative scaling)

A mirror image can be created with a negative value of the axial scaling factor.

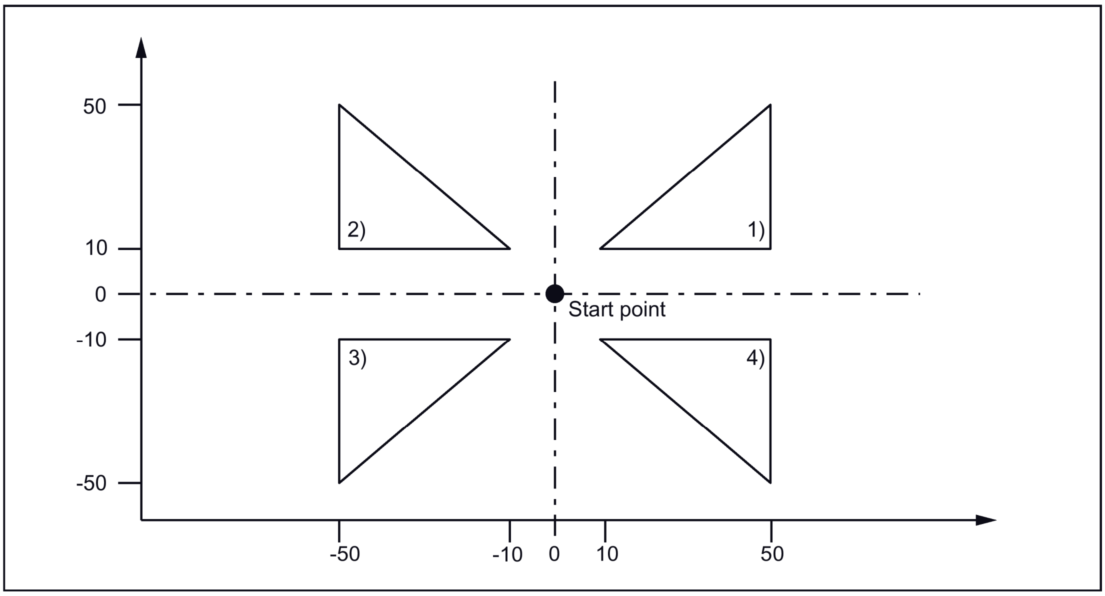
To do this, MD22914 $MC_AXES_SCALE_ENABLE = 1 must be active. If I, J or. K is omitted from the blocks with G51, the values preset in the setting data 43120 $SA_DEFAULT_SCALE_FACTOR_AXIS are activated.

Example

```
_N_0512_MPF                           ;(Part program)
N10 G17 G90 G00 X0 Y0                 ;Start position for the approach motion
N30 G90 G01 G94 F6000
N32 M98 P0513                         ;1) Contour programmed as in the subroutine
N34 G51 X0. Y0. I-1000 J1000          ;2) Contour, mirrored on X
N36 M98 P0513
N38 G51 X0. Y0. I-1000 J-1000         ;3) Contour, mirrored on X and Y
N40 M98 P0513
N42 G51 X0. Y0. I1000 J-1000          ;4) Contour, mirrored on Y
N44 M98 P0513
N46 G50                               ;Deselection of scaling and mirroring
N50 G00 X0 Y0
N60 M30

_N_0513_MPF                           ;(Subroutine of 00512)
N10 G90 X10. Y10.
N20 X50
N30 Y50
N40 X10. Y10.
N50 M99
```

Scaling for each axis and programmable mirroring:



Tool offset

This scaling is not valid for cutter radius compensations, tool length compensations and tool offset values.

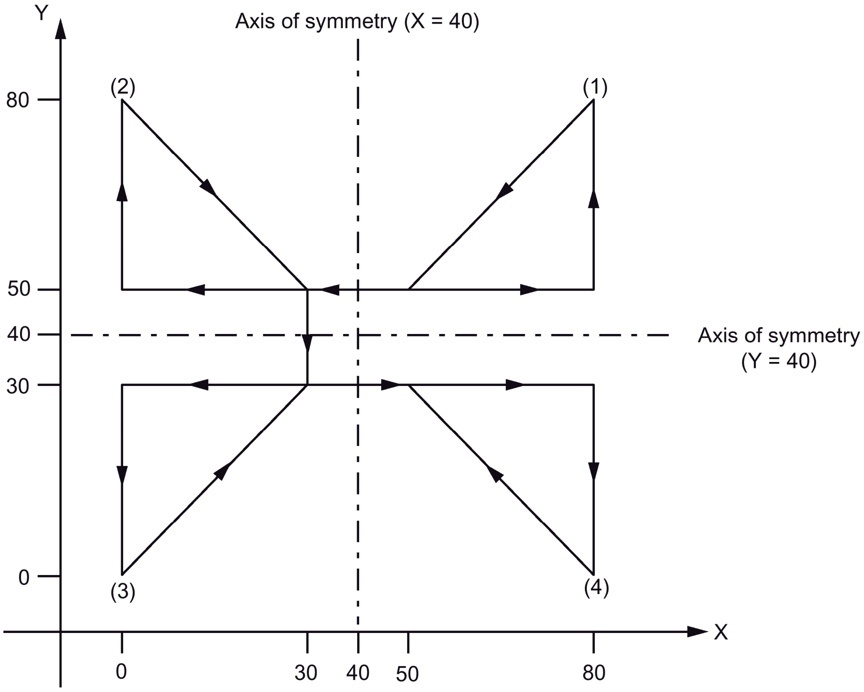
Commands for reference point approach and for changing the coordinate system

The G27, G28 and G30 functions as well as commands related to the coordinate system (G52 to G59, G92), should not be used when scaling is active.

4.3.2.4 Programmable mirroring (G50.1, G51.1)

G51.1 can be used to mirror workpiece shapes on coordinate axes. All programmed traversing movements are then executed as mirrored.

Programmable Mirroring:



(1) Programmed contour original
(2) Symmetrical mapping with reference to the line that runs parallel to Y and intersects X at 40
(3) Symmetrical mapping with reference to the Point (40,40)
(4) Symmetrical mapping with reference to the line that runs parallel to X and intersects Y at 40

Format

X, Y, Z: Positions and mirroring axis

G51.1: Command for activating the mirroring

Mirroring takes place on a mirroring axis which is parallel to X, Y or Z and whose position is programmed with X, Y or Z. G51.1 X0 is used to mirror on the X axis, G51.1 X10 is used to mirror on a mirroring axis that runs 10 mm parallel to the X axis.

Example

```
N1000 G51.1 X... Y... Z...         ; Activate mirroring
...                                ; All the axis positions mirrored in the following blocks are mir-
                                   rored at the mirroring axis programmed in N1000
...                                ;
...                                ;
...                                ;
G50.1 X... Y... Z..                ; Deselection of programmable mirroring
N32 M98 P0513                      ; 1) Contour programmed as in the subroutine
```

Mirroring with reference to a single axis in a specified plane

The following commands can change if the mirroring is used on one of the axes in the specified plane as described below:

Command	Explanation
Circular interpolation	G02 and G03 are exchanged mutually
Cutter radius compensation	G41 and G42 are exchanged with each other
Coordinate rotation	The "clockwise" (CW) and "counter-clockwise" (CCW) directions of rotation are exchanged mutually.

Commands for reference point approach and for changing the coordinate system

The G27, G28 and G30 functions as well as commands related to the coordinate system (G52 to G59, G92, etc), should not be used when mirroring is active.

4.3.3 Time-controlled commands (G04)

One can use G04 to interrupt workpiece machining between two NC blocks for a programmed time/number of spindle revolutions, e.g. for backing off.

One can set with MD20734 $MC_EXTERN_FUNCTION_MASK, whether the dwell time for Bit 2 is to be interpreted as time (s or ms) or alternatively as spindle revolutions. If $MC_EXTERN_FUNCTION_MASK, Bit 2=1 is set, the dwell time is interpreted in seconds if G94 is active; it is specified in spindle revolutions (R) if G95 is selected.

Format

G04 X_; or G04 P_;

X_: Time display (commas possible)

P_: Time display (commas not possible)

- The dwell time (G04 ..) must be programmed alone in a block.

If the values of X and U are programmed in the standard notation (without decimal point), they are converted to internal units, depending on IS B, IS C (for input resolution, see Section "Programming the decimal point (Page 9)"). P is always interpreted in internal units.

```
N5 G95 G04 X1000
```

Standard notation: 1000*0.001 = 1 Spindle revolution

Calculator notation: 1000 spindle revolutions

4.3.4 Tool offset functions

4.3.4.1 Tool offset data memory

The Siemens tool data memory must be used, as programs in Siemens Mode and in ISO Direct Mode must run alternately on the control system. Hence, length, geometry and wear exist in each tool offset data memory. In Siemens mode, the offset data memory is addressed with "T" (Tool No.) and "D" (cutting edge no.), abbreviated as T/D No.

In the programs that are written in ISO dialect, the tool offset no. is addressed with "D" (radius) or H (length), denoted hereafter as D/H No.

For unique assignment between D and H numbers or the T/D number, one must add the $TC_DPH[t,d] element to the tool data offset memory. The D/H number is input in ISO dialect in this element.

T	D/cutting edge	ISO_H $TC_DPH	Radius	Length
1	1	10		
1	2	11		
1	3	12		
2	1	13		
2	2	14		
2	3	15		

For an assignment of tool length compensations of the geometry axes that is independent of the plane selection, the setting data $SC_TOOL_LENGTH_CONST must contain the value "17". Length 1 is always assigned to the Z axis in this case.

4.3.4.2 Tool length compensation (G43, G44, G49)

In tool length compensation, the amount of the specified values in the program stored in the tool offset data memory is added to the Z axis or subtracted from it to undertake a offset of the programmed paths according to the length of the cutting tool.

Commands

While executing the tool length compensation, the addition or subtraction of the tool offset data is determined through the used G function and the direction of offset is determined with the H functions.

G functions used for the tool length compensation

The tool length compensation is called with the following G functions.

G function	Function	G group
G43	Addition	08
G44	Subtraction	08
G49	Deselection	08

- G43 and G44 are modal and remain active till they are deselected through G49. The tool length compensation is deselected with G49. H00 can also be used to deselect the tool length compensation.
- By specifying "G43 (or G44) Z... H... ; " the tool offset amount specified with the H function is added to or subtracted from the specified position of the Z axis, and the Z axis then traverses to the corrected target position, i.e., the target position of the Z axis specified in the program is shifted by the magnitude of the tool offset.
- By specifying "(G01) Z... ; G43 (or G44) H... ; " the Z axis traverses the path that corresponds to the tool offset amount specified via the H function.
- By specifying "G43 (or G44) Z...H...H... ; " the Z axis traverses the path that corresponds to the difference between the previous tool offset amount and the new tool offset amount.

H function for specification of the tool offset direction

The direction of tool offset is determined by the sign of the tool length compensation that is activated by the H function, and the programmed G function. See the table below.

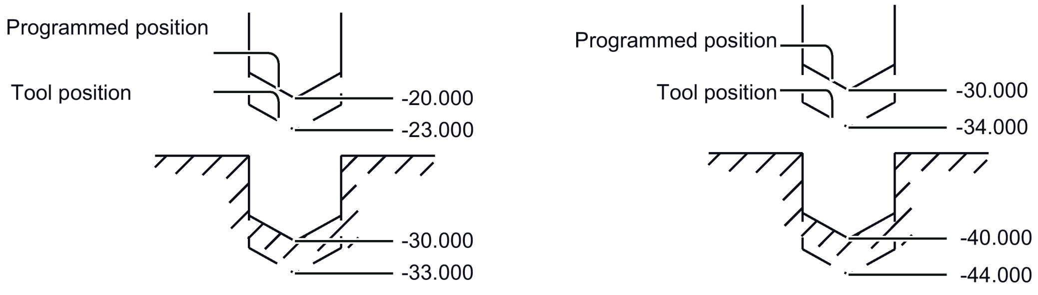
	Signs of tool offset amount (H function)	
	positive	negative
G43	Tool offset in positive direction	Tool offset in negative direction
G44	Tool offset in negative direction	Tool offset in positive direction

Tool position offset:

Programming example
H10 .... Offset -3.0
H11 .... Offset 4.0

Position data indicator including offset (Z axis only)

```
N101  G92 Z0; 0.000
N102  G90 G00 X1.0 Y2.0; 0.000
N103  G43 Z-20. H10; -23.000
N104  G0I Z-30. FI000; -33.000
N105  G00  Z0H00; 0.000
   .   .
   .   .
   .   .
N201 G00 X-2.0 Y-2.0
N202 G44 Z-30 H11; -34.000
N203 G01 Z-40 F1000; -44.000
N204 G00 Z0 H00; 0.000
```



Settings

- The machine data $MC_TOOL_CORR_MOVE_MODE determines whether the tool length compensation is to be undertaken with the selection of the tool offset or only during the programming of an axis motion.

  $MC_CUTTING_EDGE_DEFAULT = 0 defines that initially no tool length compensation is active during a tool change.

  $MC_AUXFU_T_SYNC_TYPE defines whether the output of the T function to the PLC takes place during or after the traversing movement.

  $MC_RESET_MODE_MASK, Bit 6, can be used to define that the currently active tool length compensation will remain active even after a RESET.
- The cutter radius compensation can also be called for an operation with tool length compensation.

Tool length compensation in several axes

Tool length compensation can also be activated for several axes. A display of the resulting tool length compensation is not possible any more in that case.

4.3.4.3 Cutter radius compensation (G40, G41, G42)

In cutter radius compensation, the programmed tool paths are automatically shifted by the radius of the cutting tool used. The path to be corrected (radius of the cutting tool) can be stored in the tool offset data memory using the NC operator panel. The tool offsets can also be overwritten with the G10 command in the part program; G10 cannot be used to create new tools.

The tool offset data in the program is called by specifying the number of the tool offset data memory with a D function.

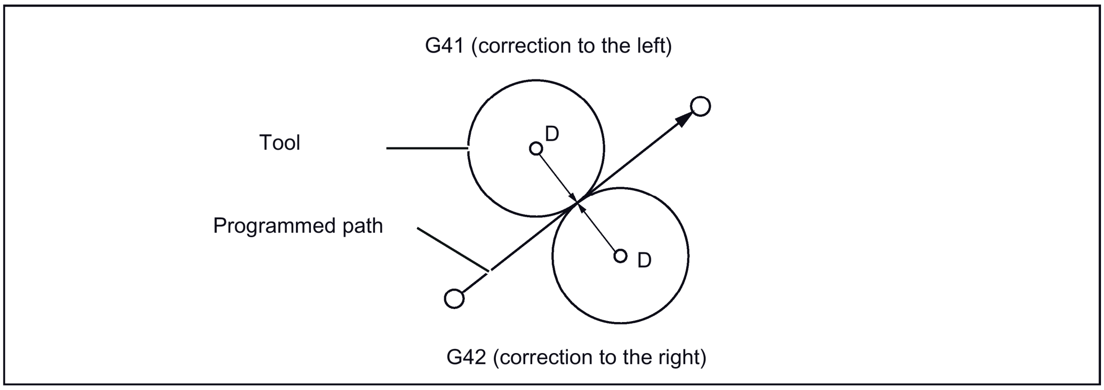
Commands

The cutter radius compensation is called with the following G functions.

G function	Function	G group
G40	Deselection of the tool radius compensation	07
G41	Tool radius compensation (tool works in machining direction to the left of the contour)	07
G42	Tool radius compensation (tool works in machining direction to the right of the contour)	07

The tool radius compensation is called by executing G41 or G42 and deselected through G40. The offset direction is determined through the specified G function (G41, G42) and the offset amount is determined through the D function.

Cutter radius compensation:



- A negative offset value of the tool radius is equivalent to a change of compensation side (G41, G42). The D function must either be programmed in the same block as G41 or G42 or in a previous block. D00 means tool radius = "0".
- The selection of the plane in which the tool radius is active is done with G17, G18 or G19. The G function used to select the plane is to be programmed in the same block as G41 or G42 or in the block before G41 or G42.

G function	Function	G group
G17	Selection of plane X-Y	02
G18	Selection of plane Z-X	02
G19	Selection of plane Y-Z	02

The selected plane should not be changed if the tool offset is selected, otherwise there is an error message.

Activation/deactivation of tool radius compensation

A drive command must be programmed with G0 or G1 if an NC block contains G40, G41 or G42. At least one axis of the selected working plane must be specified in this drive command.

Note

Compensation mode

Compensation mode may only be interrupted by a certain number of consecutive blocks or M functions which do not contain drive commands or positional data in the compensation plane: Standard 3.

Note

Machine manufacturer

The number of successive interruptions blocks or M functions can be set via the machine data 20250 CUTCOM_MAXNUM_DUMMY_BLOCKS (refer to machine manufacturer).

Note

A block with path zero is also taken as interruption!

Changeover between G41 and G42 in operation with cutter radius compensation

The offset direction (left or right) can be changed over directly without having to leave the compensation mode.

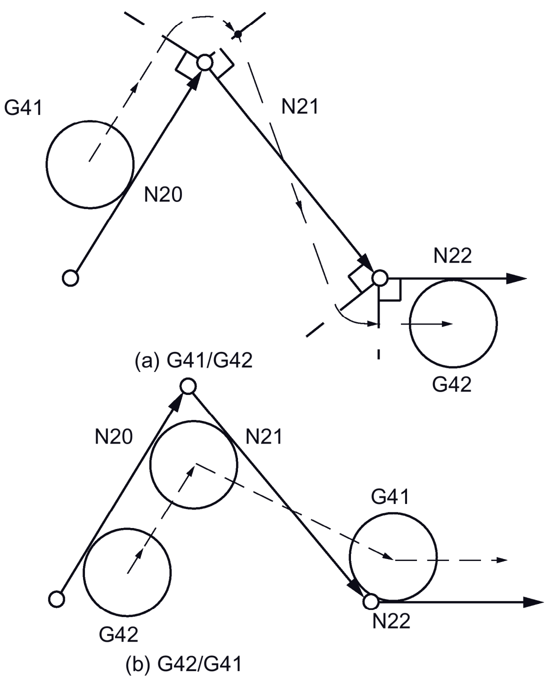
The new offset direction is approached with the next block, through an axis motion.

Changeover of the tool offset direction at block start and end of block:

Programming example

```
N10 G17 G01 F... ;
N11 G41 (G42) X... Y... ;
      .
      .
      .
N20 G01 X... Y... F... ;
N21 G42 (G41) X... Y... ;
N22 X... ;
```

Set for switching the offset direction



Note: If the content of Block N21 is reproduced in two blocks as specified below,
G42 (or G41);
X Y;
the offset direction is also switched over in the same way.

Deselection of the tool offset

There are two methods of deselecting the tool offset, which can be set through the setting data 42494 $SC_CUTCOM_ACT_DEACT_CTRL.

- Method A:

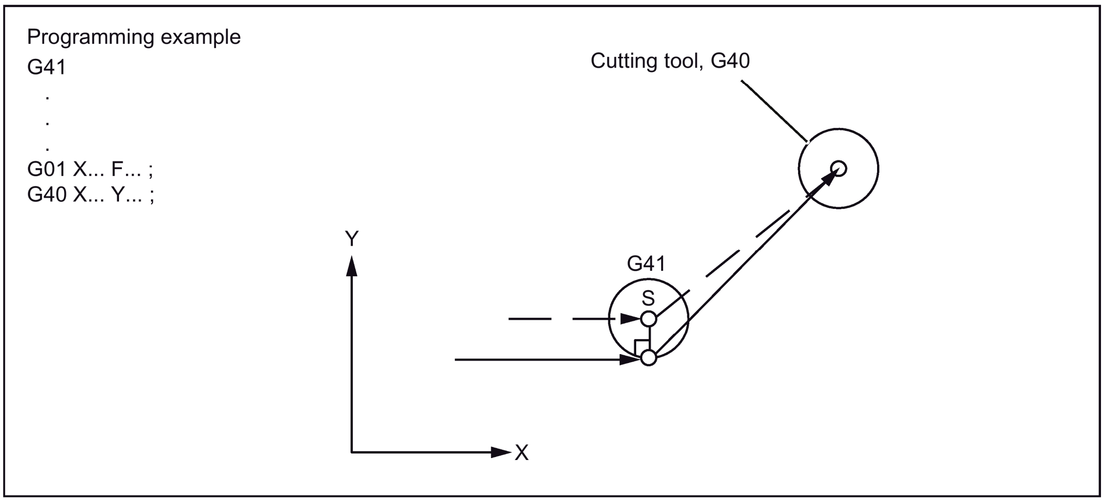
  If G40 is programmed in a block without axis motion, the tool radius compensation is deselected only with the next block through an axis motion.

- Method B:

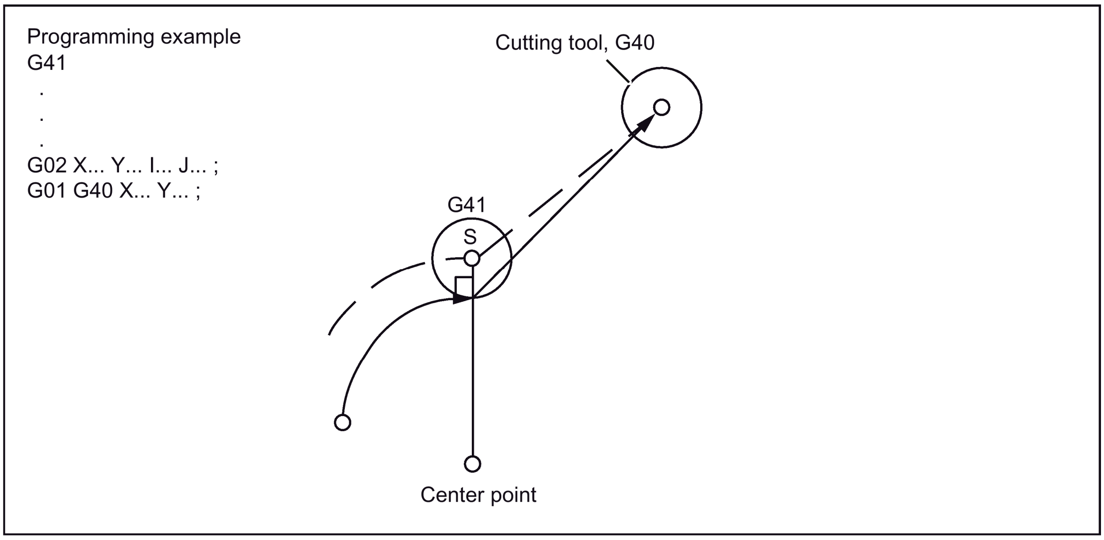
  If G40 is programmed in a block without axis motion, the tool radius compensation is deselected immediately. In other words, that linear interpolation (G00 or G01) must be active in the block, because the tool radius compensation can be deselected only with a linear movement. An alarm is triggered if no linear interpolation is active during the selection of the tool radius compensation.

Deselection of the compensation mode at an internal angle (smaller than 180°):

Straight line - straight line:



Circular arc - straight line:



4.3.4.4 Collision detection

Activation via the NC program

Although the "Collision detection" function is available only in Siemens mode, it can also be used in ISO dialect mode. Activation and deactivation must be undertaken only in Siemens mode.

```
G290        ;Activation of Siemens mode
CDON        ;Activation of the detection of bottlenecks
G291        ;Activation of ISO dialect mode
...
...
G290        ;Activation of Siemens mode
CDOF        ;Deactivation of the detection of bottlenecks
G291        ;Activation of ISO dialect mode
```

Activation by setting machine data

MD20150 $MC_GCODE_RESET_VALUES[22] = 2: CDON (effective modal)

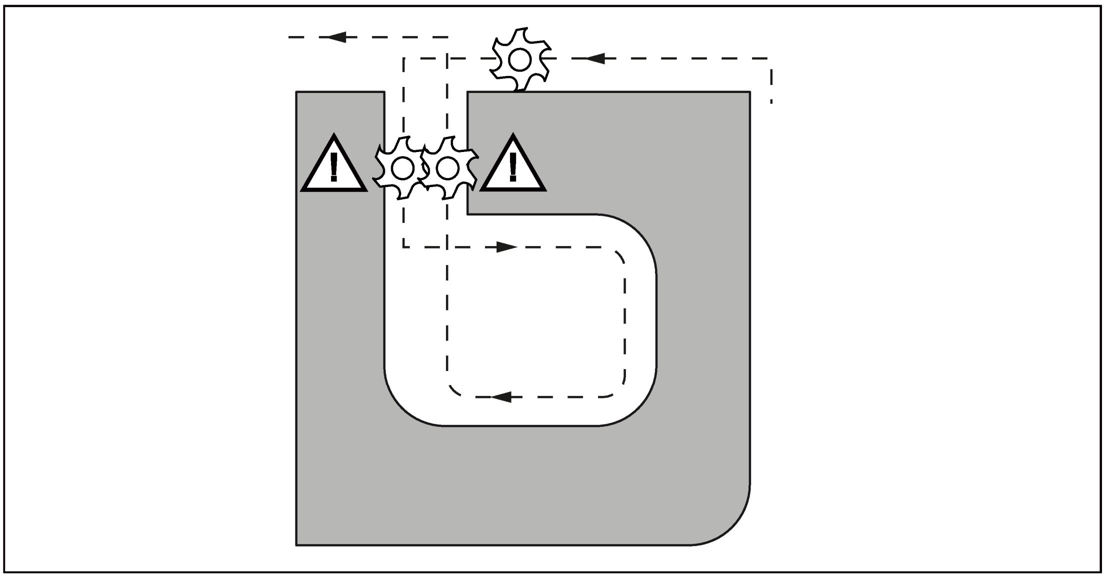
MD20150 $MC_GCODE_RESET_VALUES[22] = 1: CDOF (not effective modal)

Function

With active CDON (Collision Detection ON) and active tool radius compensation, the control system monitors tool paths through look-ahead contour calculation. This Look Ahead function allows possible collisions to be detected in advance and permits the control to actively avoid them.

With deactivated bottleneck detection (CDOF), a search is made in the previous traversing block (at inside corners) for a common point of intersection for the current block; if necessary the search is extended to even earlier blocks. An error message is triggered if no point of intersection is found with this method.

Collision detection:



CDOF can be used to avoid the faulty detection of bottlenecks, resulting, for example, from missing information that is not available in the NC program.

Note

Machine manufacturer

The number of NC blocks that are included in the monitoring can be set via machine data (see machine manufacturer).

Examples

In the following pages, you will find a few examples of critical machining situations that can be detected by the control system and corrected through changes in the tool paths.

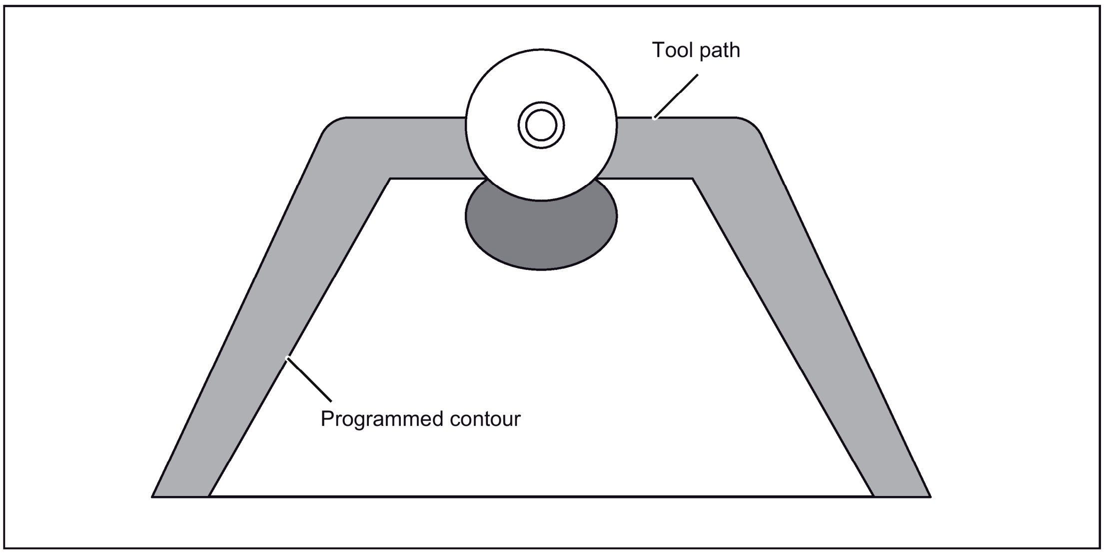
To avoid program interruptions, during program validation only the ones that have the biggest radius from among all tools should be selected.

In each of the following examples, a tool with a too large radius was selected for machining the contour.

Detection of bottlenecks

As the selected tool radius for machining this inside contour is too big, the bottlenecks are bypassed. An alarm is output.

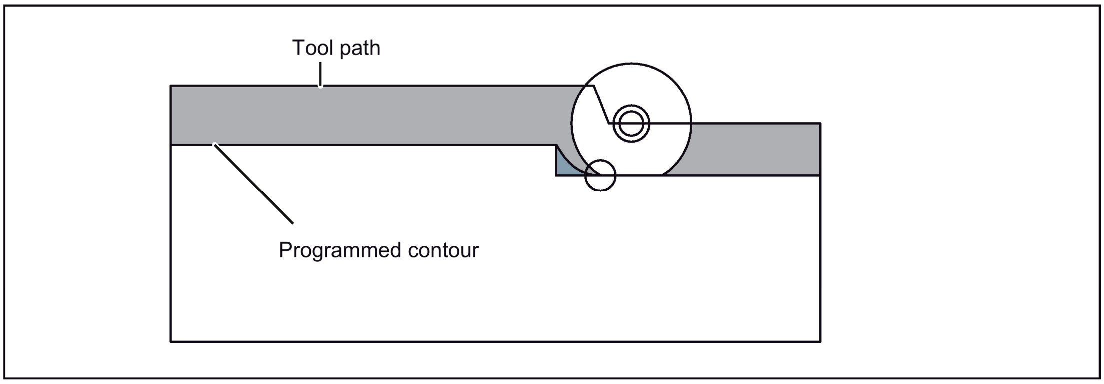
Detection of bottlenecks:



Contour definition shorter than tool radius

The tool traverses the tool angle on a transition circle and then follows exactly the programmed contour.

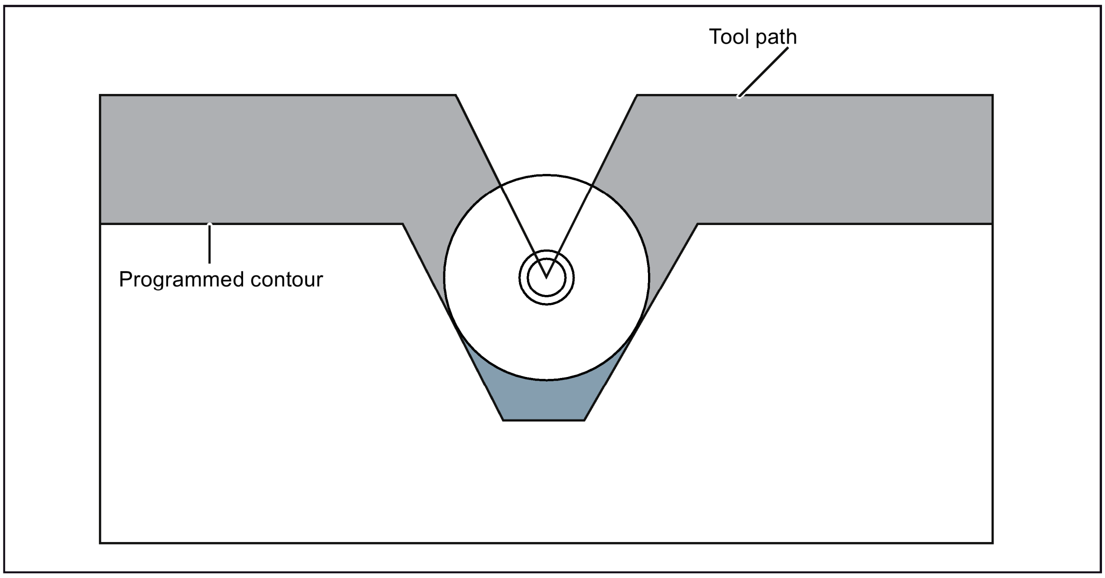
Contour definition shorter than tool radius:



Tool radius too large for internal machining

In such cases, a machining of the contour takes place only to the extent possible without damaging the contour.

Tool radius too large for internal machining:



4.3.5 S-, T-, M- and B functions

4.3.5.1 Spindle function (S function)

The spindle speed is specified in rpm in Address S. The direction of spindle rotation is selected with M3 and M4. M3 = right direction of spindle rotation, M4 = left direction of spindle rotation. The spindle stops with M5. Details are available in the documentation of your machine manufacturer.

- S commands are modal, i.e., they remain active up to the next S command once they are programmed. The S command is maintained if the spindle is stopped with M05. If M03 or M04 is programmed thereafter without specifying an S command, then the spindle starts at the originally programmed speed.
- If the spindle speed is changed, please pay attention to which gear stage is currently set for the spindle. Details are available in the documentation of your machine manufacturer.
- The lower limit for the S command (S0 or an S command near S0) depends on the drive motor and the drive system of the spindle and is different from machine to machine. Negative values are not permitted for S! Details are available in the documentation of your machine manufacturer.

4.3.5.2 Tool function

There are different options of command output for the tool function. Details are available in the documentation of your machine manufacturer.

4.3.5.3 Additional function (M function)

The M functions initiate switching operations, such as "Coolant ON/OFF" and other functions on the machine. Various M functions have already been assigned a fixed functionality by the CNC manufacturer (see the following section).

Programming

M... Possible values: 0 to 9999 9999 (maximum INT value), integer

All free M function numbers can be assigned by the machine manufacturer, e.g. for switching functions to control the clamping devices or for switching on/off of further machine functions. See data of the machine manufacturer.

The NC-specific M functions are described below.

M functions to stop operations (M00, M01, M02, M30)

A program stop is triggered with this M function and the machining is interrupted or ended. Whether the spindle is also stopped depends on the specification of the machine manufacturer. Details are available in the documentation of your machine manufacturer.

M00 (program stop)

The machining is stopped in the NC block with M00. One can now, e.g., remove chips, re-measure, etc. A signal is output to the PLC. The program can be continued with the following key:

M01 (optional stop)

M01 can be set via the following methods:

- HMI/dialog box "Program control" or the
- VDI interface

The program processing of the NC is maintained with M01 only when the corresponding signal of the VDI interface is set or "Program control" was selected in the HMI/dialog box.

M30 or M02 (end of program)

A program is ended with M30 or M02.

Note

A signal is output to the PLC with M00, M01, M02 or M30.

Note

Data on whether spindle is stopped with the commands M00, M01, M02 or M30 or the coolant supply is interrupted is available in the documentation of your machine manufacturer.

4.3.5.4 M functions of spindle control (M19, M29)

M function	Function
M19	Positioning the spindle
M29	Changeover of spindle to the axis/open-loop control mode

The spindle is traversed to the spindle position defined in the setting data 43240 $SA_M19_SPOS[spindle number] with M19. The positioning mode is stored in $SA_M19_SPOS.

The M function number for the changeover of the spindle mode (M29) can also be set over a machine data variable. MD20095 $MC_EXTERN_RIGID_TAPPING_N_NR is used to pre-set the M function number. Only the M function numbers that are not used as standard M functions can be assigned. For example, M0, M5, M30, M98, M99 etc are not allowed.

4.3.5.5 M functions for subroutine calls (M98, M99)

M function	Function
M98	Subprogram call
M99	Subprogram end

In ISO mode, the spindle is switched to the axis mode with M29.

4.3.5.6 Macro call via M function

Via M numbers, one can call a subroutine (macro) similar to G65.

The configuration of a maximum of 10 M functions replacements is undertaken via machine data 10814 $MN_EXTERN_M_NO_MAC_CYCLE and machine data 10815 $MN_EXTERN_M_NO_MAC_CYCLE_NAME.

Programming takes place identical to G65. Repetitions can be programmed with the L address.

Restrictions

Only one M function replacement (or only one subroutine call) can be executed per part program line. Conflicts with other subroutine calls are signaled by alarm 12722. There is no further M function replacement in the replaced subroutine.

Otherwise, the same restrictions are valid as in G65.

Conflicts with pre-defined and other defined M numbers are rejected with an alarm.

Configuration example

Call of subroutine M101_MAKRO via the M101 M function:

$MN_EXTERN_M_NO_MAC_CYCLE[0] = 101

$MN_EXTERN_M_NO_MAC_CYCLE_NAME[0] = "M101_MAKRO"

Call of subroutine M6_MAKRO via the M6 M function:

$MN_EXTERN_M_NO_MAC_CYCLE[1] = 6

$MN_EXTERN_M_NO_MAC_CYCLE_NAME[1] = "M6_MAKRO"

Programming example for tool change with M function:

```
PROC MAIN
...
N10                 M6 X10 V20                          ;Call of M6_MAKRO program
...
N90                 M30
PROC M6_MAKRO
...
N0010               R10 = R10 + 11.11
N0020               IF $C_X_PROG == 1 GOTOF N40         ;($C_X_PROG)
N0030               SETAL(61000)                        ;programmed variable not
                                                        ;transferred correctly
N0040               IF $C_V == 20 GTOF N60              ;($C_V)
N0050               SETAL(61001)
N0060               M17
```

4.3.5.7 M functions (M08, M09)

General M functions

The non-specific M functions are defined by the machine manufacturer. A representative example of the use of general M functions is available under. Details are available in the documentation of your machine manufacturer. If an M command is programmed with an axis motion in the same block, whether the M function is to be executed at the start or end of the block on reaching the axis position depends on the machine data setting of the machine manufacturer. Details are available in the documentation of your machine manufacturer.

M function	Function	Remarks
M08	Coolant ON	These M functions are defined by the machine manufacturer.
M09	Coolant OFF	

Specification of several M functions in one block

A maximum of five M functions can be programmed in on block. Possible combinations of M functions and possible restrictions are specified in the documentation of your machine manufacturer.

Additional auxiliary functions (B functions)

If B is not used as axis identifier, B can be used as extended auxiliary function. B functions are output to the PLC as auxiliary functions (H functions with the address extension H1=).

Example: B1234 is output as H1=1234.

4.3.6 Controlling the feedrate

4.3.6.1 Compressor in ISO dialect mode

The commands COMPON, COMPCURV, COMPCAD are commands of the Siemens language and they activate a compressor function that combines several linear blocks into one machining section. If this function is activated in Siemens mode, even linear blocks in ISO mode can be compressed with this function.

The blocks can at the most consist of the following commands:

- Block number
- G01, modal or in block
- Axis assignments
- Feedrate
- Comments

If a block contains other commands (e.g., auxiliary functions, other G codes, etc.), compression does not take place.

Value assignments with $x for G, axes and feedrate are possible, just as the skip function.

Example: These blocks are compressed

```
N5      G290
N10     COMPON
N15     G291
N20     G01 X100. Y100. F1000
N25     X100 Y100 F$3
N30     X$3 /1 Y100
N35     X100 (Axis 1)
```

These blocks are **not** compressed

```
N5      G290
N10     COMPON
N20     G291
N25     G01 X100 G17                ; G17
N30     X100 M22                    ; Auxiliary function in block
N35     X100 S200                   ; Spindle speed in block
```

4.3.6.2 Exact stop (G09, G61), continuous-path mode (G64), and tapping (G63)

The path feedrate is controlled as specified in the table below.

Identifier	G function	Efficacy of the G functions	Description
Exact stop	G09	effective only in the block in which the relevant G function is programmed	Braking and stop at end of block and position control before transition to the next block
Exact stop	G61	Modal G function; remains effective till it is deselected via G63 or G64.	Braking and stop at end of block and position control before transition to the next block
Continuous-path mode	G64	Modal G function; remains effective till it is deselected via G61 or G63.	No braking at end of block before transition to the next block

Identifier	G function	Efficacy of the G functions	Description
Tapping	G63	Modal G function; remains effective till it is deselected via G61 or G64.	No braking at end of block before transition to the next block; feedrate override is not effective

Format

G09 X... Y... Z...	;	Exact stop, non-modal
G61	;	Exact stop, modal
G64	;	Continuous-path mode
G63	;	Tapping

4.4 Additional functions

4.4.1 Program supporting functions

4.4.1.1 Fixed drilling cycles (G73, G74, G76, G80 to G87, G89)

The fixed drilling cycles simplify the creation of new programs for the programmer. Frequently occurring machining steps can be executed with a G function; without fixed cycles, several NC blocks must be programmed. Thus fixed drilling cycles shorten the machining program and save memory space.

In ISO dialect mode, a shell cycle is called which uses the functionality of the Siemens standard cycles. This way, the addresses programmed in the NC block are transferred to the shell cycle via system variables. The shell cycle adjusts this data and calls a Siemens standard cycle.

A fixed cycle could be canceled only with G80 or a G code of G code Group 1 before the program can be continued with a blockwise cycle.

The fixed drilling cycles are called with the following G functions.

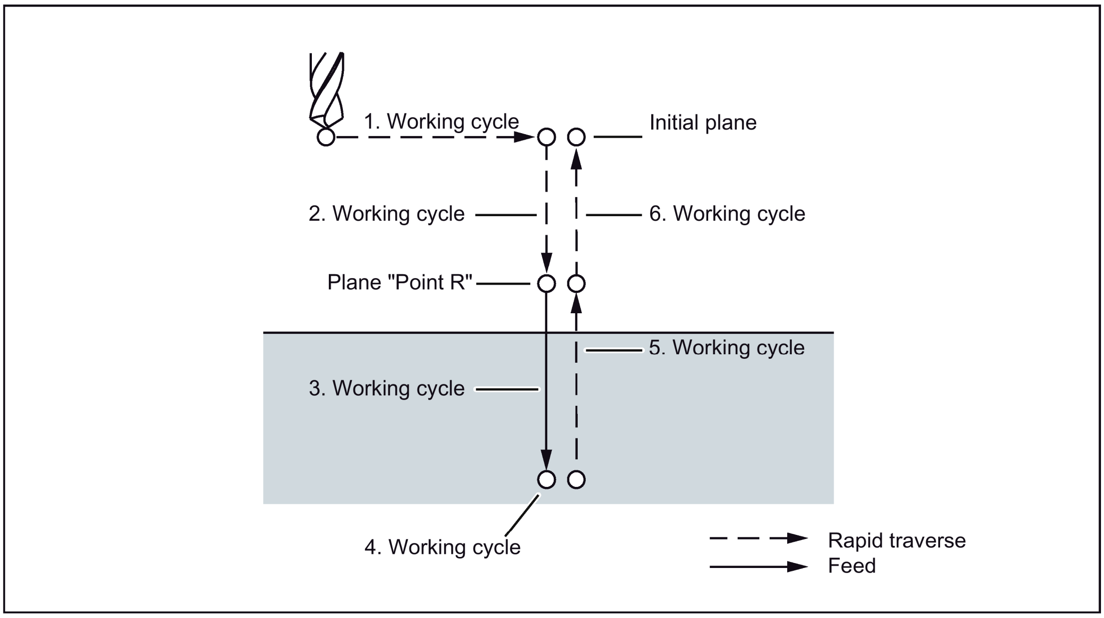
G function	Drilling (-Z direction)	Machining at drilling base	Return (+Z direction)	Applications
G73	Interrupted working feedrate (delay possible at each in-feed)	—	Rapid traverse	High-speed deep hole drilling
G74	Cutting feedrate	Spindle stop → Spindle revolution after dwelling in the opposite direction	Cutting feedrate → dwell time → Spindle turns in the opposite direction	Left-hand thread boring (in the opposite direc-tion)
G76	Cutting feedrate	Spindle positioning → Withdraw lift-off path	Rapid traverse → Return lift-off path, spindle start	Precision drilling boring
G80	—	—	—	Deselection
G81	Cutting feedrate	—	Rapid traverse	Drilling, Preboring
G82	Cutting feedrate	Dwell	Rapid traverse	Drilling, countersinking
G83	Interrupted working feedrate	—	Rapid traverse	Deep-hole drilling
G84	Cutting feedrate	Spindle stop→ Spindle start after dwelling in the opposite direction	Cutting feedrate → dwell time → Spindle turns in the opposite direction	Tapping
G85	Cutting feedrate	—	Cutting feedrate	Drilling
G86	Cutting feedrate	Spindle stop	Rapid traverse → spindle start	Drilling
G87	Spindle positioning → With-draw lift-off path → Rapid traverse → Return lift-off path → Spindle run right → Cutting feedrate	Spindle positioning after dwelling → Withdraw lift-off path	Rapid traverse → Return lift-off path → Spindle start	Drilling
G89	Cutting feedrate	Dwell	Cutting feedrate	Drilling

Explanations

On using fixed cycles, the sequence of operation in general is always as described below:

- 1. Working cycle
  Positioning in X-Y plane with cutting feedrate or rapid traverse rate
- 2. Working cycle
  Rapid traverse movement to plane R
- 3. Working cycle
  Machining up to drilling depth Z
- 4. Working cycle
  Machining at drilling base
- 5. Working cycle
  Return up to plane R with cutting feedrate or rapid traverse rate
- 6. Working cycle
  Rapid retraction with rapid traverse rate to positioning plane X-Y

Sequence of operations in the drilling cycle:



If the term "drill" is used in this Chapter, it refers only to the working cycle that are executed with the help of fixed cycles, even though naturally there are fixed cycles for tapping, boring or drilling cycles too.

Definition of the current plane

In drilling cycles, one generally assumes that the current coordinate system in which the machining operation is to be executed, is defined through the selection of plane G17, G18 or G19 and activation of a programmable work offset. Drilling axis is then always the application of this coordinate system.

Before calling the cycle, one must always select a tool length compensation. It is always effective perpendicular to the selected plane and remains active even beyond the end of the cycle.

G function	Positioning plane	Drilling axis
G17	Xp-Yp plane	Zp
G18	Zp-Xp plane	Yp
G19	Yp-Zp plane	Xp

Xp: X axis or an axis parallel to the X axis

Yp: Y axis or an axis parallel to the Y axis

Zp: Z axis or an axis parallel to the Z axis

Note

Whether the Z axis should always be used as the drilling axis can be defined with USER DATA, _ZSFI[0]. The Z axis is then always the drilling axis, if _ZSFI[0] is equal to "1".

Execution of a fixed cycle

The following is necessary to execute a fixed cycle:

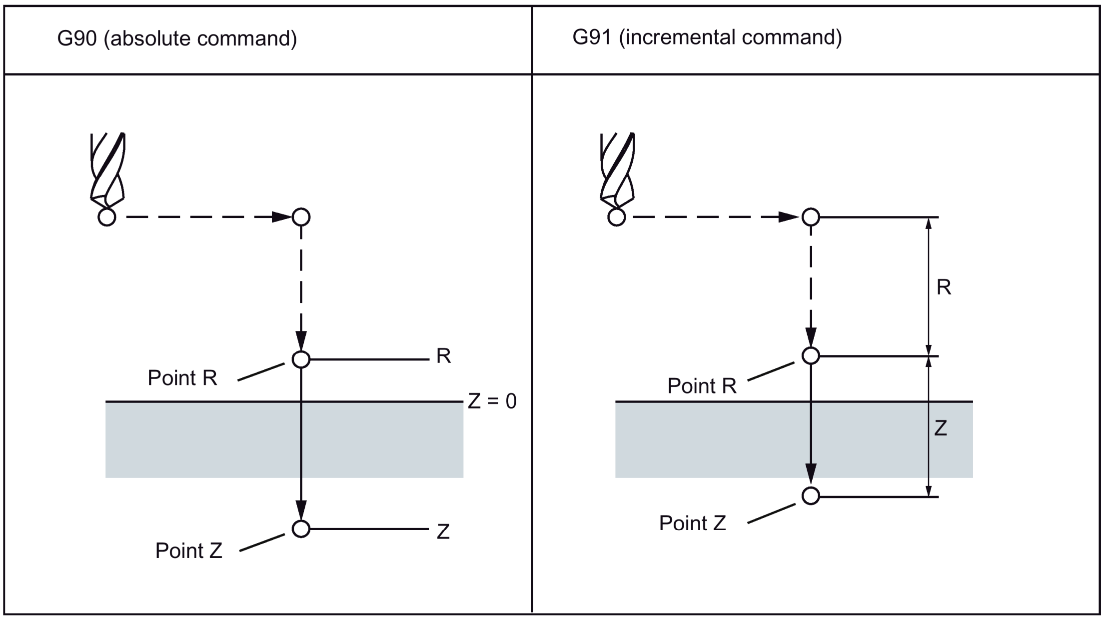
1. Cycle call

   G73, 74, 76, 81 to 87 and 89

   as a function of the desired machining

2. Data format G90/91

   Absolute/incremental command G90/G91:



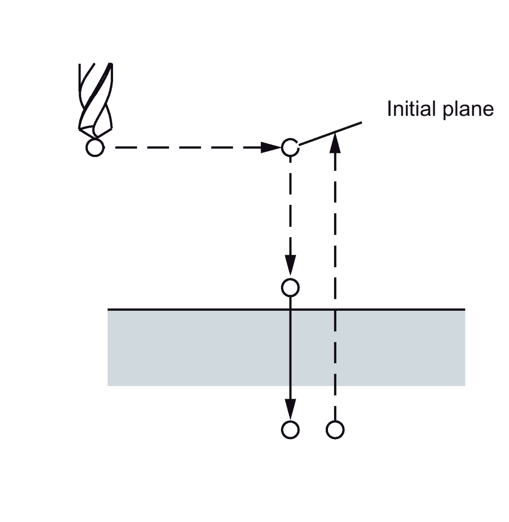
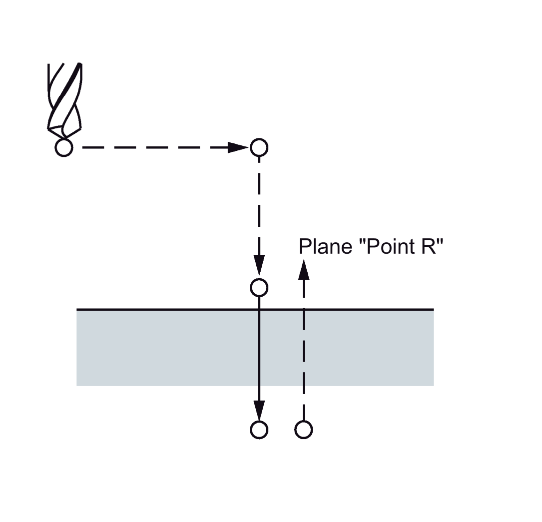
3. Drilling mode

   G73, G74, G76 and G81 to G87 and G89 are modal G functions and they remain active till they are deselected. The selected drilling cycle is called in each block. The complete parameter assignment of the drilling cycles must be programmed only during the selection (e.g., G81). Only the parameters that are supposed to change are to be programmed in the following blocks.

4. Positioning/reference plane (G98/G99)

   While using the fixed cycles, the retraction plane for the Z-axis is defined with G98/99. G98/G99 are modal G functions. The closed position is normally G98.

Plane for the return point (G98/G99):

G98 (Return to initial plane)	G99 (Return to plane "Point R")
	

Repeat

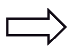
If several holes are drilled at uniform spacing, the number of repetitions is specified with "K". "K" is effective only in the block in which it is programmed. If the drilled hole position is programmed as absolute (G90), drilling is done at the same position again; hence the drilled hole position is to be specified as incremental (G91).

Comments

A cycle call remains active till it is deselected again with the G functions G80, G00, G01, G02 or G03 or another cycle call.

Symbols and numbers

The individual fixed cycles are explained in the following sections. The following symbols are used in the numbers occurring in these explanations:

	Positioning (Rapid traverse G00)
	Cutting feed (Linear interpolation G01)
	Manual feed
M19	Oriented spindle stop (The spindle stops at a defined rotary position.)
	Traverse (Rapid traverse G00)
P	Dwell

4.4.1.2 High-speed deep hole drilling cycle with chip breakage (G73)

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth. Deep hole drilling is performed with a depth infeed of a maximum defined depth executed several times, increasing gradually until the final drilling depth is reached. Optionally, the twist drill can be retracted after each infeed depth either to the reference plane + safety clearance for chip removal or by the length of the programmed retraction path for chip breakage.

Format

G73 X.. Y... R... Q... F... K... ;

X,Y: Drilled hole position

Z: Distance from Point R to the base of the drilled hole

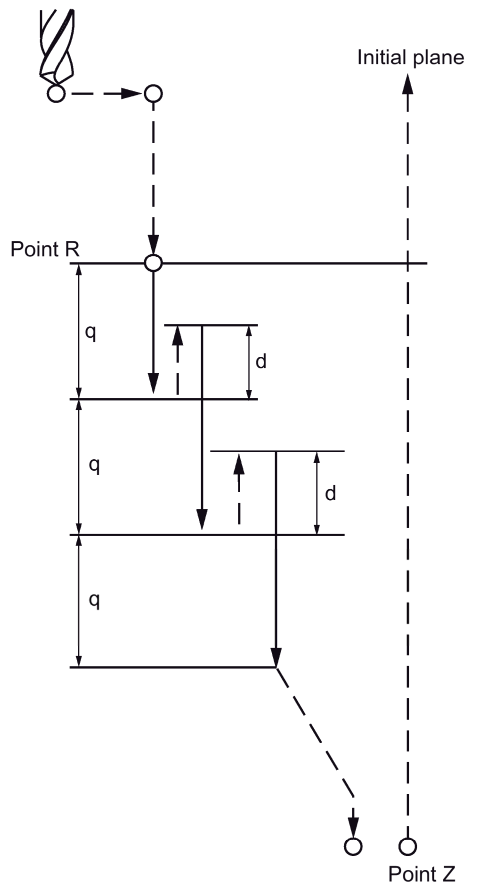
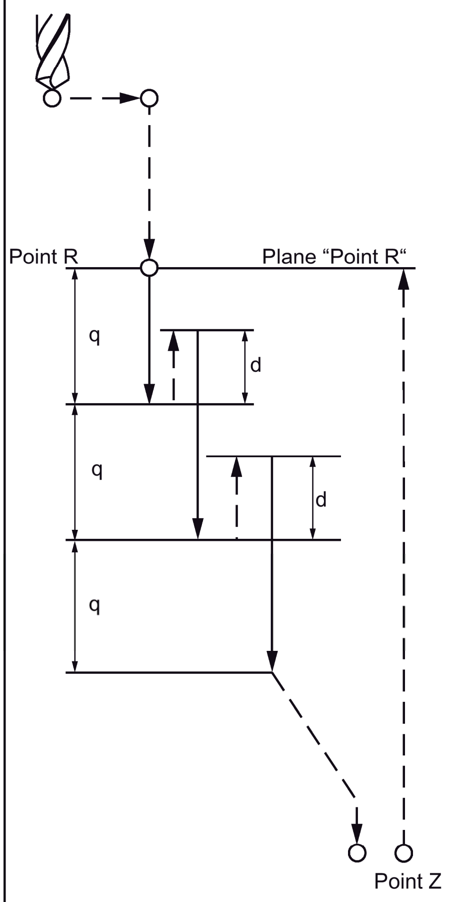
R: Distance from the initial plane to plane R

Q: Single drilling depth

F: Feedrate

K: Number of repetitions

High-speed deep hole drilling cycle with chip breakage (G73):

G73 (G98)	G73 (G99)
	

Explanations

On using the G73 cycle, the retraction motion takes place after the drilling with rapid traverse. The safety clearance can be specified with GUD _ZSFR[0]. The retraction amount from chip breaking (d) is defined with GUD _ZSFR[1]:

_ZSFR[1] > 0 Retraction amount as input

_ZSFR[1] = 0 Retraction amount in chip breaking is always 1 mm

The in-feed takes place by using the cutting depth for each cutting Q which is incremented with the retraction amount d as second in-feed.

A rapid drilling infeed results with this drilling cycle. Chip removal takes place through the retraction motion.

Restrictions

None.

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle.

Deep-hole drilling

The drilling cycle is executed only if an axis motion, e.g., is programmed with X, Y, Z or R.

Q/R

Always program Q and R in one block with an axis motion, otherwise the programmed values will not be stored modally.

Deselection

The G functions of Group 01 (G00 to G03) and G73 should not be used together in one block, as otherwise G73 is deselected

Example

```
M3 S1500                                ;Rotary motion of stem
G90 G0 Z100.
G90 G99 G73 X200. Y-150. Z-100. R50.    ;Positioning, drilled hole 1,
Q10. F150.                              ;then return to Point R
Y-500.                                  ;Positioning, drilled hole 2,
                                        ;then return to Point R
Y-700.                                  ;Positioning, drilled hole 3,
                                        ;then return to Point R
X950.                                   ;Positioning, drilled hole 4,
                                        ;then return to Point R
Y-500.                                  ;Positioning, drilled hole 5,
                                        ;then return to Point R
G98 Y-700.                              ;Positioning, drilled hole 6,
                                        ;then return to initial plane
G80                                     ;Deselection of the fixed cycle
G28 G91 X0 Y0 Z0                        ;Return to reference position
M5                                      ;Spindle stop
```

4.4.1.3 Fine drilling cycle (G76)

Precision drilling takes place with the fine drilling cycle.

Format

G76 X... Y... R... Q... P... F... K... ;

X,Y: Drilled hole position

Z_: Distance from point R to the bottom of the hole

R_: Distance from the initial plane to plane "Point R"

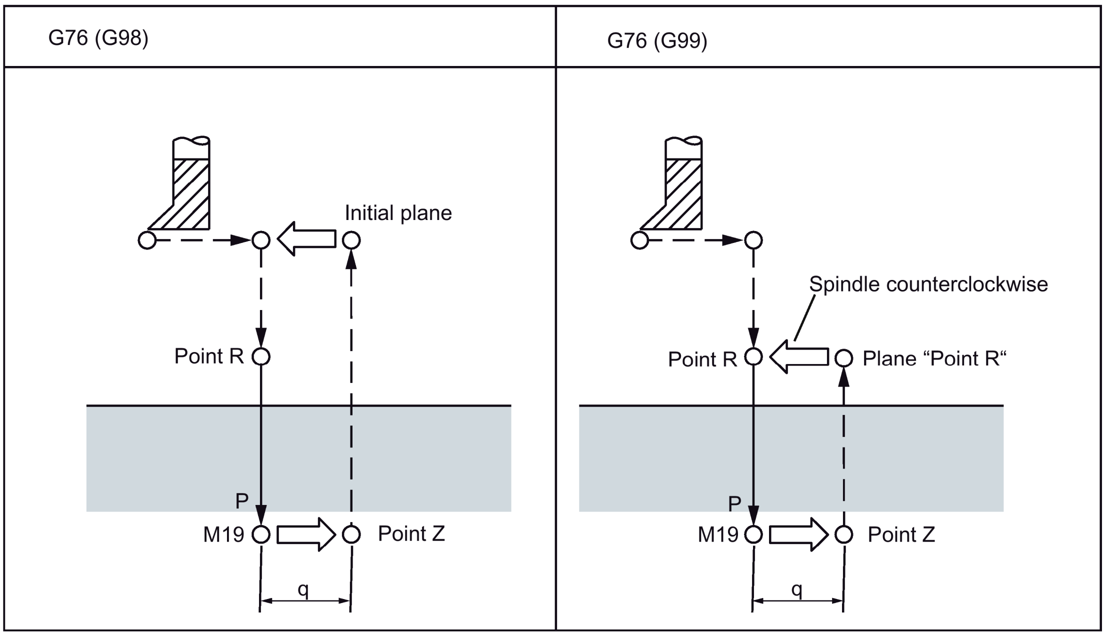
Q_: Amount of offset at the bottom of a hole

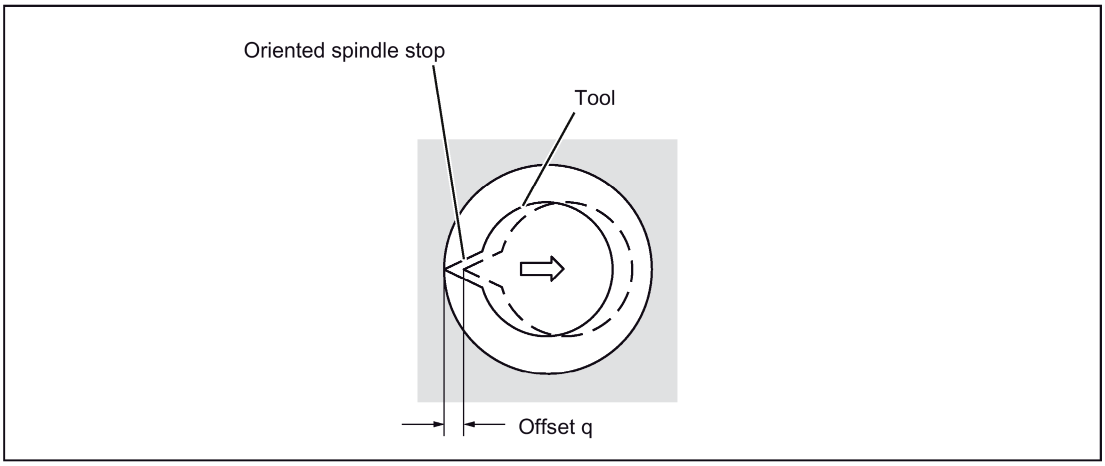
P_: Dwell time at the bottom of a hole

F_: Feedrate

K_: Number of repetitions

Fine drilling cycle (G76):





Note

Address Q is a modal value that is stored in the fixed cycles. Please ensure that this address is also used as interface for the cycles G73 and G83!

Explanations

The spindles stops at a fixed spindle position after the bottom of a hole is reached. The tool is returned opposite the tool tip.

The safety clearance can be specified with GUD _ZSFR[0]. The lift-off path can be specified with _ZSFI[5].

	G17	G18	G19
_ZSFI[5] = 1	+X	+Z	+Y
_ZSFI[5] = 0 or 2	-X	-Z	-Y
_ZSFI[5] = 3	+Y	+X	+Z
_ZSFI[5] = 4	-Y	-X	-Z

The angle must therefore be specified in USER DATA, _ZSFR[2] in such a way that the tool tip points at the opposite direction after the spindle stop, for the lift-off path.

Restrictions

None.

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle.

Drilling

The drilling cycle is executed only if an axis motion, e.g. is programmed with X, Y, Z or R.

Q/R

Always program Q and R only in one block with a retracting movement, otherwise the programmed values are not stored modally.

Only one positive value is to be specified in each case for the value of Address Q. If a negative value is specified for Q, the sign is ignored. Q is set as equal to "0" if no lift-off path is programmed. In this case, the cycle is executed without lifting.

Deselection

The G functions of Group 01 (G00 to G03) and G76 should not be used together in one block, as otherwise G76 is deselected.

Example

```
M3 S300                              ;Rotary motion of stem
G90 G0 Z100.
G90 G99 G76 X200. Y-150. Z-100. R50. ;Positioning, drilling of drilled hole 1,
Q10. P1000 F150.                     ;then return to point R and
                                     ;for 1 s stop at the bottom of a hole
Y-500.                               ;Positioning, drilled hole 2,
                                     ;then return to point R
Y-700.                               ;Positioning, drilled hole 3,
                                     ;then return to point R
X950.                                ;Positioning, drilled hole 4,
                                     ;then return to point R
Y-500.                               ;Positioning, drilled hole 5,
                                     ;then return to point R
G98 Y-700.                           ;Positioning, drilled hole 6,
                                     ;then return to initial plane
G80                                  ;Deselection of the fixed cycle
G28 G91 X0 Y0 Z0                     ;Return to reference position
M5                                   ;Spindle stop
```

4.4.1.4 Drilling cycle, counterboring (G81)

This cycle can be used for centering and preboring. The retraction motion starts immediately with rapid traverse rate on reaching the drilling depth Z.

Format

G81 X... Y... R... F... K... ;

X,Y: Drilled hole position

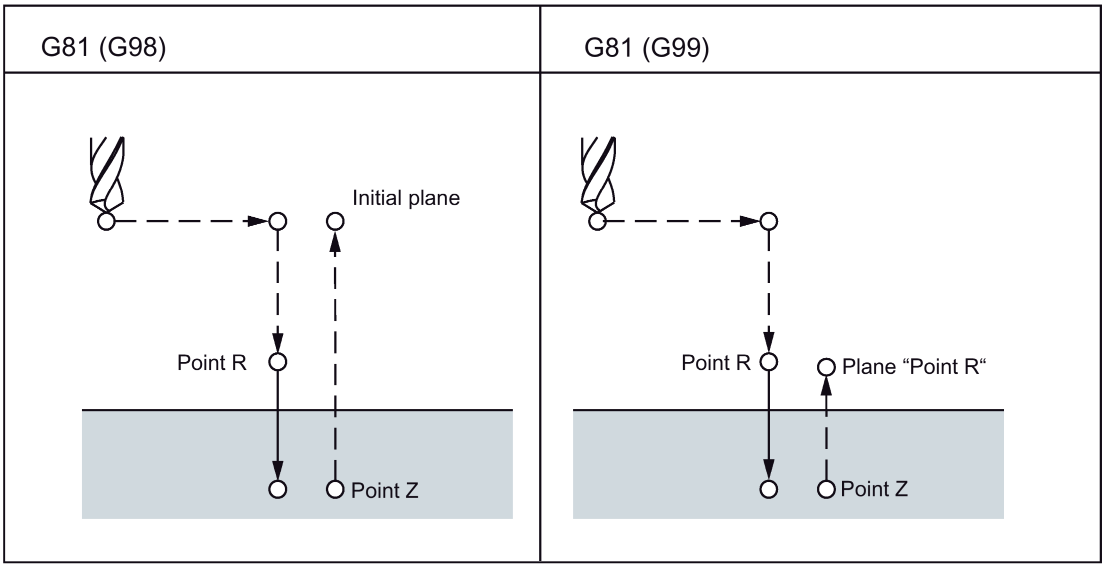
Z: Distance from point R to the bottom of the hole

R: Distance from the initial plane to plane R

F: Cutting feedrate

K: Number of repetitions

Drilling cycle counterboring (G81):



Restrictions

None.

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle.

Drilling

The drilling cycle is executed only if an axis motion, e.g. is programmed with X, Y, Z or R.

R

Always program R only in one block with an axis motion, otherwise the programmed values are not stored modally.

Deselection

The G functions of Group 01 (G00 to G03) and G76 should not be used together in one block, as otherwise G76 is deselected.

Example

```
M3 S1500                             ;Rotary motion of stem
G90 G0 Z100.
G90 G99 G81 X200. Y-150. Z-100. R50. ;Positioning, drilled hole 1,
F150.                                ;then return to point R and
                                     ;for 1 s stop at the bottom of a hole
Y-500.                               ;Positioning, drilled hole 2,
                                     ;then return to point R
Y-700.                               ;Positioning, drilled hole 3,
                                     ;then return to point R
X950.                                ;Positioning, drilled hole 4,
                                     ;then return to point R
Y-500.                               ;Positioning, drilled hole 5,
                                     ;then return to point R
G98 Y-700.                           ;Positioning, drilled hole 6,
                                     ;then return to initial plane
G80                                  ;Deselection of the fixed cycle
G28 G91 X0 Y0 Z0                     ;Return to reference position
M5                                   ;Spindle stop
```

4.4.1.5 Countersink drilling cycle (G82)

This cycle can be used for normal drilling. A programmed dwell time can be active on reaching the drilling depth Z; the retraction motion is then executed in rapid traverse.

Format

G82 X... Y... R... P... F... K... ;

X,Y: Drilled hole position

Z: Distance from point R to the bottom of the hole

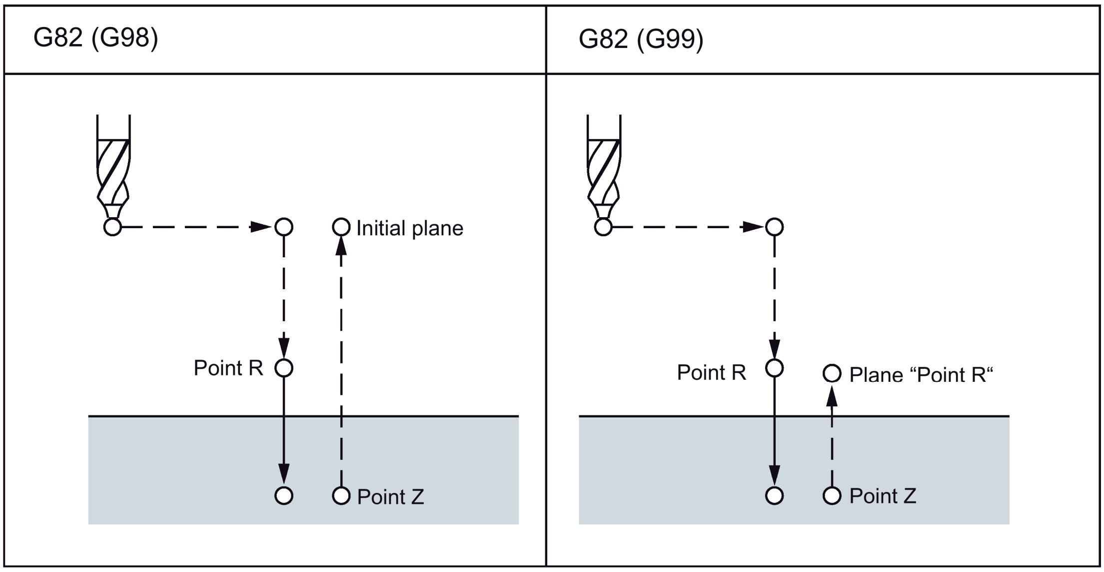
R: Distance from the initial plane to plane R

P: Dwell time at the bottom of a hole

F: Feedrate

K: Number of repetitions

Countersink drilling cycle (G82):



Restrictions

None.

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle.

Drilling

The drilling cycle is executed only if an axis motion, e.g. is programmed with X, Y, Z or R.

R

Always program R only in one block with an axis motion, otherwise the programmed values are not stored modally.

Deselection

The G functions of Group 01 (G00 to G03) and G82 should not be used together in one block, as otherwise G82 is deselected.

Example

```
M3 S2000                                ;Rotary motion of stem
G90 G0 Z100.
```

```
G90 G99 G82 X200. Y-150. Z-100. R50.    ;Positioning, drilled hole 1,
P1000 F150.                             ;stop on the bottom of a hole for 1 s
                                        ;then return to point R
Y-500.                                  ;Positioning, drilled hole 2,
                                        ;then return to point R
Y-700.                                  ;Positioning, drilled hole 3,
                                        ;then return to point R
X950.                                   ;Positioning, drilled hole 4,
                                        ;then return to point R
Y-500.                                  ;Positioning, drilled hole 5,
                                        ;then return to point R
G98 Y-700.                              ;Positioning, drilled hole 6,
                                        ;then return to initial plane
G80                                     ;Deselection of the fixed cycle
G28 G91 X0 Y0 Z0                        ;Return to reference position
M5                                      ;Spindle stop
```

4.4.1.6 Deep hole drilling cycle with chip removal (G83)

The cycle "Deep hole drilling with chips removal" can, e.g., be used for deep hole drilling with recutting.

Format

G83 X... Y... R... Q... F... K... ;

X,Y: Drilled hole position

Z: Distance from point R to the bottom of the hole

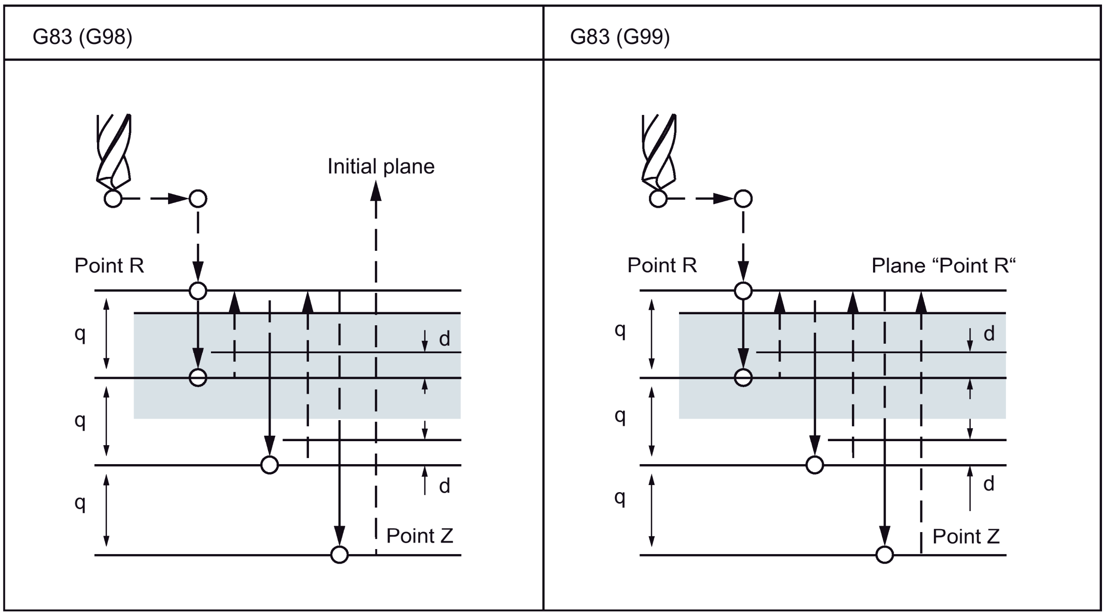
R: Distance from the initial plane to plane R

Q: Cutting depth for each cutting feedrate

F: Feedrate

K: Number of repetitions

Deep hole drilling cycle with chip removal (G83):



Restrictions

None.

Explanations

After the programmed cutting depth is reached for each cutting feedrate Q, the return to the reference plane R takes place in rapid traverse. The approach motion for a renewed step is also executed in rapid traverse, around the path (d) that can be set in USER DATA, _ZSFR[10]. The path and the cutting depth for each cutting feedrate Q are traversed with cutting feedrate. Q is incremental without having to specify signs.

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle.

Drilling

The drilling cycle is executed only if an axis motion, e.g. X, Y, Z or R is programmed.

Q/R

Always program Q and R in one block with an axis motion, otherwise the programmed values will not be stored modally.

Deselection

The G functions of Group 01 (G00 to G03) and G83 should not be used together in one block, as otherwise G83 is deselected.

Example

```
M3 S2000                                ;Rotary motion of stem
G90 G0 Z100.
G90 G99 G83 X200. Y-150. Z-100. R50.    ;Positioning, drilled hole 1,
Q10. F150.                              ;then return to point R
Y-500.                                  ;Positioning, drilled hole 2,
                                        ;then return to point R
Y-700.                                  ;Positioning, drilled hole 3,
                                        ;then return to point R
X950.                                   ;Positioning, drilled hole 4,
                                        ;then return to point R
Y-500.                                  ;Positioning, drilled hole 5,
                                        ;then return to point R
G98 Y-700.                              ;Positioning, drilled hole 6,
                                        ;then return to initial plane
G80                                     ;Deselection of the fixed cycle
G28 G91 X0 Y0 Z0                        ;Return to reference position
M5                                      ;Spindle stop
```

Note

If _ZSFR[10]

- > 0: the value is used for the derivative path "d" (minimum distance 0.001).
- = 0: the drilling depth is 30 mm and the value of the derivative path is always 0.6 mm. The drilling depth/50 formula is always used for larger drilling depths (maximum value 7 mm).

4.4.1.7 Boring cycle (G85)

Format

G85 X... Y... R... F... K... ;

X,Y: Drilled hole position

Z: Distance from point R to the bottom of the hole

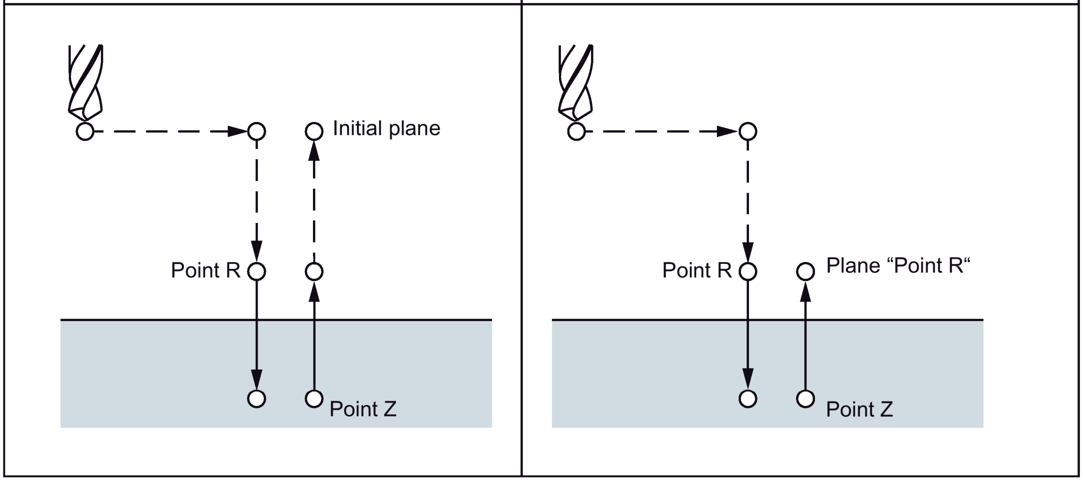
R: Distance from the initial plane to plane R

F: Feedrate

K: Number of repetitions

Boring cycle (G85):

G85 (G98)	G85 (G99)



Explanations

A traversing movement takes place to point R in rapid traverse after the positioning along the X and Y axis. Drilling takes place from point R to point Z. On reaching point Z, a retraction motion to point R takes place with cutting feedrate.

Restrictions

None.

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle.

Drilling

The drilling cycle is executed only if an axis motion, e.g. is programmed with X, Y, Z or R.

R

Always program R only in one block with an axis motion, otherwise the programmed values are not stored modally.

Deselection

The G functions of Group 01 (G00 to G03) and G85 should not be used together in one block, as otherwise G85 is deselected.

Example

```
M3 S150                               ;Rotary motion of stem
G90 G0 Z100.
G90 G99 G85 X200. Y-150. Z-100. R50.  ;Positioning, drilled hole 1,
F150.                                 ;then return to point R
Y-500.                                ;Positioning, drilled hole 2,
                                      ;then return to point R
Y-700.                                ;Positioning, drilled hole 3,
                                      ;then return to point R
X950.                                 ;Positioning, drilled hole 4,
                                      ;then return to point R
Y-500.                                ;Positioning, drilled hole 5,
                                      ;then return to point R
G98 Y-700.                            ;Positioning, drilled hole 6,
                                      ;then return to initial plane
G80                                   ;Deselection of the fixed cycle
G28 G91 X0 Y0 Z0                      ;Return to reference position
M5                                    ;Spindle stop
```

4.4.1.8 Boring cycle (G86)

Format

G86 X... Y... R... F... K... ;

X,Y: Drilled hole position

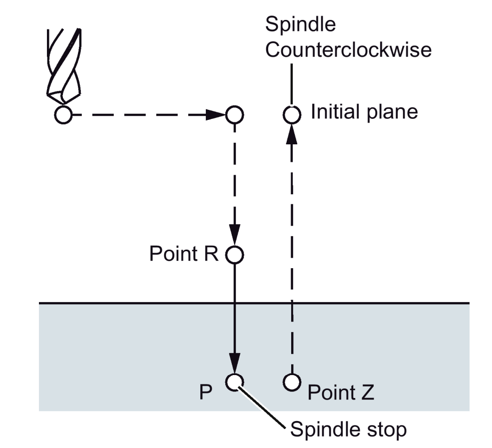
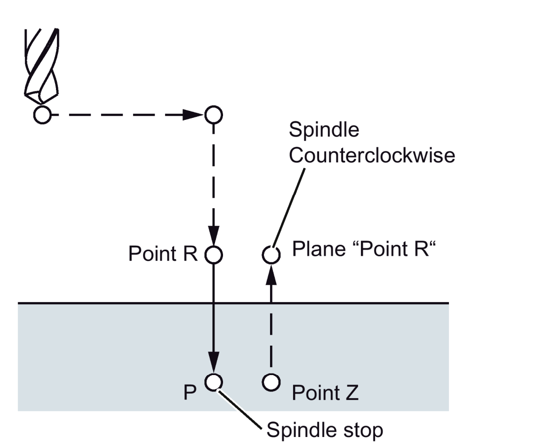
Z: Distance from point R to the bottom of the hole

R: Distance from the initial plane to point R

F: Feedrate

K: Number of repetitions

Boring cycle (G86):

G86 (G98)	G86 (G99)
	

Explanations

Point R is approached in rapid traverse after positioning the X and Y axes. Drilling takes place from point R to point Z. The tool returns in rapid traverse mode after the spindle is stopped at the bottom of a hole.

Restrictions

None.

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle.

Drilling

The drilling cycle is executed only if an axis motion, e.g. is programmed with X, Y, Z or R.

R

Always program R only in one block with an axis motion, otherwise the programmed values are not stored modally.

Deselection

The G functions of Group 01 (G00 to G03) and G86 should not be used together in one block, as otherwise G86 is deselected.

Example

```
M3 S150                             ;Rotary motion of stem
G90 G0 Z100.
G90 G99 G86 X200. Y-150. Z-100. R50.    ;Positioning, drilled hole 1,
```

```
F150.                                    ;then return to point R
Y-500.                                   ;Positioning, drilled hole 2,
                                         ;then return to point R
Y-700.                                   ;Positioning, drilled hole 3,
                                         ;then return to point R
X950.                                    ;Positioning, drilled hole 4,
                                         ;then return to point R
Y-500.                                   ;Positioning, drilled hole 5,
                                         ;then return to point R
G98 Y-700.                               ;Positioning, drilled hole 6,
                                         ;then return to initial plane
G80                                      ;Deselection of the fixed cycle
G28 G91 X0 Y0 Z0                         ;Return to reference position
M5                                       ;Spindle stop
```

4.4.1.9 Boring cycle - reverse countersinking (G87)

This cycle can be used for precision drilling.

Format

G87 X... Y... R... Q... P... F... K... ;

X,Y: Drilled hole position

Z: Distance from bottom of the hole to point Z

R: Distance from the initial plane to plane R (bottom of a hole)

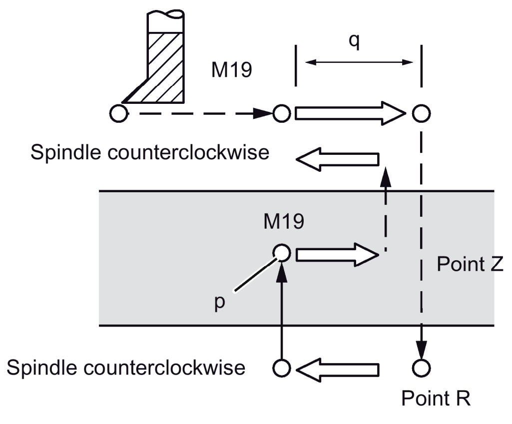
Q: Tool offset amount

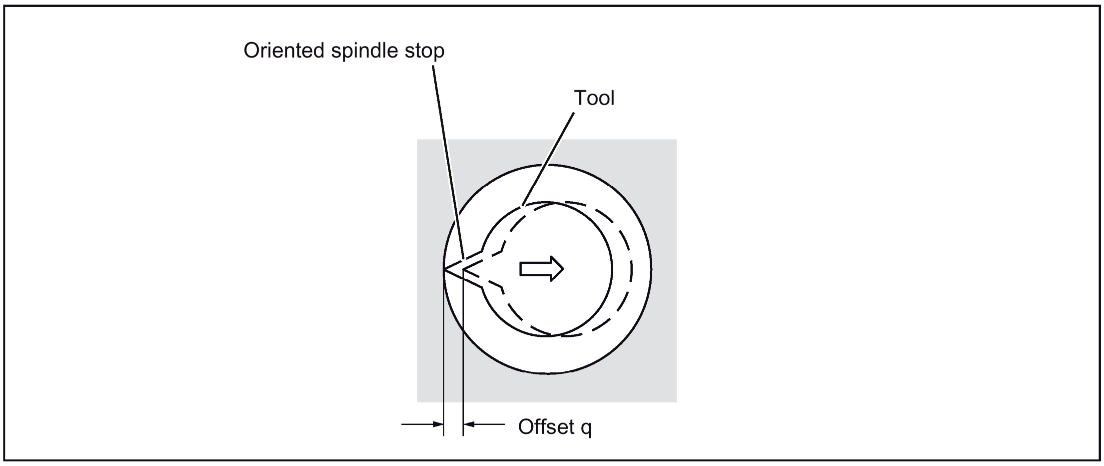
P: Dwell time

F: Feedrate

K: Number of repetitions

Boring cycle, reverse countersinking (G87):

G87 (G98)	G87 (G99)
	Is not used



Note

Address Q (gear change at the base of a drilled hole) is a modal value that is stored in fixed cycles. Please ensure that this address is also used as interface for the cycles G73 and G83!

Explanations

The spindle stops at a fixed rotary position after positioning along the X and Y axis. The tool travels in the direction opposite to that of the tool tip. It is positioned on the bottom of a hole (Point R) at rapid traverse.

Finally, the tool is shifted in the direction of the tool tip and the spindle is moved with clockwise rotation. Drilling takes place along the Z axis in the positive direction up to point Z.

The spindles stops at a fixed spindle position after the bottom of a hole is reached. The tool is returned opposite the tool tip.

The safety clearance can be specified with GUD _ZSFR[0].

The lift-off path can be specified with _ZSFI[5].

	G17	G18	G19
_ZSFR[5] = 1	+X	+Z	+Y
_ZSFI[5] = 0 or 2	-X	-Z	-Y
_ZSFI[5] = 3	+Y	+X	+Z
_ZSFI[5] = 4	-Y	-X	-Z

The angle must therefore be specified in USER DATA, _ZSFR[2] in such a way that the tool tip points at the opposite direction after the spindle stop for the lift-off path.

Example:

If plane G17 is activated, the tool tip must point in direction +X.

Restrictions

None.

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle.

Drilling

The drilling cycle is executed only if an axis motion, e.g. is programmed with X, Y, Z or R.

Q/R

Always program Q and R in one block with an axis motion, otherwise the programmed values will not be stored modally.

Only one positive value is to be specified in each case for the value of Address Q. If a negative value is specified for "Q", the sign is ignored. "Q" is set as equal to "0" if no lift-off path is programmed. In this case, the cycle is executed without lifting.

Deselection

The G functions of Group 01 (G00 to G03) and G87 should not be used together in one block, as otherwise G87 is deselected.

Example

```
M3 S400                                 ;Rotary motion of stem
G90 G0 Z100.
G90 G87 X200. Y-150. Z-100. R50. Q3.    ;Positioning, drilled hole 1,
P1000 F150.                             ;orientation towards initial plane,
                                        ;then travel 3 mm,
                                        ;halt for 1 s at point Z
Y-500.                                  ;Positioning, drilled hole 2
Y-700.                                  ;Positioning, drilled hole 3
X950.                                   ;Positioning, drilled hole 4
Y-500.                                  ;Positioning, drilled hole 5
G98 Y-700.                              ;Positioning, drilled hole 6
G80                                     ;Deselection of the fixed cycle
G28 G91 X0 Y0 Z0                        ;Return to reference position
M5                                      ;Spindle stop
```

4.4.1.10 Boring cycle (G89)

Format

G89 X... Y... R... P... F... K... ;

X,Y: Drilled hole position

Z: Distance from point R to the bottom of the hole

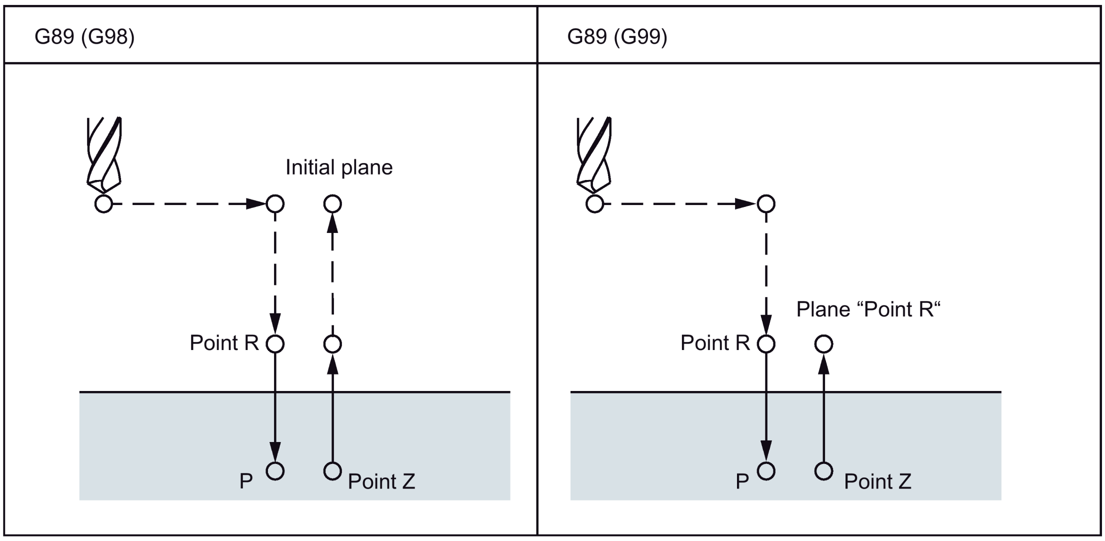
R: Distance from the initial plane to point R

P: Dwell time at the bottom of a hole

F: Feedrate

K: Number of repetitions

Boring cycle (G89):



Explanations

This cycle is similar to G86, with the only exception that here, a dwell time at the bottom of the hole is still available.

Before programming G89, the spindle must be started with an M function.

Restrictions

None.

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle.

Drilling

The drilling cycle is executed only if an axis motion, e.g. is programmed with X, Y, Z or R.

R

Always program R only in one block with an axis motion, otherwise the programmed values are not stored modally.

Deselection

The G functions of Group 01 (G00 to G03) and G89 should not be used together in one block, as otherwise G89 is deselected.

Example

```
M3 S150                                  ;Rotary motion of stem
G90 G0 Z100.
G90 G99 G89 X200. Y-150. Z-100. R50.     ;Positioning, drilled hole 1,
P1000 F150.                              ;then 1 s stop at the bottom of a hole
Y-500.                                   ;Positioning, drilled hole 2,
                                         ;then return to point R
Y-700.                                   ;Positioning, drilled hole 3,
                                         ;then return to point R
X950.                                    ;Positioning, drilled hole 4,
                                         ;then return to point R
Y-500.                                   ;Positioning, drilled hole 5,
                                         ;then return to point R
G98 Y-700.                               ;Positioning, drilled hole 6,
                                         ;then return to initial plane
G80                                      ;Deselection of the fixed cycle
G28 G91 X0 Y0 Z0                         ;Return to reference position
M5                                       ;Spindle stop
```

4.4.1.11 "Drilling a right-hand thread without any compensating chuck" cycle (G84)

The tool drills at the programmed spindle speed and feedrate to the entered final thread depth. With G84 you can produce rigid tapping.

Note

G84 can be used if the spindle to be used for the drilling operation is technically able to be operated in the position-controlled spindle mode.

Format

G84 X... Y... Z... R... P... F... K... ;

X,Y: Drilled hole position

Z: Distance from point R to the bottom of the hole

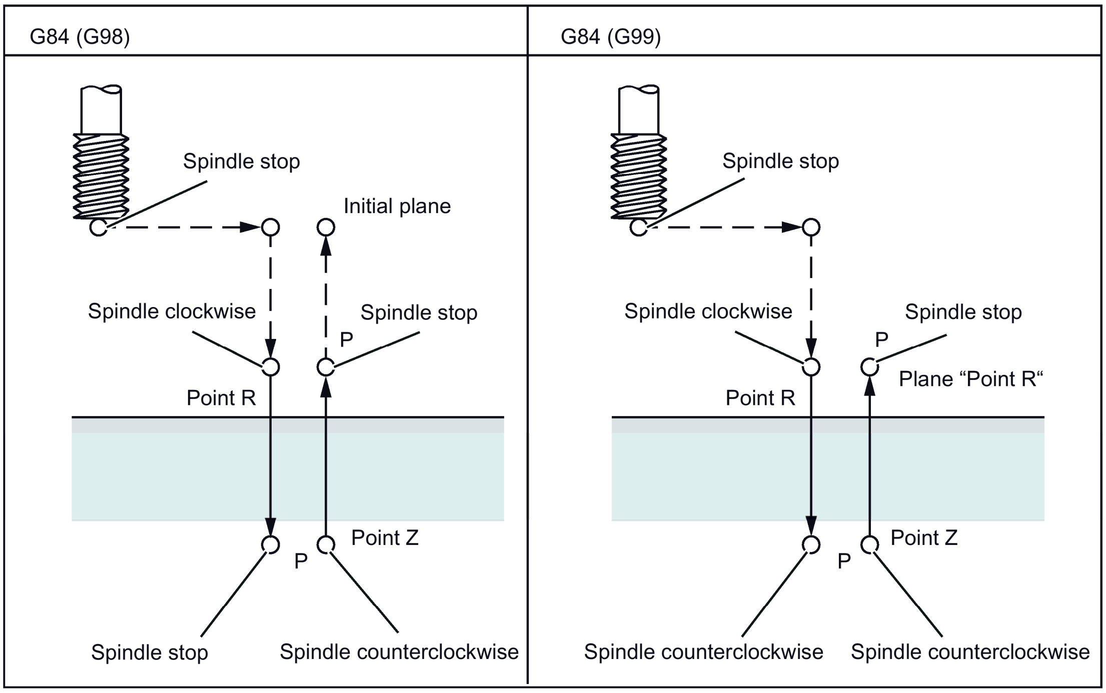
R: Distance from the initial plane to plane R

P: Dwell time at the bottom of the hole and at point R during return

F: Cutting feedrate

K: Number of repetitions (if necessary)

"Drilling a right-hand thread without any compensating chuck" cycle (G84):



Explanations

The cycle creates the following sequence of motions:

- Approach of reference plane shifted by the amount of the safety clearance with G0.
- Oriented spindle stop and transfer of spindle in the Axis mode.
- Tapping to the final drilling depth.
- Execution of dwell time at thread depth.
- Retraction to the reference plane and reversion of direction of rotation brought forward by the safety clearance.
- Retraction to the retraction plane with G0.

During tapping, rapid traverse override and spindle override are accepted at 100%.

The speed of rotation can be affected during the retraction with GUD _ZSFI[2]. Example: _ZSFI[2]=120; the retraction takes place at 120% of the speed during tapping.

Restrictions

None.

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle. An alarm is output if the drilling axis in the "Drilling without compensating chuck" mode is changed over.

Tapping

The drilling cycle is executed only if an axis motion, e.g. is programmed with X, Y, Z or R.

R

Always program R only in one block with an axis motion, otherwise the programmed values are not stored modally.

Deselection

The G functions of Group 01 (G00 to G03) and G84 should not be used together in one block, as otherwise G84 is deselected.

S command

An error message is displayed if the specified gear stage is one step higher than the maximum permissible value.

F function

An error message is displayed if the value specified for the cutting feedrate exceeds the maximum permissible value.

Unit of the F command

G function	Metric input	Input in inch	Remarks
G94	1 mm/min	0.01 inch/min	Decimal point programming is permitted
G95	0.01 mm/rev	0.0001 inch/rev	Decimal point programming is permitted

Example

Feedrate for the Z axis 1000 mm/min

Spindle speed 1000 rev/min

Thread lead 1.0 mm

```
<Programming as feedrate per minute>
S100 M03S1000
G94                                    ;Feedrate per minute
G00 X100.0 Y100.0                      ;Positioning
G84 Z-50.0 R-10.0 F1000                ;Tapping without compensating chuck
<Programming as revolutional feedrate>
G95                                    ; Rev. feedrate
G00 X100.0 Y100.0                      ;Positioning
G84 Z-50.0 R-10.0 F1.0                 ;Tapping without compensating chuck
```

4.4.1.12 "Drilling a left-hand thread without any compensating chuck" cycle (G74)

The tool drills at the programmed spindle speed and feedrate to the entered final thread depth. With G74 you can produce left rigid tapping.

Note

G74 can be used if the spindle to be used for the drilling operation is technically able to be operated in the position-controlled spindle mode.

Format

G74 X... Y... Z... R... P... F... K... ;

X,Y: Drilled hole position

Z: Distance from point R to the bottom of the hole

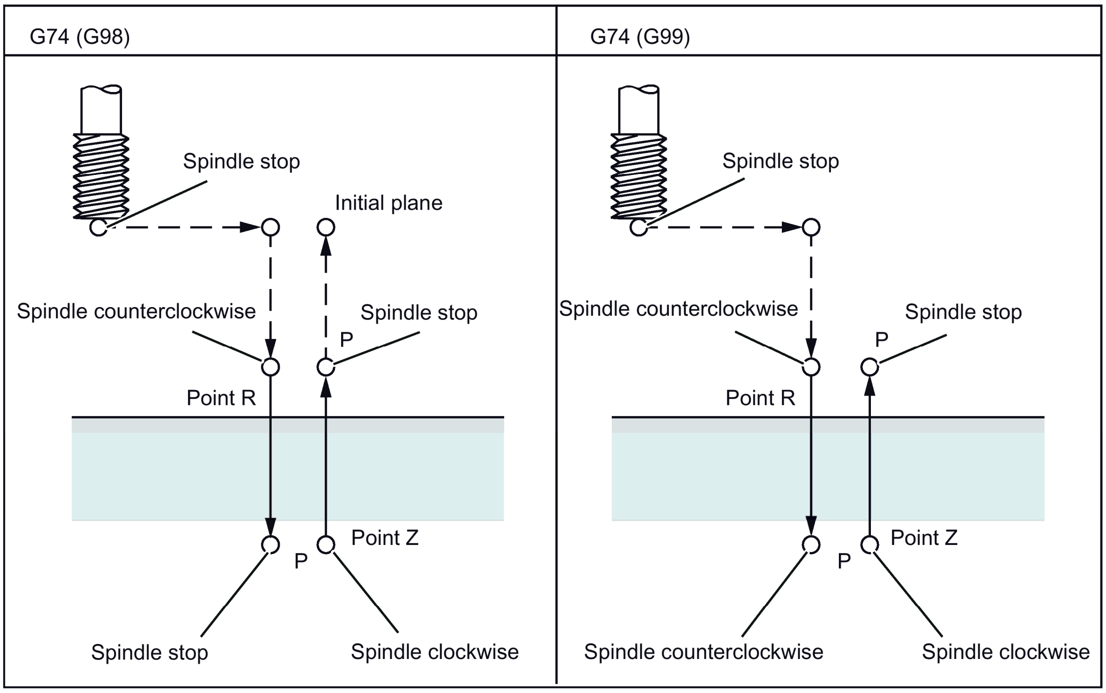
R: Distance from the initial plane to point R

P: Dwell time at the bottom of the hole and at point R during return

F: Cutting feedrate

K: Number of repetitions (if necessary)

"Drilling a left-hand thread without any compensating chuck" cycle (G74):



Explanations

The cycle creates the following sequence of motions:

- Approach of reference plane shifted by the amount of the safety clearance with G0.
- Oriented spindle stop and transfer of spindle in the Axis mode.
- Tapping to the final drilling depth.
- Execution of dwell time at thread depth.
- Retraction to the reference plane and reversion of direction of rotation brought forward by the safety clearance.
- Retraction to the retraction plane with G0.

During tapping, rapid traverse override and spindle override are accepted at 100%.

The speed of rotation can be affected during the retraction with GUD _ZSFI[2]. Example: _ZSFI[2]=120; the retraction takes place at 120% of the speed during tapping.

Restrictions

None.

Changeover of the axes

Before changing over the drilling axis, one must first deselect the fixed cycle. An alarm is output if the drilling axis in the "Drilling without compensating chuck" mode is changed over.

Tapping

The drilling cycle is executed only if an axis motion, e.g. is programmed with X, Y, Z or R.

R

Always program R only in one block with an axis motion, otherwise the programmed values are not stored modally.

Deselection

The G functions of Group 01 (G00 to G03) and G74 should not be used together in one block, as otherwise G74 is deselected.

S command

An error message is displayed if the specified gear stage is one step higher than the maximum permissible value.

F function

An error message is displayed if the value specified for the cutting feedrate exceeds the maximum permissible value.

Unit of the F command

G function	Metric input	Input in inch	Remarks
G94	1 mm/min	0.01 inch/min	Decimal point programming is permitted
G95	0.01 mm/rev	0.0001 inch/rev	Decimal point programming is permitted

Example

Feedrate for the Z axis 1000 mm/min

Spindle speed 1000 rev/min

Thread lead 1.0 mm

```
<Programming as feedrate per minute>
S100 M03S1000
G94                                     ;Feedrate per minute
G00 X100.0 Y100.0                       ;Positioning
G74 Z-50.0 R-10.0 F1000                 ;Tapping without any compensating chuck
<Programming as revolutional feedrate>
G95                                     ; Rev. feedrate
G00 X100.0 Y100.0                       ;Positioning
G74 Z-50.0 R-10.0 F1.0                  ;Tapping without any compensating chuck
```

4.4.1.13 Left or right tapping cycle (G84/G74)

Due to the chips adhering to the tool and an increased resistance associated with this, it may be difficult to perform the deep hole tapping without compensating chuck. In such cases the tapping cycle with chip breakage or chip removal is helpful.

The cutting movement is executed in this cycle till the root is reached. There are a total of two tapping cycles for this: Deep hole tapping with chip breakage and deep hole tapping with chip removal.

The G84 and G74 cycles can be selected with GUD _ZSFI[1] as follows:

_ZSFI[1] = 2: Deep hole tapping with chip breakage

_ZSFI[1] = 3: Deep hole tapping with chip removal

Format

G84 (or G74) X... Y... Z... R... P... Q... F... K... ;

X,Y: Drilled hole position

Z: Distance from point R to the bottom of the hole

R: Distance from the initial plane to "Point R"

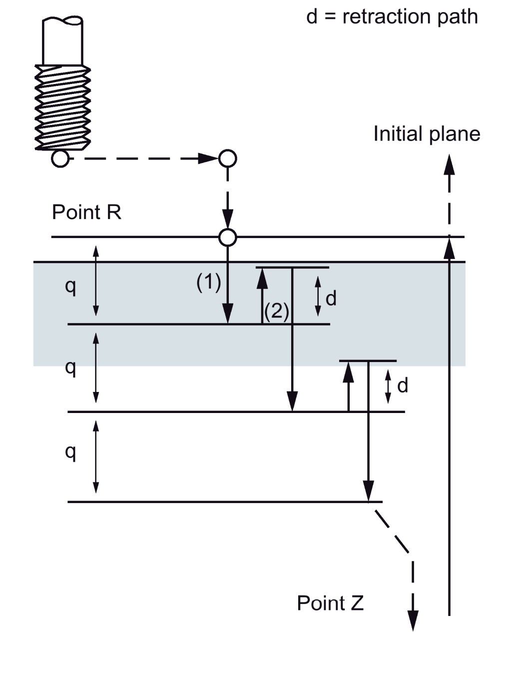
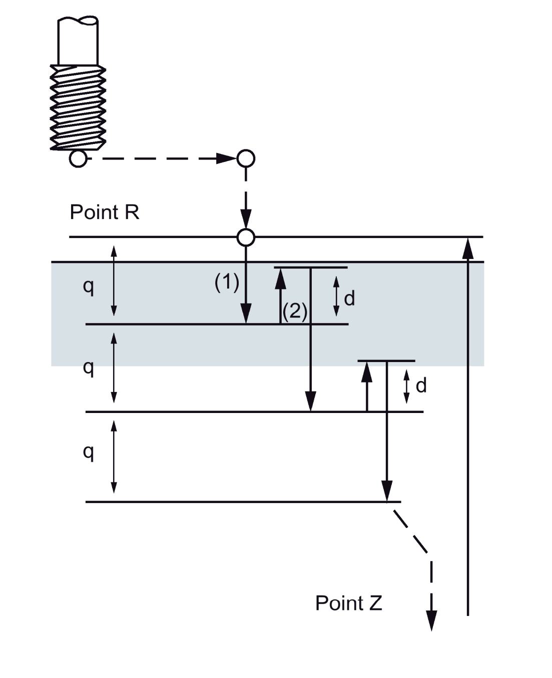
P: Dwell time at the bottom of the hole and at point R during return

Q: Cutting depth for each cutting feedrate

F: Feedrate

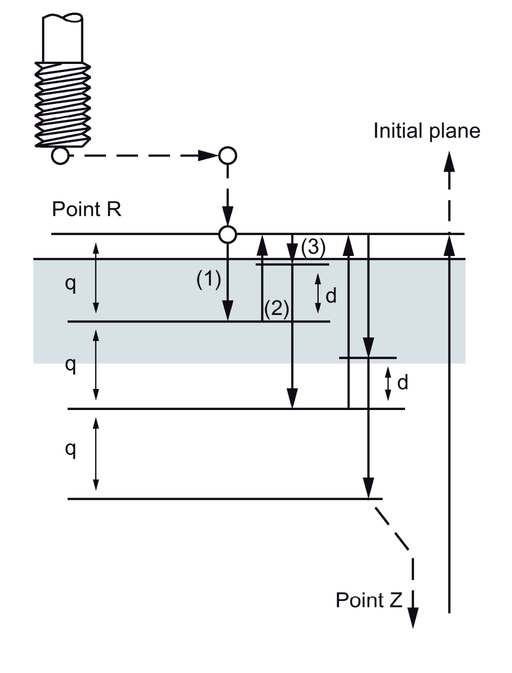
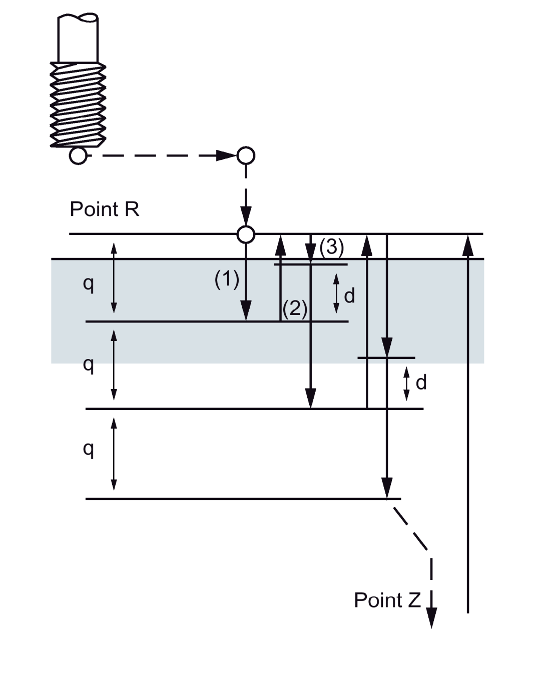
K: Number of repetitions

Deep hole tapping with chip breakage (USER DATA, _ZSFI[1] = 2):

G84, G74 (G98)	G84, G74 (G99)
	

1. The tool is traversed with the programmed feedrate.
2. The retraction velocity can be affected with USER DATA, _ZSFI[2].

Deep hole tapping with chip breakage (USER DATA, _ZSFI[1] = 3):

d = Path for starting the cutting processes 	

Deep hole tapping with chip breakage/removal

After positioning along the X and Y axes, there is a traversing movement at rapid traverse to point R. The machining is done from point R onwards with a cutting depth Q (cutting depth per cutting feedrate). Finally, the tool is retracted by the distance d. If a value not equal to 100% is specified in USER DATA, _ZSFI[2], it can be specified whether the retraction is overlaid or not. The spindle stops as soon as point Z is reached; the direction of rotation is finally reversed and a retraction is executed. The retraction path d is set in USER DATA, _ZSFR[1].

Note

If 0 is specified in _ZSFR[1], the default setting of 1 mm or 1 inch is effective for the retraction distance.

If 0 mm or 0 inch is to be specified, a value less than the travel triggering should be specified.

4.4.1.14 Deselection of a fixed cycle (G80)

Fixed cycles can be deselected with G80.

Format

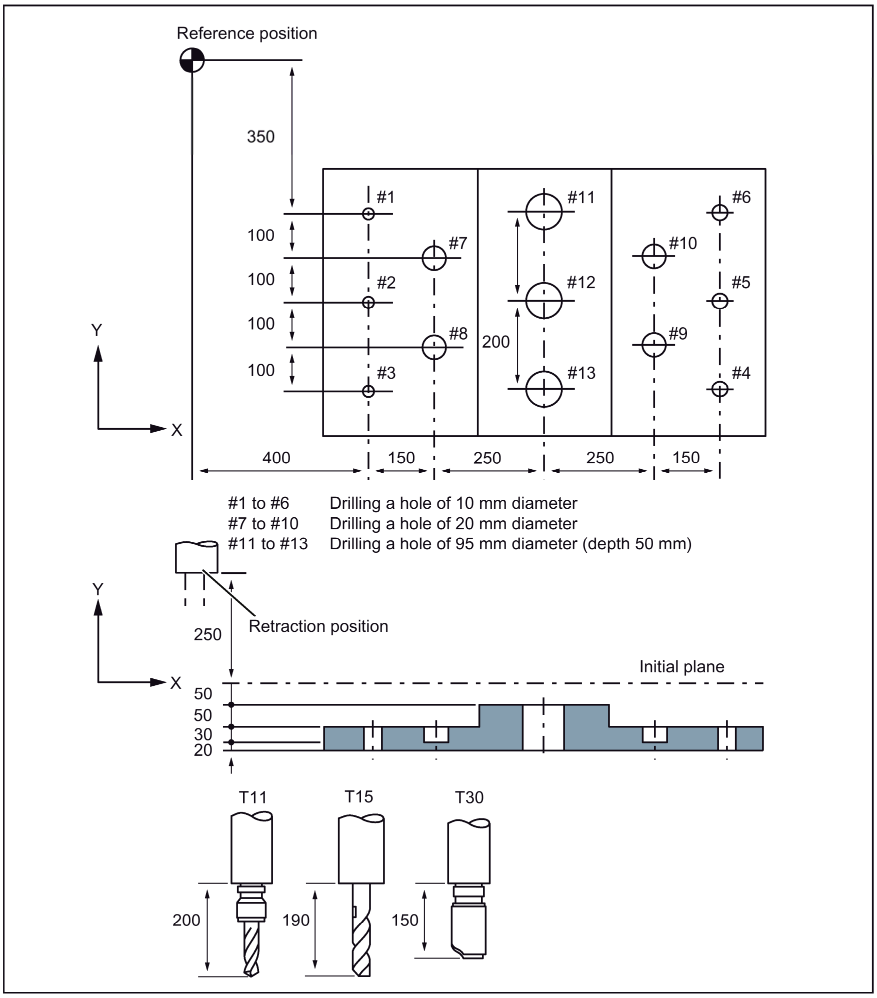
G80;

Explanations

All modal cycles are deselected in ISO mode with G80 or with a G function of the 1st group (G00, G03,...).

4.4.1.15 Program example with a tool length compensation and fixed cycles

Program example (drilling cycle):



Offset value +200.0 is set in TO No. 11, +190.0 is set in TO No. 15 and +150.0 is set in tool offset No. 30.

Sample program

```
;
N001 G49                                ; Deselect the tool length compensation
N002 G10 L10 P11 R200.                  ; Setting the tool offset 11 to +200.
N003 G10 L10 P15 R190.                  ; Setting the tool offset 15 to +190.
N004 G10 L10 P30 R150.                  ; Setting the tool offset 30 to +150.
N005 G92 X0 Y0 Z0                       ; Setting the coordinates at the reference position
                                        ;
N006 G90 G00 Z250.0 T11 M6              ; Tool change
N007 G43 Z0 H11                         ; Initial plane, tool length compensation
N008 S30 M3                             ; Spindle start
N009 g99 G81 X400.0 Y-350.0 Z-153.0 R-  ; Positioning, then drill #1
97.0 F1200
N010 Y-550.0                            ; Positioning, then drilling #2 and return to plane
                                        ; point R
N011 G98 Y-750.0                        ; Positioning, then drilling #3 and return to ini-
                                        ; tial plane
N012 G99 X1200.0                        ; Positioning, then drilling #4 and return to plane
                                        ; point R
N013 Y-550.0                            ; Positioning, then drilling #5 and return to plane
                                        ; point R
N014 G98 Y-350.0                        ; Positioning, then drilling #6 and return to ini-
                                        ; tial plane
N015 G00 X0 Y0 M5                       ; Return to reference position,
                                        ; Spindle stop
N016 G49 Z250.0 T15 M6                  ; Deselection of tool length compensation, tool
                                        ; change
N017 G43 Z0 H15                         ; Initial plane, tool length compensation
N018 S20 M3                             ; Spindle start
N019 G99 G82 X550.0 Y-450.0 Z-130.0 R-  ; Positioning, then drilling #7 and return to plane
97.0 P300 F700                          ; point R
N020 G98 Y-650.0                        ; Positioning, then drilling #8 and return to ini-
                                        ; tial plane
N021 G99 X1050.0                        ; Positioning, then drilling #9 and return to plane
                                        ; point R
N022 G98 Y-450.0                        ; Positioning, then drilling #10 and return to ini-
                                        ; tial plane
N023 G00 X0 Y0 M5                       ; Return to reference position,
                                        ; Spindle stop
N024 G49 Z250.0 T30 M6                  ; Deselection of tool length compensation, tool
                                        ; change
 N025 G43 Z0 H30                        ; Initial plane, tool length compensation
N026 S10 M3                             ; Spindle start
N027 G85 G99 X800.0 Y-350.0 Z-153.0     ; Positioning, then drilling #11 and return to plane
R47.0 F500                              ; point R
N028 G91 Y-200.0 K2                     ; Positioning, then drilling #12 and 13, and return
                                        ; to plane point R
G80                                     ; Deselection of the fixed cycle
N029 G28 X0 Y0 M5                       ; Return to reference position,
                                        ; Spindle stop
N030 G49 Z0                             ; Deselect the tool length compensation
N031 M30                                ; End of the program
```

4.4.2 Programmable data input (G10)

4.4.2.1 Changing the tool offset value

Existing tool offsets can be overwritten via G10. It is not possible to create new tool offsets.

Format

G10 L10 P... R... ; Tool length compensation, geometry

G10 L11 P... R... ; Tool length compensation, wear and tear

G10 L12 P... R... ; Tool radius compensation, geometry

G10 L13 P... R... ; Tool radius compensation, wear and tear

P: Number of tool offset memory

R: Value statement

L1 can also be programmed instead of L11.

4.4.2.2 M function for calling subroutines (M98, M99)

This function can be used if subroutines are stored in the part program memory. Subroutines that are registered in the memory and whose program numbers are assigned can be called and executed any number of times.

Commands

The following M functions are used to call subroutines.

M function	Function
M98	Subprogram call
M99	End of subroutine

Subroutine call (M98)

- M98 Pnnnnmmmm

  m: Program no. (max. 4 digits)
  n: No. of repetitions (max. 4 digits)

  Before using program M98 Pnnnnmmmm to call a program, name the program correctly, that is, add the program number always to 4 digits with 0.
- If for example, M98 P21 is programmed, the part program memory is browsed by program name 21.mpf and the subroutine is executed once. To call the subroutine three times, one must program M98 P30021. An alarm is output if the specified program no. is not found.
- A nesting of subroutines is possible, up to 16 subroutines are allowed. An alarm is output if more subroutine levels are assigned than is allowed.

End of subroutine (M99)

A subroutine is ended with the command M99 Pxxxx and program processing is continued in Block No. Nxxxx. The control system first searches forward for the block number (from the subroutine call up to the end of the program). If no matching block number is found, the part program is eventually searched in the reverse direction (in the direction of the part start of program).

If M99 is without a block number (Pxxxx) in a main program, the control goes to the start of the main program and the main program is processed afresh. In case of M99 with navigation to the block number in the main program (M99xxxx), the block number is always searched from the start of program.

4.4.3 Eight-digit program number (M98, G65/G66)

An eight-digit program number selection is activated with the machine data 20734 $MC_EXTERN_FUNCTION_MASK, Bit 6=1. This function affects M98 and G65/66.

y: Number of program runs

x: Program number

Subprogram call

$MC_EXTERN_FUNCTION_MASK, Bit 6 = 0

M98 Pyyyyxxxx or

M98 Pxxxx Lyyyy

Max. four-digit program number

Addition of program number always to 4 digits with 0

Example:

M98 P20012: calls 0012.mpf 2 flows

M98 P123 L2: calls 0123.mpf 2 flows

$MC_EXTERN_FUNCTION_MASK, Bit 6 = 1

M98 Pxxxxxxxx Lyyyy

There is no extension with 0, even if the program number has less than 4 digits.

The programming of number of passes and program number in P(Pyyyyxxxxx) is not possible, the number of passes must always be programmed with L!

Example:

M98 P123: calls 123.mpf 1 Pass

M98 P20012: calls 20012.mpf 1 Pass

Caution: This is no longer compatible with ISO dialect original

M98 P12345 L2: calls 12345.mpf 2 Passes

Modal and blockwise Macro G65/G66

$MC_EXTERN_FUNCTION_MASK, Bit 6 = 0

G65 Pxxxx Lyyyy

Addition of program number to 4 digits with 0. Program number with more than 4 digits leads to an alarm.

$MC_EXTERN_FUNCTION_MASK, Bit 6 = 1

G65 Pxxxx Lyyyy

There is no extension with 0, even if the program number has less than 4 digits. A program number with more than 8 digits leads to an alarm.

4.4.4 Polar coordinates (G15, G16)

While programming in polar coordinates, the positions in the coordinate system are defined with a radius and/or angle. Polar coordinate programming is selected with G16. It is deselected again with G15. The first axis of the plane is interpreted as polar radius, the second axis as polar angle.

Format

```
G17 (G18, G19) G90 (G91) G16          ;Polar coordinates command ON
G90 (G91) X... Y... Z...              ;Polar coordinates command
...
...
G15                                   ;Polar coordinates command OFF
```

G16: Polar coordinates command

G15: Deselection of polar coordinates command

G17, G18, G19: Selection of plane

G90: The pole is located on the workpiece zero.

G91: The pole is located on the current position.

X, Y, Z: First axis: Radius of polar coordinate, second axis: Angle of polar coordinate

---

Note

If the pole is moved from the current position to the workpiece zero, the radius is calculated as the distance from the current position to the workpiece zero.

Example

```
N5 G17 G90 X0 Y0
N10 G16 X100. Y45.            ;Polar coordinates ON,
                              ;the pole is at the workpiece zero,
                              ;Position X 70,711 Y 70,711
                              ;in the Cartesian coordinate system
N15 G91 X100 Y0               ;the pole is the current Position,
                              ;i.e. the Position X 141.421 Y 141.421
N20 G90 Y90.                  ;No X in the block
                              ;The pole is on the workpiece zero,
                              ;Radius = SORT(X*X +Y*Y) = 184.776
G15
```

The polar radius is always taken as absolute value, while the polar angle can be interpreted as absolute value as well as incremental value.

Note

If _ZSFI[3]

- = 0: disables the polar coordinate function.
- = 1: enables the polar coordinate function.

4.4.5 Measuring functions

4.4.5.1 Measuring with "delete distance-to-go" (G31)

Measuring with "Delete distance-to-go possible" is activated by specifying "G31 X... Y... Z... F... ;". The linear interpolation is interrupted and the distance-to-go of the axes is deleted if, during the linear interpolation, the measurement input of the 1st probe is active. The program is continued with the next block.

Format

G31 X... Y... Z... F... ;

G31: Non-modal G function (operates only in the block in which it is programmed)

PLC signal "Measurement input = 1"

With the rising edge of the measurement input 1, the current axis positions are stored in the axial system parameters or $AA_MM[<Axis>], $AA_MW[<Axis>]. These parameters can be read in Siemens mode.

$AA_MW[X]	Saving the coordinate value of the X axis in the workpiece coordinate system
$AA_MW[Y]	Saving the coordinate value of the Y axis in the workpiece coordinate system
$AA_MW[Z]	Saving the coordinate value of the Z axis in the workpiece coordinate system
$AA_MM[X]	Saving the coordinate value of the X axis in the machine coordinate system
$AA_MM[Y]	Saving the coordinate value of the Y axis in the machine coordinate system
$AA_MM[Z]	Saving the coordinate value of the Z axis in the machine coordinate system

Note

Alarm 21700 is output if G31 is activated when the measuring signal is still active.

Program continuation after the measuring signal

If incremental axis positions are programmed in the next block, these axis positions are related to the measuring point, i.e. the reference point of the incremental position is the axis position at which the delete distance-to-go was executed by the measuring signal.

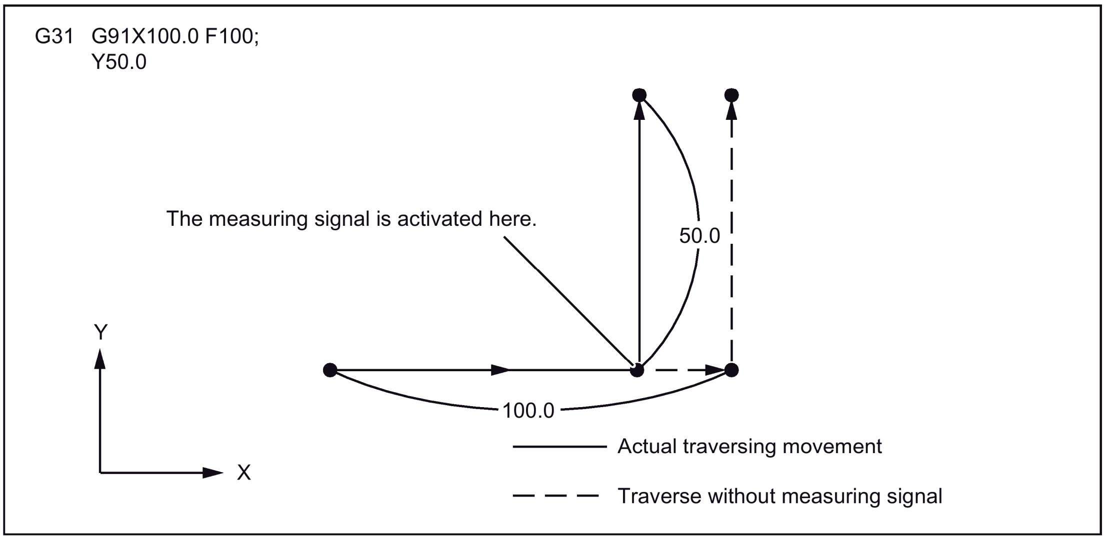
If the axis positions in the next block are programmed as absolute, then the programmed positions are approached.

Note

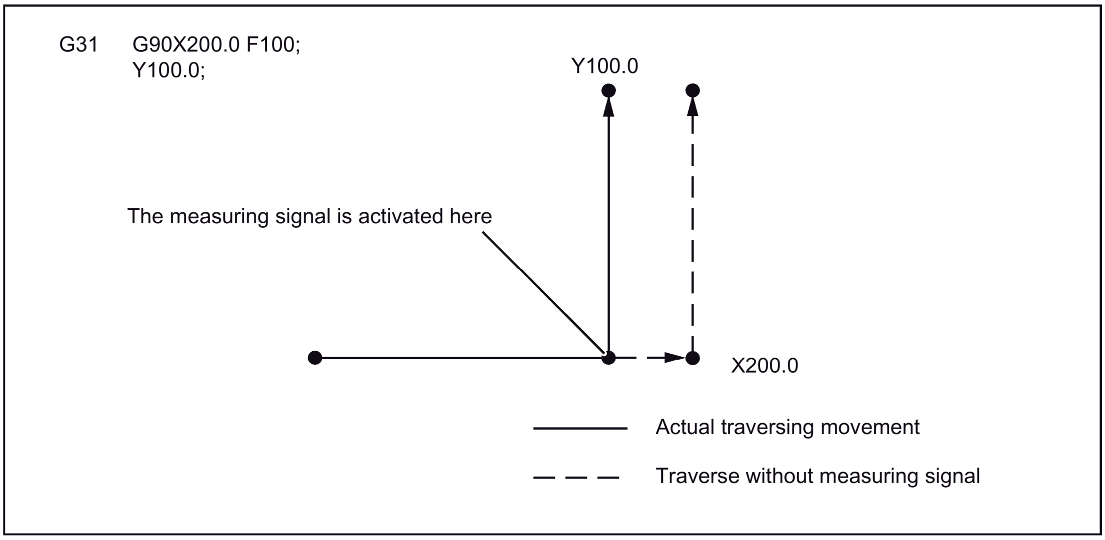
No cutter radius compensation should be active in a block containing G31. Hence, the cutter radius compensation is to be deselected before programming of G31, with G40.

Example

G31 with incremental position specification:

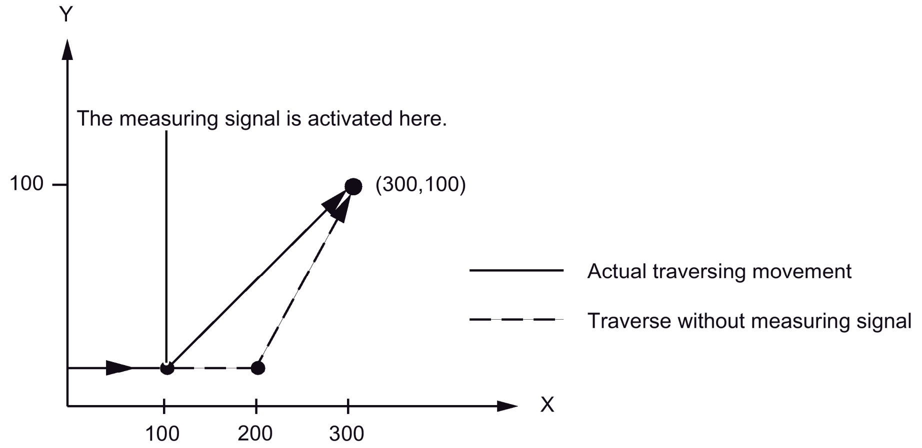


G31 with absolute position specification:



G31 is an absolute command for 2 axes:

```
G31    G90X200.0 F100;
       X300.0 Y100.0;
```



4.4.5.2 "Tool life control" function

Tool life monitoring and workpiece count can be undertaken with Siemens Tool Management.

4.4.6 Macro programs

Macros may consist of several part program blocks that are completed with M99. In principle, macros are subroutines that are called with G65 Pxx or G66 Pxx in the part program.

Macros that are called with G65 are non-modal. Macros that are called with G66 are modal and are deselected again with G67.

4.4.6.1 Differences with subroutines

Macro programs (G65, G66) can be used to specify parameters that can be evaluated in the macro programs. No parameters can be specified in subroutine calls (M98).

4.4.6.2 Macro program call (G65, G66, G67)

Macro programs are generally executed immediately after their call.

The procedure of calling a macro program is described in the table below.

Call method	Command code	Remarks
Simple call	G65	
Modal call (a)	G66	Deselection through G67

Simple call (G65):
Format

G65 P_ L_ ;

A macro program to which a program number was assigned with "P" is called and executed "L" times by specifying "G65 P ... L... <Argument>; ".

The required parameters must be programmed in the same block (with G65).

Explanation

In a part program block containing G65 or G66, the address Pxx is interpreted as program number of the subroutine in which the macro functionality is programmed. The number of passes of the macro can be defined with the address Lxx. All other addresses in this part program block are interpreted as transfer parameters and their programmed values are stored in the system variables $C_A to $C_Z. These system variables can be read in the subroutine and evaluated for the macro functionality. If other macros with parameter transfer are called in a macro (subroutine), then the transfer parameters in the subroutine must be saved in internal variable before the new macro call.

To enable internal variable definitions, one must switch automatically to Siemens mode during macro call. One can do this by inserting the instruction PROC<Program name> in the first line of the macro program. If another macro call is programmed in the subroutine, then ISO dialect mode must be reselected in advance.

For the P and L commands, see the table below:

Address	Description	Number of digits
P	Program number	4 to 8 digits
L	Number of repetitions	

System variables for the addresses I, J, K

As the addresses I, J, and K can be programmed up to ten times in a block containing macro call, the system variables of these addresses must be accessed with an array index. The syntax of these three system variables thus is $C_I[..], $C_J[..], $C_K[..]. The values remain in the programmed sequence in the array. The number of I, J, K addresses programmed in the block is given in the variables $C_I_NUM, $C_J_NUM, $C_K_NUM.

The transfer parameters I, J, K for macro calls are treated in each case as one block even if the individual addresses are not programmed. If a parameter is reprogrammed, or a following parameter based on the I, J, K sequence was programmed, it belongs to the next block.

The system variables $C_I_ORDER, $C_J_ORDER, $C_K_ORDER are set to detect the programming sequence in ISO mode. These are identical arrays of $C_I, $C_K and they contain the associated numbers of the parameters.

Note

The transfer parameters can be read only in the subroutine in Siemens mode.

Example:

```
N5 I10 J10 K30 J22 K55 I44 K33
  Block1  Block2  Block3
$C_I[0]=10
$C_I[1]=44
$C_I_ORDER[0]=1
$C_I_ORDER[1]=3

$C_J[0]=10
$C_J[1]=22
$C_J_ORDER[0]=1
$C_J_ORDER[1]=2

$C_K[0]=30
$C_K[1]=55
$C_K[2]=33
$C_K_ORDER[0]=1
$C_K_ORDER[1]=2
$C_K_ORDER[2]=3
```

Cycle parameter $C_x_PROG

In the ISO-dialect-0 mode, the programmed values can be evaluated in different ways, depending on the programming method (integer or actual value). The different evaluation is activated through a machine data.

If the MD is set, the control system responds as in the following example:

X100 ; X axis is traversed by 100 mm (100. with point) => actual value

Y200 ; Y-axis is traversed by 0.2 mm (200 without point) => integer value

If the addresses programmed in the block are used as transfer parameters of cycles, then the programmed values always exist as real values in the $C_x variables. For integer values, one cannot take recourse to the programming method (real/integer) in the cycles any more, and therefore there is no evaluation of the programmed values with the correct conversion factor.

There are two system variables $C_TYP_PROG. $C_TYP_PROG for information as to whether REAL or INTEGER programming was undertaken. The structure is the same as that of $C_ALL_PROG and $C_INC_PROG. If the value is programmed as INTEGER, then Bit is set to 0, for REAL it is set to 1. If the value is programmed over a variable $<Number>, then the corresponding bit is also set to 1.

Example:

P1234 A100. X100 -> $C_TYP_PROG == 1.

Only Bit 0 is present, because only A was programmed as REAL.

P1234 A100. C20. X100 -> $C_TYP_PROG == 5.

Bit 1 and Bit 3 (A and C) are present.

Restrictions:

A maximum of ten I, J, K parameters can be programmed in each block. Only one bit each is provided for I, J, K in the variable $C_TYP_PROG. Hence in $C_TYP_PROG the corresponding bit for I, J and K is always set to 0. Therefore it cannot be derived whether I, J or K is programmed as REAL or as INTEGER.

Modal call (G66, G67)

A modal macro program is called with G66. The specified macro program is executed only if the specified conditions are fulfilled.

- The modal macro program is activated on specifying "G66 P... L... <Parameters>;". The transfer parameters are handled as in G65.
- G66 is deselected by G67.

For modal call conditions, see the table below:

Call conditions	Function for mode selection	Function for mode deselection
after executing a traversing command	G66	G67

Specification of a parameter

The transfer parameters are defined by programming an Address A - Z.

Interrelation between addresses and system variables

Interrelation between addresses and variables	
Address	**System variable**
A	$C_A
B	$C_B
C	$C_C
D	$C_D
E	$C_E
F	$C_F
H	$C_H
I	$C_I[0]
J	$C_J[0]
K	$C_K[0]
M	$C_M
Q	$C_Q
R	$C_R
S	$C_S
T	$C_T
U	$C_U

Interrelation between addresses and variables	
V	$C_V
W	$C_W
X	$C_X
Y	$C_Y
Z	$C_Z

Interrelation between addresses and system variables

To be able to use I, J and K, these must be specified in the I, J, K sequence.

As the I, J and K addresses in a block containing a macro call can be programmed up to 10 times, access to the system variables within the macro program for these addresses must take place with an index. The syntax of these three system variables thus is $C_I[..], $C_J[..], $C_K[..]. The corresponding values are saved in the matrix in the sequence in which they were programmed. The number of I, J, K addresses programmed in the block is saved in the variables $C_I_NUM, $C_J_NUM and $C_K_NUM.

Unlike for the remaining variables, one index must always be specified while reading the three variables. The index "0" is always used for cycle calls (e.g. G81), for example, N100 R10 = $C_I[0]

Interrelation between addresses and variables	
Address	**System variable**
A	$C_A
B	$C_B
C	$C_C
I1	$C_I[0]
J1	$C_J[0]
K1	$C_K[0]
I2	$C_I[1]
J2	$C_J[1]
K2	$C_K[1]
I3	$C_I[2]
J3	$C_J[2]
K3	$C_K[2]
I4	$C_I[3]
J4	$C_J[3]
K4	$C_K[3]
I5	$C_I[4]
J5	$C_J[4]
K5	$C_K[4]
I6	$C_I[5]
J6	$C_J[5]
K6	$C_K[5]
I7	$C_I[6]
J7	$C_J[6]
K7	$C_K[6]
I8	$C_I[7]
J8	$C_J[7]
K8	$C_K[7]
I9	$C_I[8]
J9	$C_J[8]
K9	$C_K[8]

Interrelation between addresses and variables	
I10	$C_I[9]
J10	$C_J[9]
K10	$C_K[9]

Note

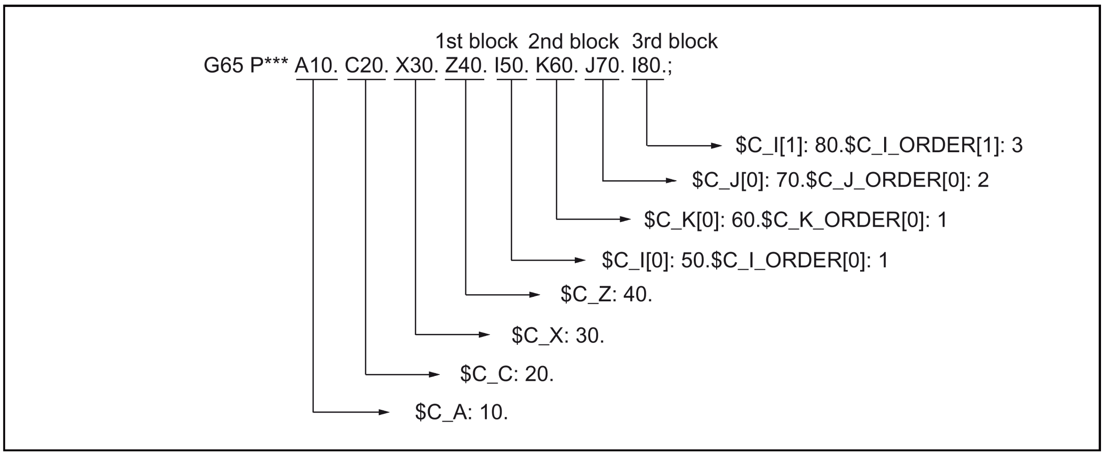
If more than one block of I, J or K addresses are specified, then the sequence of the addresses for each block of I/J/K is determined in such a way that the numbers of the variables are defined according to their sequence.

Example of entering a parameter

The value of the parameter contains a sign and a decimal point, independently of the address.

The value of the parameters is always saved as actual value.

Example of entering an argument:



Execution of macro programs in the Siemens and ISO modes

A called macro program can be called either in Siemens mode or in ISO mode. The language mode in which the program is executed is defined in the first block of the macro program.

If a PROC <Program name> instruction exists in the first block of a macro program, then an automatic changeover to Siemens mode is conducted. If this instruction is missing, the processing is done in ISO mode.

The transfer parameters can be saved in local variables by executing a program in Siemens mode. In ISO mode however, it is not possible to store transfer parameters in local variables.

To read transfer parameters in a macro program executed in ISO mode, one must first change over to Siemens mode with the G290 command.

Examples

Main program with macro call:

```
_N_M10_MPF:
N10 M3 S1000 F1000
N20 X100 Y50 Z33
N30 G65 P10 F55 X150 Y100 S2000
N40 X50
N50 ....
N200 M30
```

Macro program in Siemens mode:

```
_N_0010_SPF:
PROC 0010 ; Changeover to Siemens mode
```

```
N10 DEF REAL X_AXIS ,Y_AXIS, S_SPEED, FEED
N15 X_AXIS = $C_X Y_AXIS = $C_Y S_SPEED = $C_S FEED = $C_F
N20 G01 F=FEED G95 S=S_SPEED
...
N80 M17
```

Macro program in ISO mode:

```
_N_0010_SPF:
G290; Changeover to Siemens mode,
    ; to read the transfer parameters
N15 X_AXIS = $C_X Y_AXIS = $C_Y S_SPEED = $C_S FEED = $C_F
N20 G01 F=$C_F G95 S=$C_S
N10 G1 X=$C_X Y=$C_Y
G291; Changeover to ISO mode,
N15 M3 G54 T1
N20
...
N80 M99
```

4.4.6.3 Macro call via G function (G21)

Macro call

A macro can be called with a G number analogous to G65.

The replacement of 50 G functions can be configured via machine data:

10816 $MN_EXTERN_G_NO_MAC_CYCLE and

10817 $MN_EXTERN_G_NO_MAC_CYCLE_NAME.

The parameters programmed in the block are stored in the $C_Variables. The number of macro repetitions is programmed with Address L. The number of the programmed G macros is stored in the variable $C_G. All the other G functions programmed in the block are treated as normal G functions. The programming sequence of the addresses and G functions in the block is random, and it does not have any effect on the functionality.

Further information about the parameters programmed in this block is available in Chapter "Macro Program Call (G65, G66, G67)".

Restrictions

- The macro call with a G function can be executed only in ISO mode (G290).
- Only one G function can be replaced per part program line (or in general, only one subroutine call). If there are possible conflicts with other subroutine calls, e.g. if a modal subroutine is active, the system outputs Alarm 12722 "Several ISO_M/T macro- or cycle calls in block".
- No other G or M macro or M subroutine can be called if a G macro is active. In this case, M macros or M subroutines are executed as M functions. G macros are executed as G functions, provided a corresponding G function exists; otherwise Alarm 12470 "Unknown G function" is output.
- Otherwise the same restrictions are applicable as for G65.

Configuration examples

Calling the subroutine G21_MAKRO via G function G21

$MN_EXTERN_G_NO_MAC_CYCLE[0] = 21

$MN_EXTERN_G_NO_MAC_CYCLE_NAME[0] = "G21_MAKRO"

$MN_EXTERN_G_NO_MAC_CYCLE[1] = 123

$MN_EXTERN_G_NO_MAC_CYCLE_NAME[1] = "G123_MAKRO"

$MN_EXTERN_G_NO_MAC_CYCLE[2] = 421

$MN_EXTERN_G_NO_MAC_CYCLE_NAME[2] = "G123_MAKRO"

Programming example

```
PROC MAIN
. . .
N0090 G291                                  ; ISO mode
N0100 G1 G21 X10 Y20 F1000 G90              ; Call of G21_MAKRO.spf,
                                            ; G1 and G90 are activated
                                            ; before the call of
                                            ; G21_MAKRO.spf
. . .
N0500 G90 X20 Y30 G123 G1 G54               ; Call of G123_MAKRO.spf,
                                            ; G1, G54 and G90 are activated
                                            ; before the call of
                                            ; G123_MAKRO.spf
. . .
N0800 G90 X20 Y30 G421 G1 G54               ; Call of G123_MAKRO.spf,
                                            ; G1, G54 and G90 are activated
                                            ; before the call of
                                            ; G123_MAKRO.spf
. . .
N0900 M30
PROC G21_MAKRO
. . .
N0010 R10 = R10 + 11.11
N0020 IF $C_X_PROG == 0
N0030 SETAL(61000)                          ; programmed variable not transferred cor-
                                            ; rectly
N0040 ENDIF
N0050 IF $C_Y_PROG == 0
N0060 SETAL(61001)
N0070 ENDIF
N0080 IF $C_F_PROG == 0
N0090 SETAL(61002)
N0100 ENDIF
N0110 G90 X=$C_X Y=$C_Y
N0120 G291
N0130 G21 M6 X100                           ; G21->activate metric measuring system
                                            ; (no macro call)
N0140 G290
. . .
N0150 M17
PROC G123_MAKRO
. . .
N0010 R10 = R10 + 11.11
N0020 IF $C_G == 421 GOTOF label_G421       ; Macro functionality for G123
N0040 G91 X=$C_X Y=$C_Y F500
. . .
. . .
N1990 GOTOF label_end
N2000 label_G421:                           ; Macro functionality for G421
N2010 G90 X=$C_X
Y=$C_Y F100
N2020
. . .
. . .
N3000 G291
```

```
N3010 G123                                   ; Alarm 12470, because G123 is not a
                                             ; G function and a
                                             ; macro call is not possible for active
                                             ; macro
                                             ;
                                             ; Exception: The macro was called
                                             ; as subroutine with CALL
                                               G123_MAKRO.
N4000 label_end: G290
N4010 M17
```

4.4.7 Special functions

4.4.7.1 Contour repetition (G72.1, G72.2)

A contour programmed once can be repeated easily with G72.1 and G72.2. This function can be used to create either a linear copy (G72.2) or a rotational copy (G72.1).

Format

G72.1 X... Y... (Z...) P... L... R...

X, Y, Z: Reference point for coordinate rotation

P: Subroutine number

L: Number of subroutine passes

R: Roll angle

A subroutine containing the contour to be copied can be called multiple times with G72.1. The coordinate system is rotated by a certain angle before calling each subroutine. The coordinate rotation is executed around a vertical axis on the selected plane.

G72.2 I... J... K... P... L...

I, J, K: Position to which the X, Y Z axes are traversed before calling the subroutine.

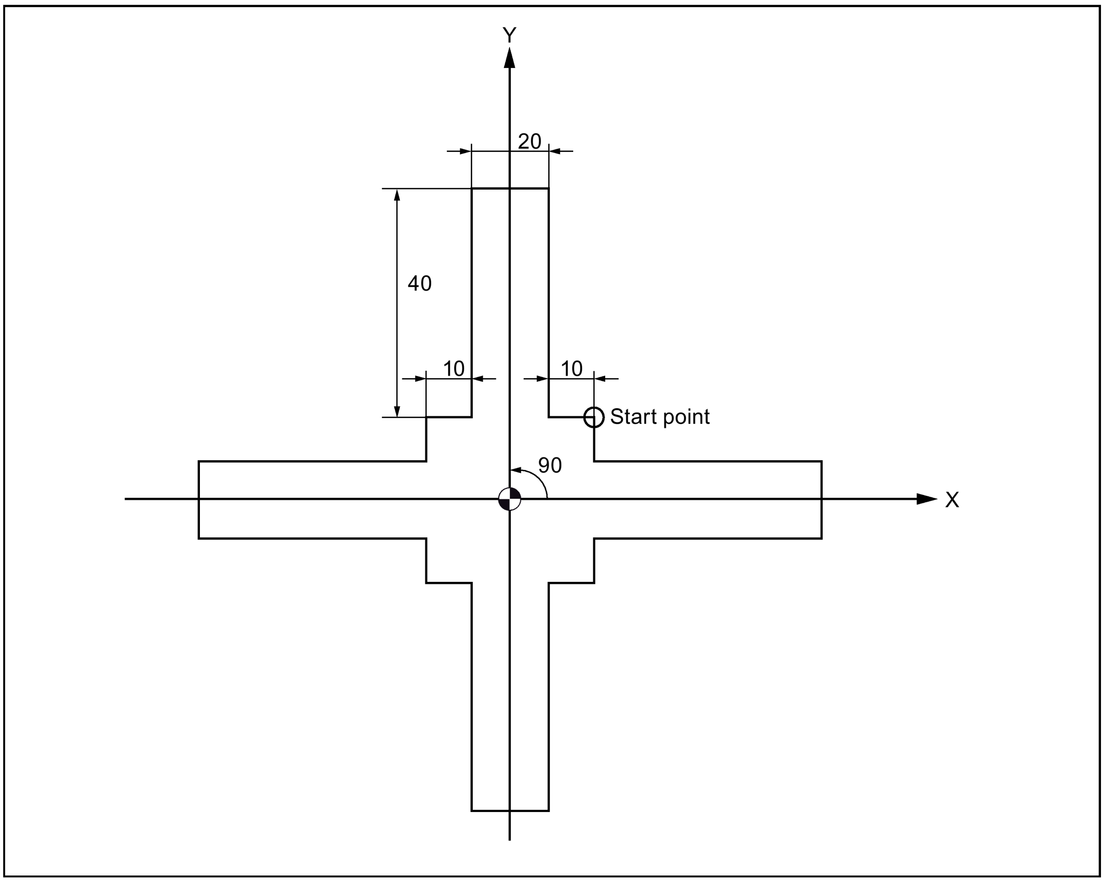
P: Subroutine number

L: Number of subroutine passes

A subroutine containing the contour to be repeated can be called multiple times with G72.2. The axes programmed with I, J and K must be traversed incrementally before each subroutine call. The cycle (CYCLE3721) is used to call the subroutine as often as is specified in Address "L". A distance programmed in I, J and K and calculated from the starting point is traversed before each subroutine call.

Examples

Contour repetition with G72.1:



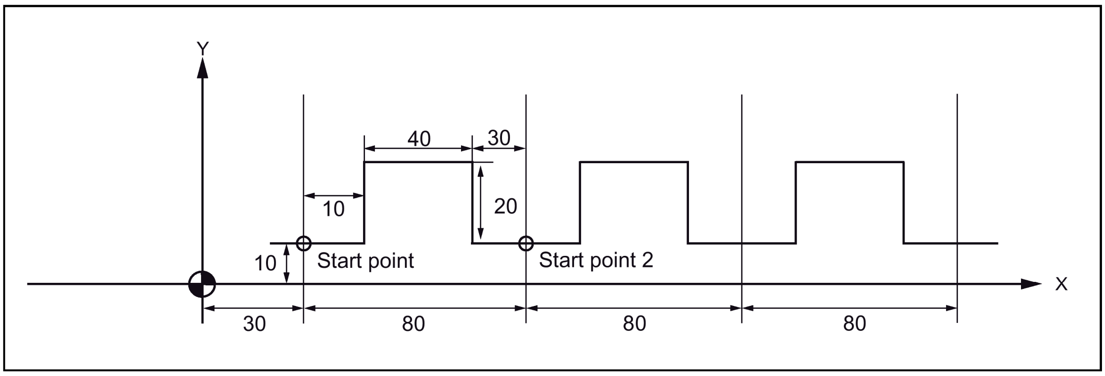
Main program

```
N10 G92 X40.0 Y50.0
N20 G01 G90 G17 G41 20 Y20 G43H99 F1000
N30 G72.1 P123 L4 X0 Y0 R90.0
N40 G40 G01 X100 Y50 Z0
N50 G00 X40.0 Y50.0 ;
N60 M30 ;
```

Subroutine 1234.spf

```
N100 G01 X10.
N200 Y50.
N300 X-10.
N400 Y10.
N500 X-20.
N600 M99
```

Contour repetition with G72.2:



Main program

```
N10 G00 G90 X0 Y0
N20 G01 G17 G41 X30. Y0 G43H99 F1000
N30 Y10.
N40 X30.
N50 G72.2 P2000 L3 I80. J0
```

Subroutine 2000.mpf

```
G90 G01 X40.
N100 Y30.
N200 G01 X80.
N300 G01 Y10.
N400 X110.
500 M99
```

4.4.7.2 Switchover modes for DryRun and skip levels

Changeover of the skip levels (DB3200.DBB2) always represents an intervention in the program run, which has led to a short-term drop in velocity on the path. The same is true of the changeover of the DryRun mode (DryRun = dry run feedrate DB3200.DBX0.6) from DryRunOff to DryRunOn or vice-versa.

All the drops in velocity can be avoided with a changeover mode that is limited in its function.

No drop in velocity is required with a setting machine data 10706 $MN_SLASH_MASK==2 while changing the skip levels (i.e., a new value in the PLC->NCK-Chan interface DB3200.DBB2).

Note

The NCL processes blocks in two steps, the preprocessing and main runs (also pre-travel and main run). The result of the premachining changes to the preprocessing memory. The main machining takes the relevant oldest block out of the preprocessing memory and traverses its geometry.

Note

The premachining is changeover with the setting machine data $MN_SLASH_MASK==2 during a change of the skip level! All blocks located in the preprocessing memory are traversed with the old skip level. The user normally does not have any control over the fill level of the preprocessing memory. The user can see the following effect: **A new skip level is effective "some time" after the changeover.**

Note

The part program command STOPRE vacates the preprocessing memory. If one switches the skip level before STOPRE, then all the blocks after STOPRE are changed over securely. The same is valid for an implicit STOPRE.

No drop in velocity is required while changing the DryRun mode with the setting machine data 10704 $MN_DRYRUN_MASK==2. Here too, only the premachining that leads to the above-mentioned restrictions, is switched. The following analogy is apparent from this: **This will also be active "sometime" after the changeover of the DryRun mode.**

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
GERMANY

Programming and Operating Manual (ISO Turning/Milling)
6FC5398-0DP40-0BA2, 06/2017