

# SIEMENS

## SINUMERIK

### SINUMERIK 840D sl / 828D ISO Turning

#### Programming Manual

<u>Fundamentals of programming</u>	<b>1</b>
<u>Travel commands</u>	<b>2</b>
<u>Traversing command</u>	<b>3</b>
<u>Additional functions</u>	<b>4</b>
<u>Abbreviations</u>	<b>A</b>
<u>G code table</u>	<b>B</b>
<u>Data Description</u>	<b>C</b>
<u>Data lists</u>	<b>D</b>
<u>Alarms</u>	<b>E</b>

Valid for

Control  
SINUMERIK 840D sl / 840DE sl  
SINUMERIK 828D

Software Version  
CNC software 4.5

**02/2012**

6FC5398-5BP40-3BA0

## 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.

 <b>DANGER</b>
indicates that death or severe personal injury <b>will</b> result if proper precautions are not taken.
 <b>WARNING</b>
indicates that death or severe personal injury <b>may</b> result if proper precautions are not taken.
 <b>CAUTION</b>
indicates that minor personal injury can result if proper precautions are not taken.
<b>NOTICE</b>
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:

 <b>WARNING</b>
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.

### Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

### Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

# Table of contents

<b>1</b>	<b>Fundamentals of programming .....</b>	<b>7</b>
1.1	Introductory remarks .....	7
1.1.1	Siemens mode .....	7
1.1.2	ISO dialect mode .....	7
1.1.3	Switching between the modes .....	8
1.1.4	Display of the G code.....	8
1.1.5	Maximum number of axes/axis identifiers.....	9
1.1.6	Define G code system A, B or C.....	9
1.1.7	Programming the decimal point .....	10
1.1.8	Comments.....	11
1.1.9	Skip block.....	12
1.2	Prerequisites for the feed.....	13
1.2.1	Rapid traverse.....	13
1.2.2	Path feed (F function) .....	13
1.2.3	Linear feed (G94).....	15
1.2.4	Inverse-time feed (G93).....	15
1.2.5	Revolutional feedrate (G95).....	15
<b>2</b>	<b>Travel commands .....</b>	<b>17</b>
2.1	Interpolation commands.....	17
2.1.1	Rapid traverse (G00) .....	17
2.1.2	Linear interpolation (G01) .....	20
2.1.3	Circular interpolation (G02, G03).....	21
2.1.4	Contour definition programming and insertion of chamfers and radii .....	24
2.1.5	Involute interpolation (G02.2, G03.2).....	26
2.1.6	Cylindrical interpolation (G07.1).....	27
2.1.7	Polar coordinates interpolation (G12.1, G13.1) (TRANSMIT) .....	29
2.2	Reference point approach with G functions.....	32
2.2.1	Reference point approach with intermediate point (G28).....	32
2.2.2	Checking the reference position (G27) .....	33
2.2.3	Reference point approach with reference point selection (G30) .....	34
2.3	Use of the thread cutting function .....	35
2.3.1	Thread cutting with constant lead (G33).....	35
2.3.2	Interlinking of threads (G33) .....	38
2.3.3	Cutting of multiple-start threads (G33).....	39
2.3.4	Cutting of thread with variable lead (G34) .....	41
2.3.5	Convex thread with G35 and G36 .....	42
<b>3</b>	<b>Traversing command .....</b>	<b>43</b>
3.1	The coordinate system.....	43
3.1.1	Machine coordinate system (G53).....	44
3.1.2	Workpiece coordinate system (G92) .....	45
3.1.3	Resetting the tool coordinate system (G92.1) .....	46
3.1.4	Selection of a workpiece coordinate system.....	46
3.1.5	Writing work offset/tool offsets (G10).....	47

3.2	Defining the input modes for the coordinate values.....	49
3.2.1	Absolute/incremental dimensioning (G90, G91) .....	49
3.2.2	Diameter and radius programming for the X-axis .....	52
3.2.3	Input inch/metric (G20, G21).....	53
3.3	Time-controlled commands.....	54
3.3.1	Dwell time (G04) .....	54
3.4	Tool offset functions .....	55
3.4.1	Tool offset data memory .....	55
3.4.2	Tool length compensation.....	56
3.4.3	Tool nose radius compensation (G40, G41/G42) .....	57
3.5	S-, T-, M- and B functions .....	62
3.5.1	Spindle function (S function) .....	62
3.5.2	Constant cutting rate (G96, G97).....	62
3.5.3	Tool change with T functions (T function).....	64
3.5.4	Additional function (M function).....	64
3.5.5	M functions of spindle control.....	66
3.5.6	M functions for subroutine calls .....	66
3.5.7	Macro call via M function.....	66
3.5.8	M functions .....	68
<b>4</b>	<b>Additional functions.....</b>	<b>69</b>
4.1	Program-support functions.....	69
4.1.1	Fixed cycles .....	69
4.1.2	Multiple repetitive cycles .....	79
4.1.3	Drilling cycle (G80 to G89).....	96
4.2	Programmable data input.....	108
4.2.1	Changing the tool offset value (G10) .....	108
4.2.2	M function for calling subroutines (M98, M99).....	109
4.3	Eight-digit program number.....	111
4.4	Measuring functions .....	113
4.4.1	Rapid lift with G10.6 .....	113
4.4.2	Delete measuring with distance-to-go (G31) .....	113
4.4.3	Measuring with G31, P1 - P4 .....	115
4.4.4	Interrupt program with M96/M97 (ASUB).....	115
4.5	Macro programs .....	119
4.5.1	Differences with subroutines .....	119
4.5.2	Macro program call (G65, G66, G67) .....	119
4.6	Special functions .....	126
4.6.1	G05.....	126
4.6.2	Multi-edge turning .....	126
4.6.3	Compressor in the ISO dialect mode .....	128
4.6.4	Switchover modes for DryRun and skip levels.....	129
4.6.5	Interrupt program with M96, M97 .....	130
<b>A</b>	<b>Abbreviations .....</b>	<b>133</b>
<b>B</b>	<b>G code table .....</b>	<b>141</b>

<b>C</b>	<b>Data Description .....</b>	<b>145</b>
	C.1 General machine/setting data .....	145
	C.2 Channel-specific machine data.....	160
	C.3 Axis-specific setting data .....	169
	C.4 Channel-specific setting data.....	170
<b>D</b>	<b>Data lists.....</b>	<b>173</b>
	D.1 Machine data.....	173
	D.2 Setting data .....	176
	D.3 Variables .....	177
<b>E</b>	<b>Alarms .....</b>	<b>179</b>
	E.1 Alarms .....	179
	<b>Glossary .....</b>	<b>183</b>
	<b>Index.....</b>	<b>211</b>



# Fundamentals of programming

## 1.1 Introductory remarks

### 1.1.1 Siemens mode

The following conditions are valid in the 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 the Siemens mode.

### 1.1.2 ISO dialect mode

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

- The ISO dialect mode can be set with machine data as the default setting of control system. The control system reboots by default in the ISO dialect mode subsequently.
- Only G functions from the ISO dialect can be programmed; the programming of Siemens G functions is not possible in the ISO Mode.
- Mixing of ISO dialect and Siemens language in the same NC block is not possible.
- Switching between ISO Dialect M and ISO Dialect T with a G command is not possible.
- Subroutines that are programmed in the Siemens mode can be called.
- If Siemens functions are to be used, one must first switch to the Siemens mode.

### 1.1.3 Switching between the modes

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

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

The active tool, the tool offsets and work offsets are not influenced by the switchover.

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

### 1.1.4 Display of the G code

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

#### Example

The G functions of the 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 the ISO dialect language continue to be displayed.

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

#### Procedure

The Siemens shell cycles are called via main programs. The 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 the 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.

### **1.1.5 Maximum number of axes/axis identifiers**

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

### **1.1.6 Define G code system A, B or C**

In the ISO-dialect-T, a differentiation is made between the G code system A, B or C. By default, the G code system B 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 not available. In this case, an incremental axis movement with the address characters U, V and W is programmed for the axis X, Y and Z. The address characters U, V and W are not available in this case as axis identifiers, 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.

---

#### **Note**

Unless otherwise specified, the G code system B is excluded in this documentation.

---

### 1.1.7 Programming the decimal point

In the 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.

The setting is done via 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.

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 Rotate

Table 1- 1 Different conversion factors for IS-B and IS-C

Address	Unit	IS-B	IS-C
Linear axis	mm	0.001	0.0001
	inch	0.0001	0.00001
Rotary axis	Degree	0.001	0.0001
F Feed G94 (mm/inch per min.)	mm	1	1
	inch	0.01	0.01
F Feed G95 (mm/inch per rotation) \$MC_EXTERN_FUNCTION_MASK			
Bit8 = 0	mm	0.01	0.01
	inch	0.0001	0.0001
Bit8 = 1	mm	0.0001	0.0001
	inch	0.000001	0.000001
F Thread lead	mm	0.0001	0.0001
	inch	0.000001	0.000001
C Chamfer	mm	0.001	0.0001
	inch	0.001	0.0001

Address	Unit	IS-B	IS-C
R Radius, G10 toolcorr	mm inch	0.001 0.001	0.0001 0.0001
I, J, K IPO-Parameter	mm inch	0.001 0.001	0.0001 0.0001
G04 X or U		0.001	0.001
A Angle contour definition		0.001	0.001
G76, G78 tapping 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	mm inch	0.01 0.0001	0.01 0.0001
Bit8 = 1 G95 F	mm inch	0.0001 0.000001	0.0001 0.000001

### 1.1.8

### Comments

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

### 1.1.9 Skip block

The sign of skipping or suppression of blocks "/" can be used at any convenient position in a block, i.e. 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.

---

## 1.2 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.

### 1.2.1 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.

### 1.2.2 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 two images).

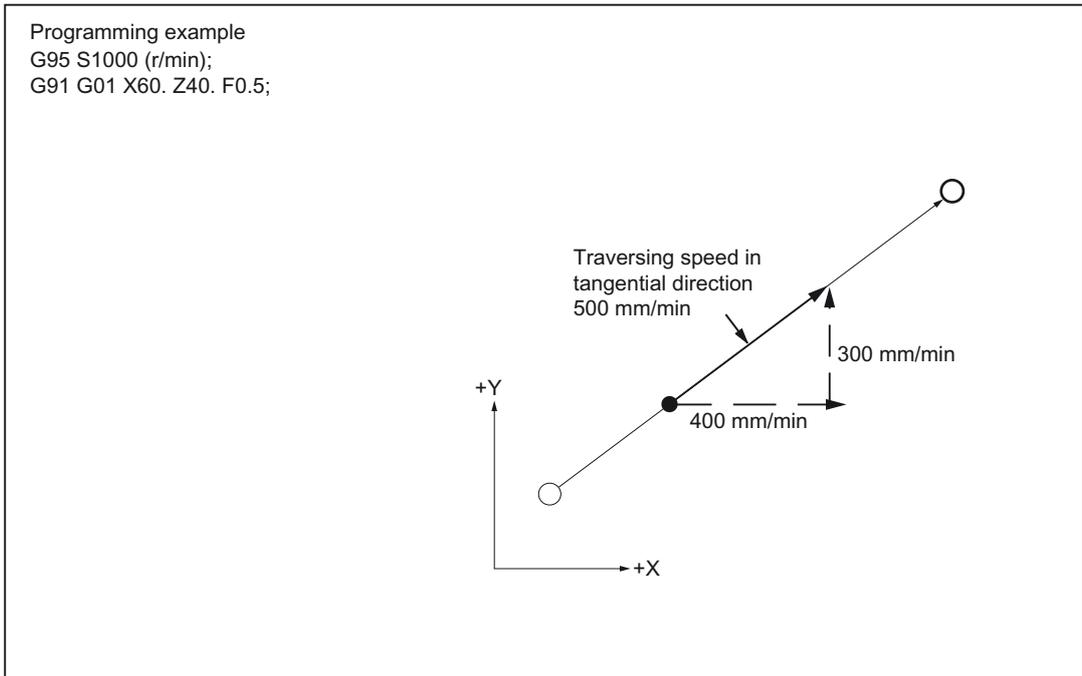


Figure 1-1 Linear interpolation with 2 axes

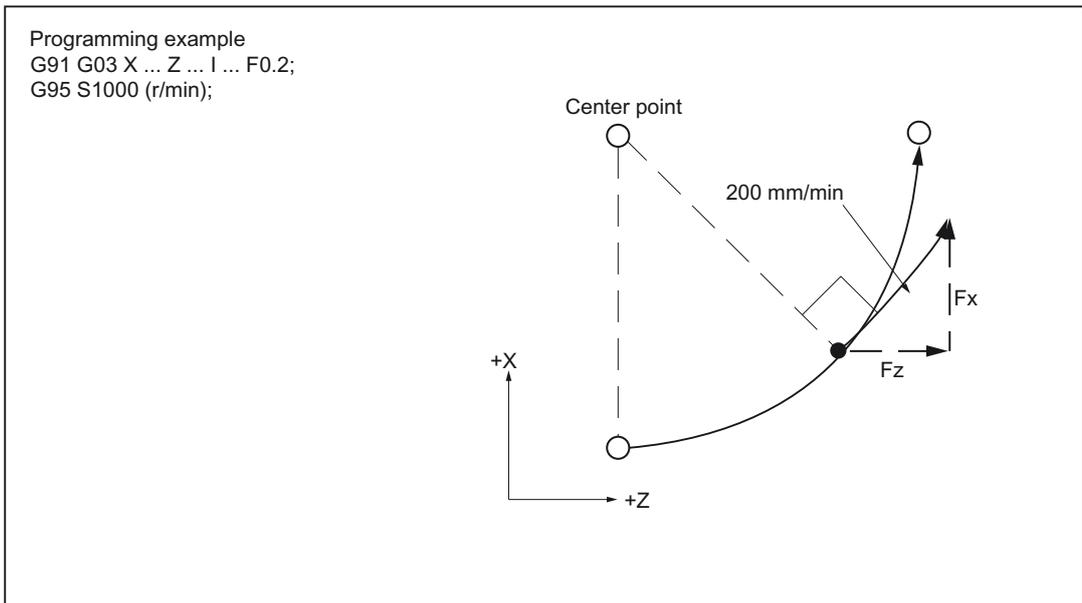


Figure 1-2 Circular interpolation with 2 axes

**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.

### 1.2.3 Linear feed (G94)

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

### 1.2.4 Inverse-time feed (G93)

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.

### 1.2.5 Revolutional feedrate (G95)

On entering G95, 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 among G93, G94 or G95, the path feed must be reprogrammed. The feed can also be specified in degree/revolution for the machining with rotary axes.

---



## Travel commands

### 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.

#### 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 (see the table below):

Table 2- 1 G functions 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... Y... 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, then 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.

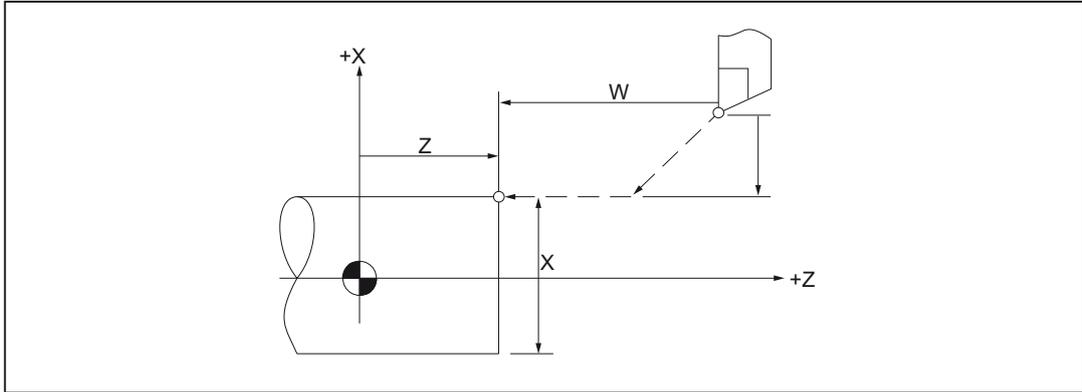


Figure 2-1 Rapid traverse with 2 non-interpolating axes

#### 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.

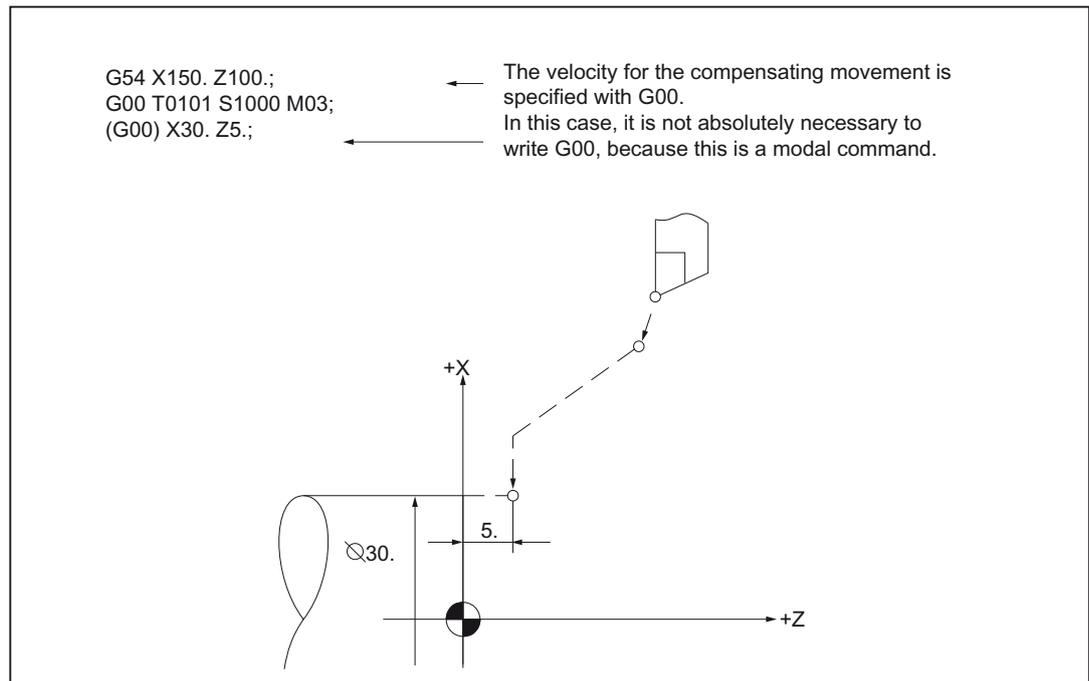


Figure 2-2 Programming example

### Linear interpolation (G00)

Linear interpolation with G00 is set by setting the machine data 20732 \$MC\_EXTERN\_GO\_LINEAR\_MODE. Hence all programmed axes traverse in rapid traverse with linear interpolation and reach their target position simultaneously.

### 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 (G93, G94, G95) 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 the Chapter "Absolute/incremental dimensioning".

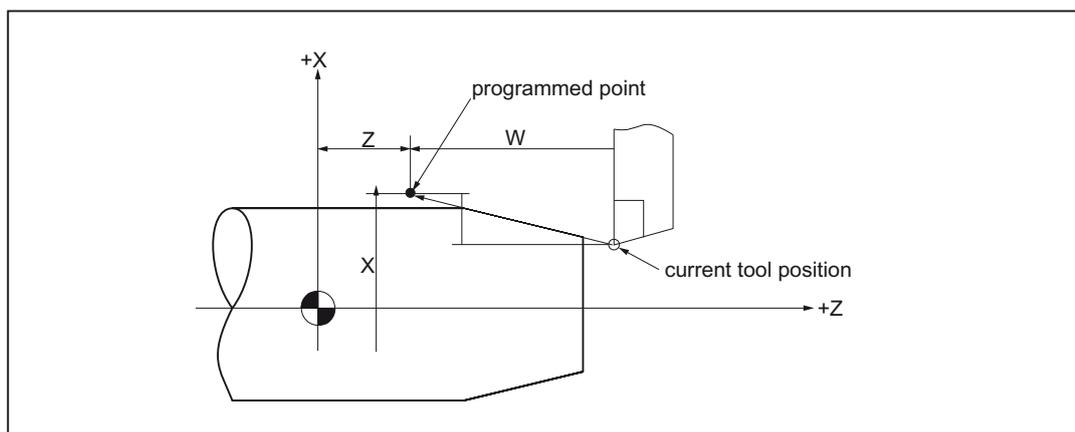


Figure 2-3 Linear interpolation

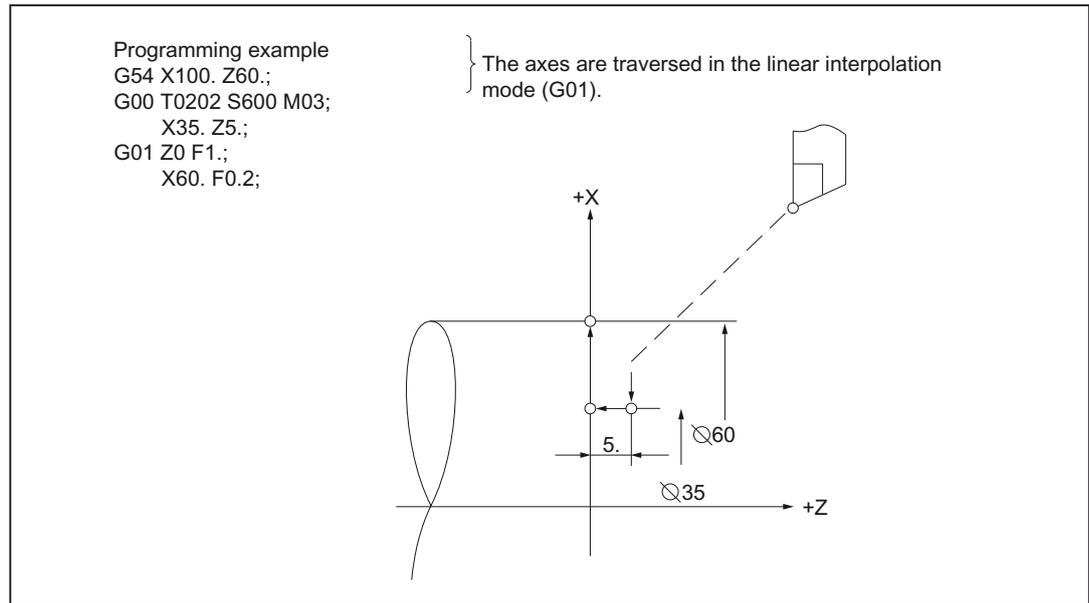


Figure 2-4 Programming example

### 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... ;

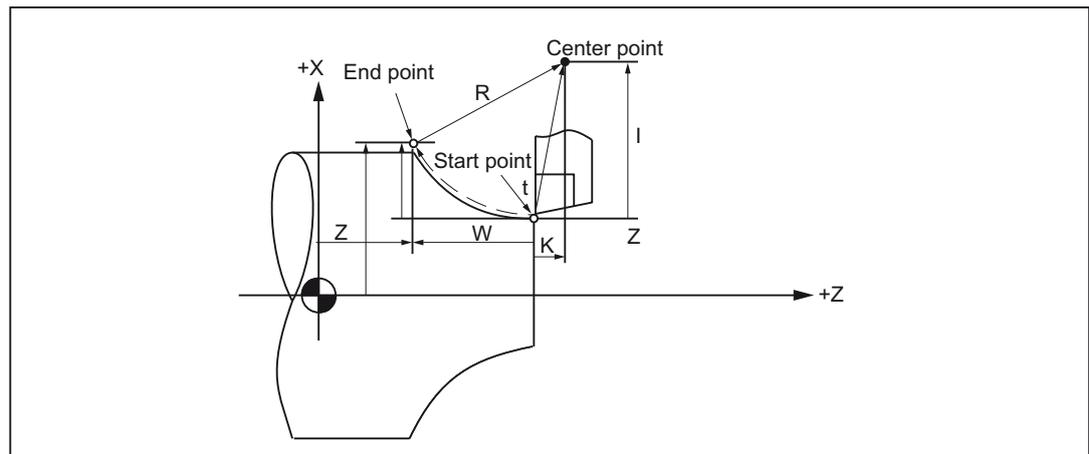


Figure 2-5 Circular interpolation

2.1 Interpolation commands

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

Table 2- 2 Commands for executing the circular interpolation

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

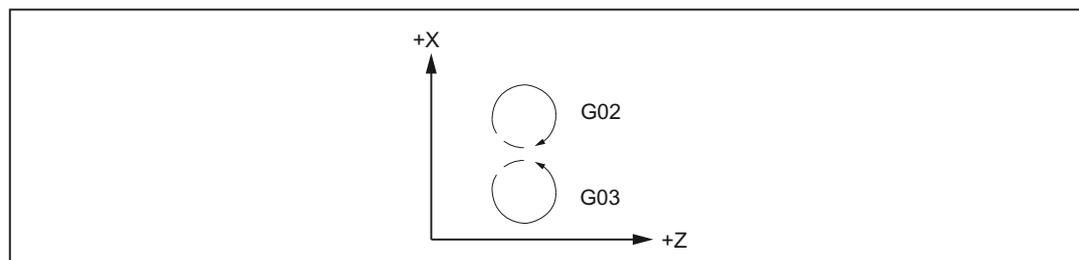


Figure 2-6 Direction of rotation of the arc

## End point

The end point can be specified with G90 or G91 either as absolute or as incremental value.

## Programming of circular movements

The 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  $\leq 180$  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.

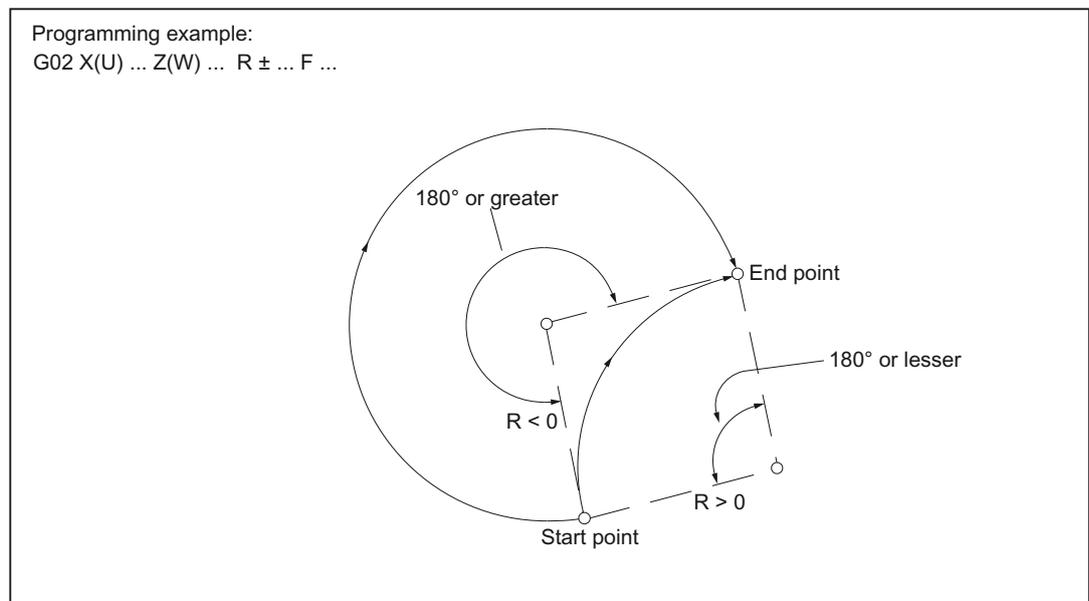


Figure 2-7 Circular interpolation with the specification of the radius R

## Feed

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

Programming example

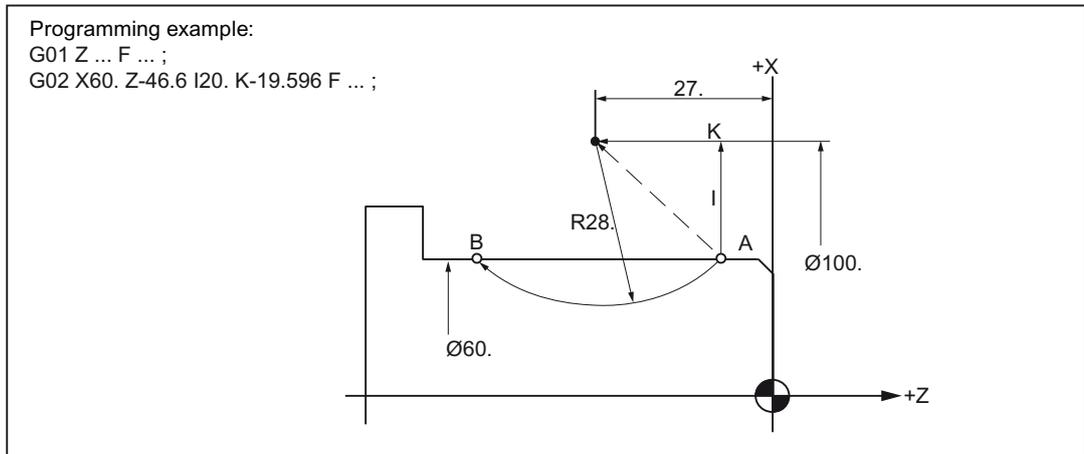


Figure 2-8 Circular interpolation over several quadrants

Center of arc	(10000, 2700)
Value of "I"	$\frac{100 - 60}{2} = 20 \text{ mm}$
Value of "K"	$-\sqrt{28^2 - 20^2} = -\sqrt{384} = -19.596 \text{ mm}$

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 X10. Z100. F1000 G18  
N20 A140 C7.5  
N30 X80. Z70. A95.824, R10
```

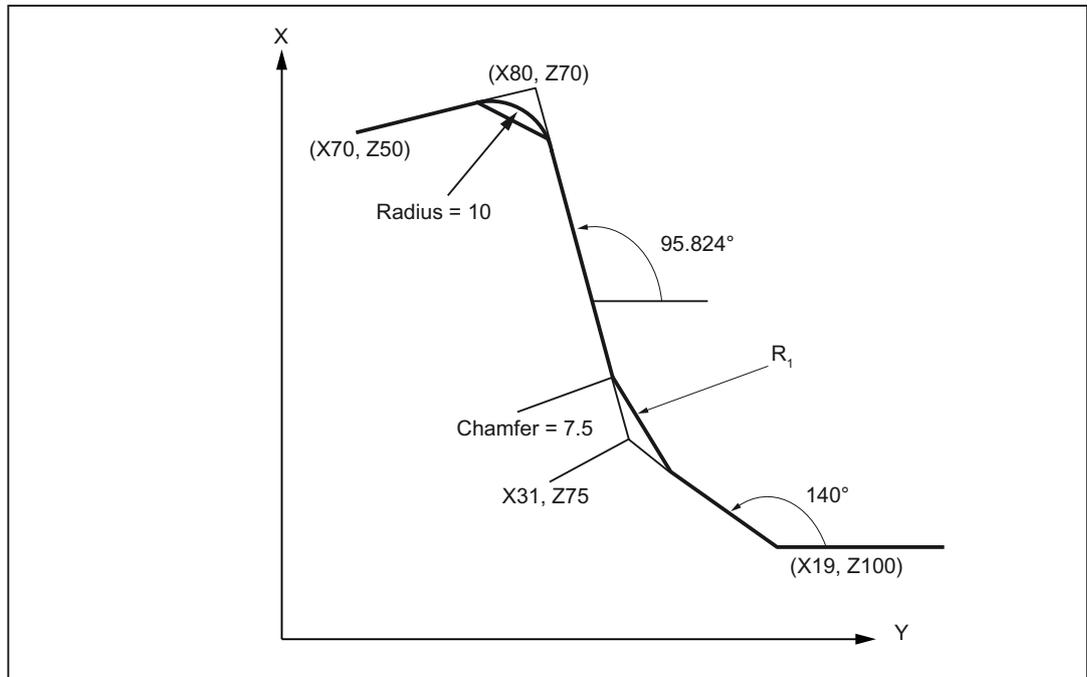


Figure 2-9 3 straight lines

### 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.

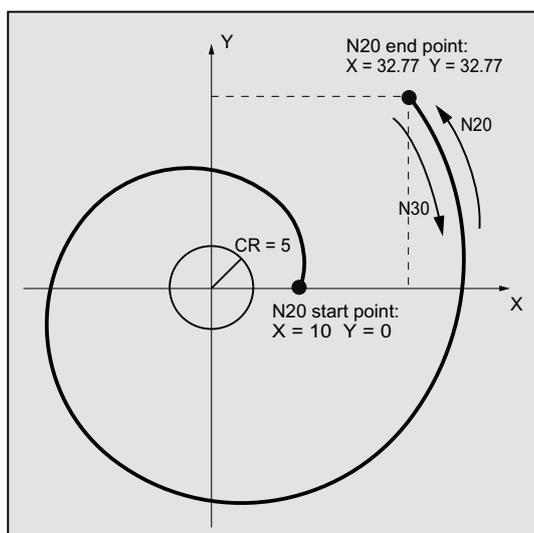
The address R can either be a cycle parameter or an identifier for the radius in a contour.

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

### 2.1.5 Involute interpolation (G02.2, G03.2)

#### Overview

The involute of a circle is a curve traced out from the end point on a "piece of string" unwinding from the curve. The involute interpolation allows trajectories along an involute. It is executed in the plane in which the base circle is defined. If the starting point and the end point are not in this plane, then, analogous to the helical interpolation for circles, there is a superimposition to a curve in space.



An involute can be traversed in space in case of additional specification of paths vertical to the active plane.

#### Format

G02.2 X... Y... Z... I... J... K... R

G03.2 X... Y... Z... I... J... K... R

- G02.2: Travel on an involute in clockwise direction
- G03.2: Travel on an involute in counterclockwise direction
- X Y Z: End point in Cartesian coordinates
- I J K: Center of the base circle in Cartesian coordinates
- R: Radius of the base circle

### Supplementary conditions

Both the starting point and the end point must be outside the area of the base circle of the involute (circle with radius R around the center point specified by I, J, K). If this condition is not satisfied, an alarm is generated and the program execution is aborted.

---

#### Note

For more information about machine data and supplementary conditions that are relevant to involute interpolation, please see References: /FB1/, A2 Chapter "Settings for involute interpolation".

---

## 2.1.6 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:

Table 2- 3 G functions for switching the cylindrical interpolation on/off

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 manual "Extended Functions", Chapter M1, "TRACYL".

The rotary axis for the cylindrical interpolation and also its name is defined with the machine data 24120 \$MC\_TRAFO\_GEOAX\_ASSIGN\_TAB\_1.

**Example**

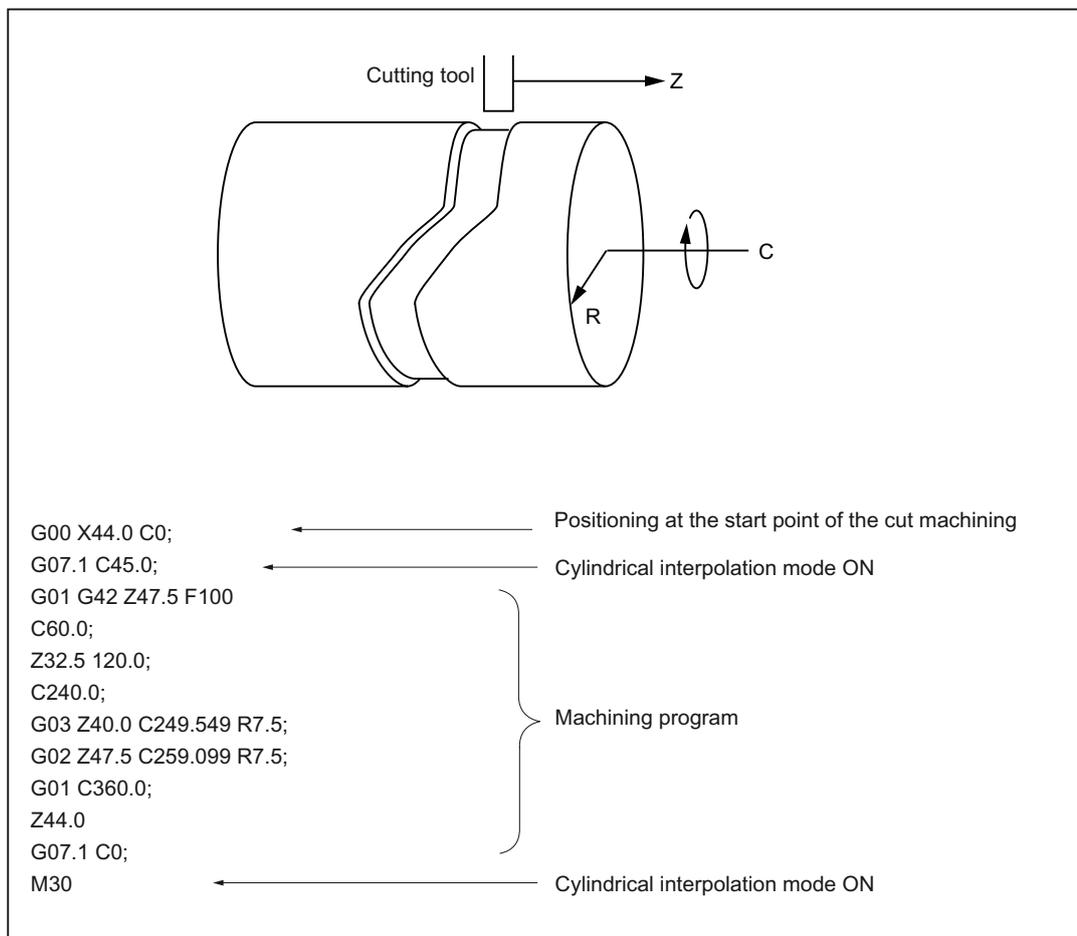


Figure 2-10 Programming example for the cylindrical interpolation

## 2.1.7 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 the 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.

Table 2- 4 G functions for switching the polar coordinates interpolation on/off

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 for as long as 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 the 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 the 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**

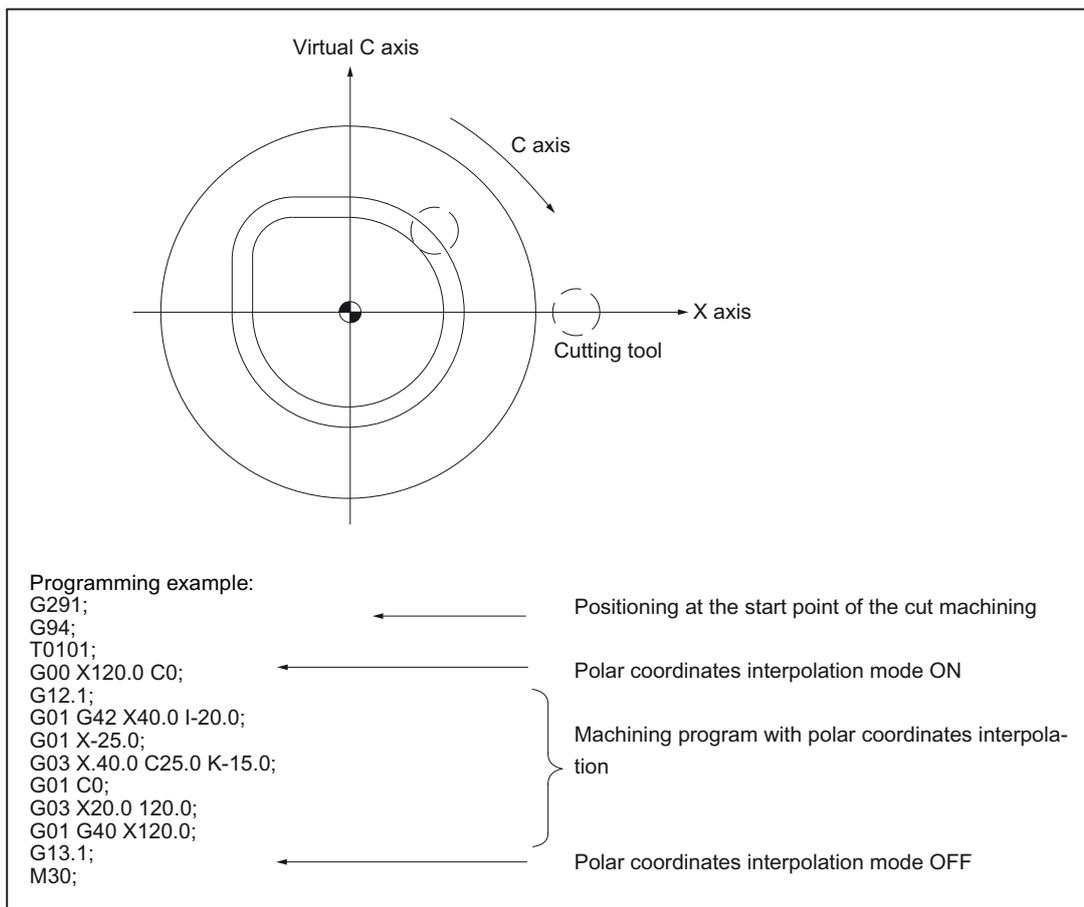


Figure 2-11 Coordinates system for the polar coordinates interpolation

For further information refer to

**References:**

Function Manual Extended Functions, Chapter TRANSMIT.

## 2.2 Reference point approach with G functions

### 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 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 set with the machine data 34100 \$ \_MA\_REFP\_SET\_POS[0] to [3]. A total of four reference positions can be defined.

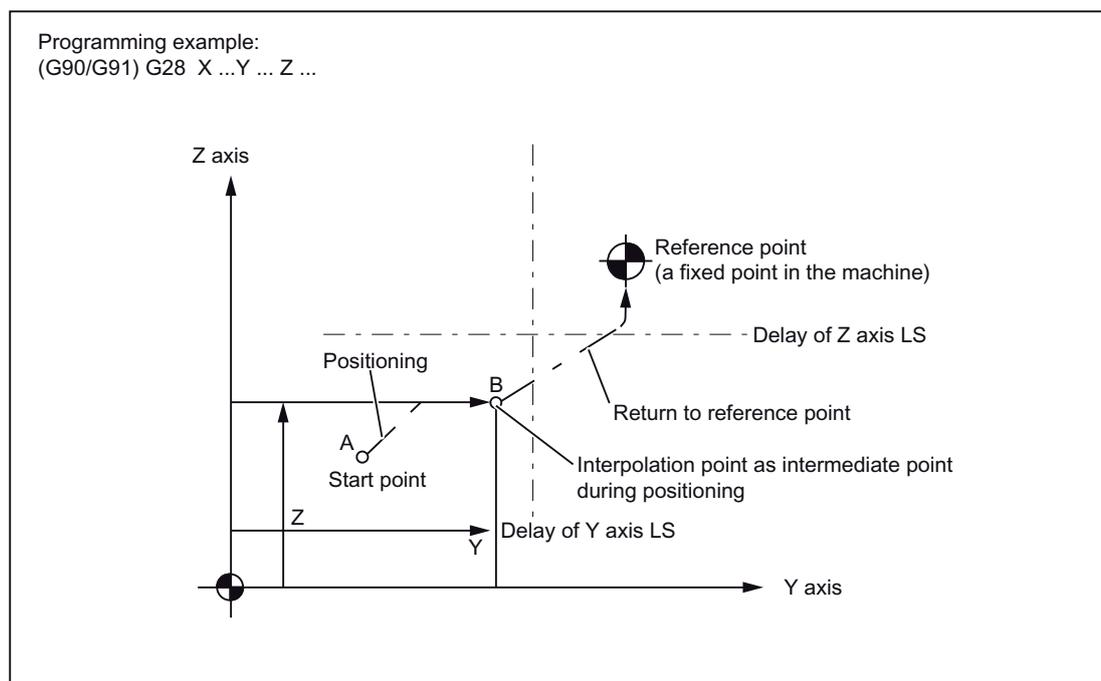


Figure 2-12 Automatic reference point approach

---

**Note**

The G28 function 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 with command TRAFOOF.

---

## 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 the Automatic mode is interrupted.

---

**Note**

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

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

---

### 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:

Table 2- 5 Reference points

Element	MD
2nd reference point	\$_MA_REFP_SET_POS[1]
3rd reference point	\$_MA_REFP_SET_POS[2]
4th reference point	\$_MA_REFP_SET_POS[3]

#### Note

Further details of the points that were considered in the programming of G30 can be found in Section "Reference point approach with intermediate point (G28)". Function G30 is implemented with cycle 330.spf.

## 2.3 Use of the thread cutting function

### 2.3.1 Thread cutting with constant lead (G33)

#### Format

With the commands "G33 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

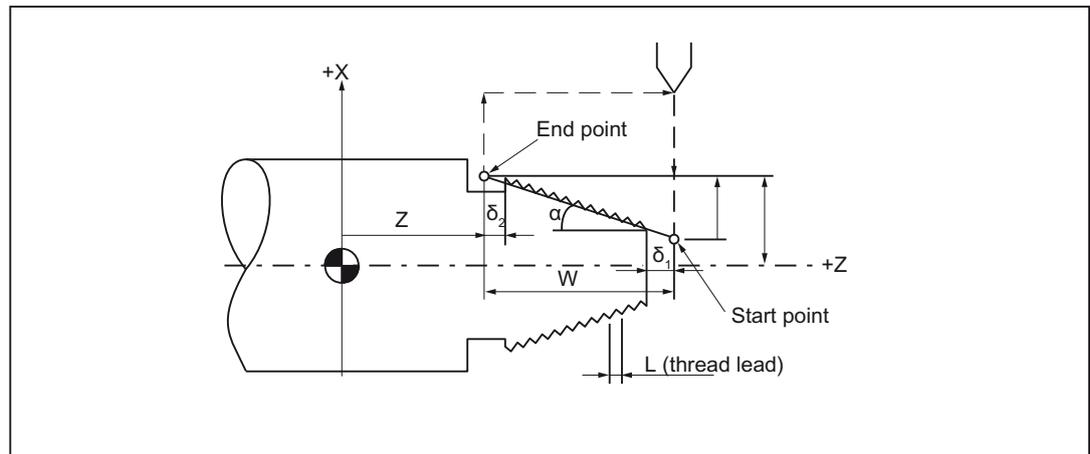


Figure 2-13 Thread cutting

#### Direction of thread lead

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

Table 2- 6 Direction of thread lead

		Direction of thread lead
	$\alpha \leq 45^\circ$	The programmed thread lead acts in the direction of the Z-axis.
	$\alpha > 45^\circ$	The programmed thread lead acts in the direction of the X-axis.

2.3 Use of the thread cutting function

Example

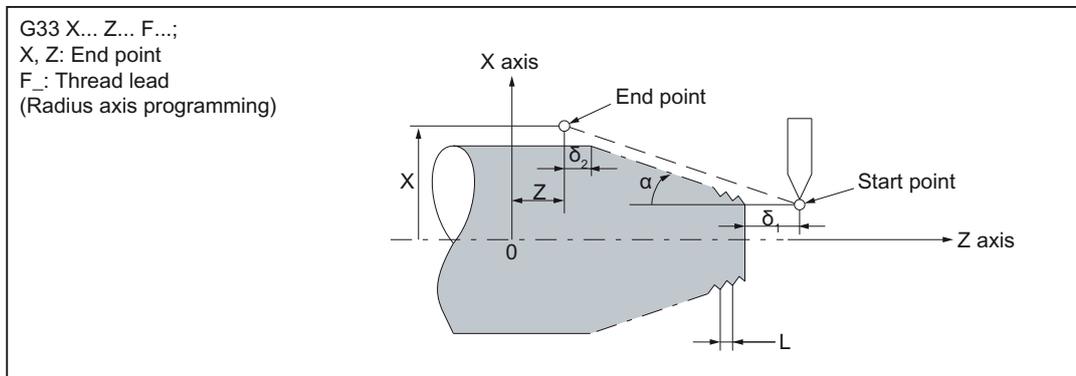


Figure 2-14 Examples of programming

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

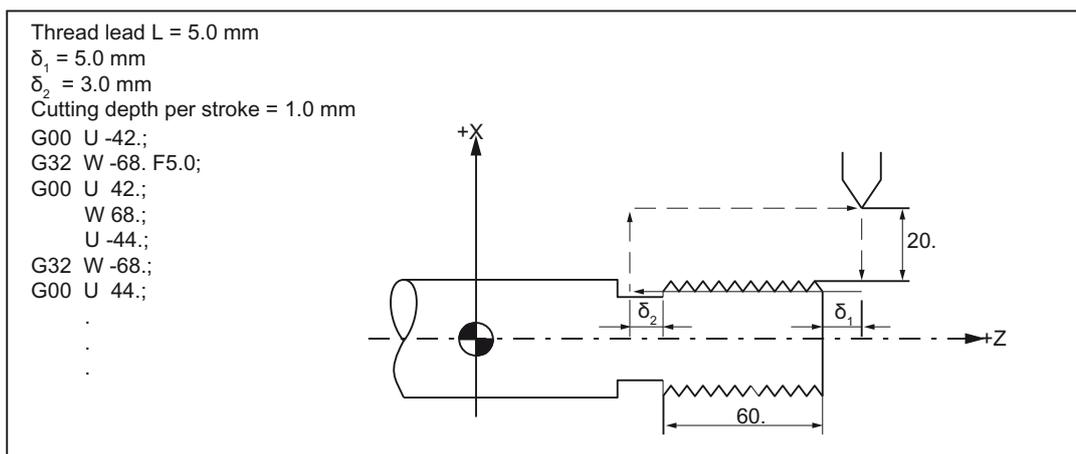


Figure 2-15 Programming example for cutting a cylindrical thread

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

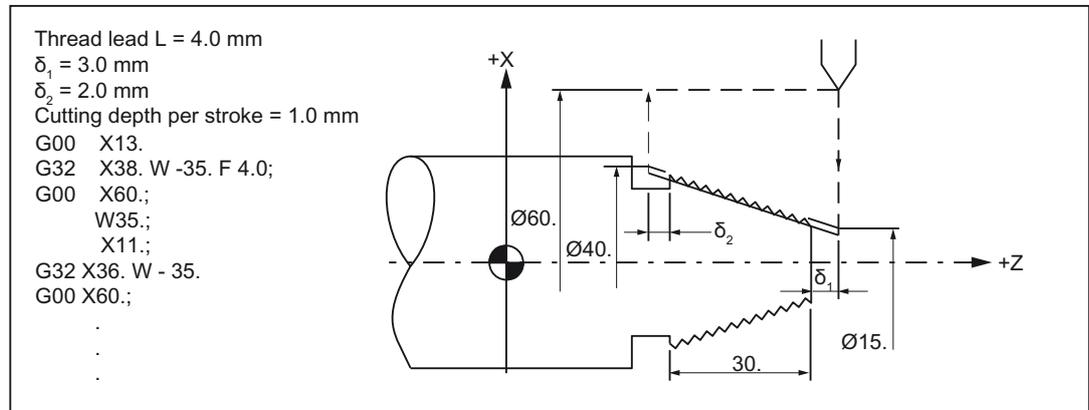


Figure 2-16 Programming example for cutting a taper thread

#### 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 G33, the limitation to maximum axis velocity (rapid traverse) is monitored by the control system.

### 2.3.2 Interlinking of threads (G33)

The thread blocks can be arranged one after the other to make a chain via several G33 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,

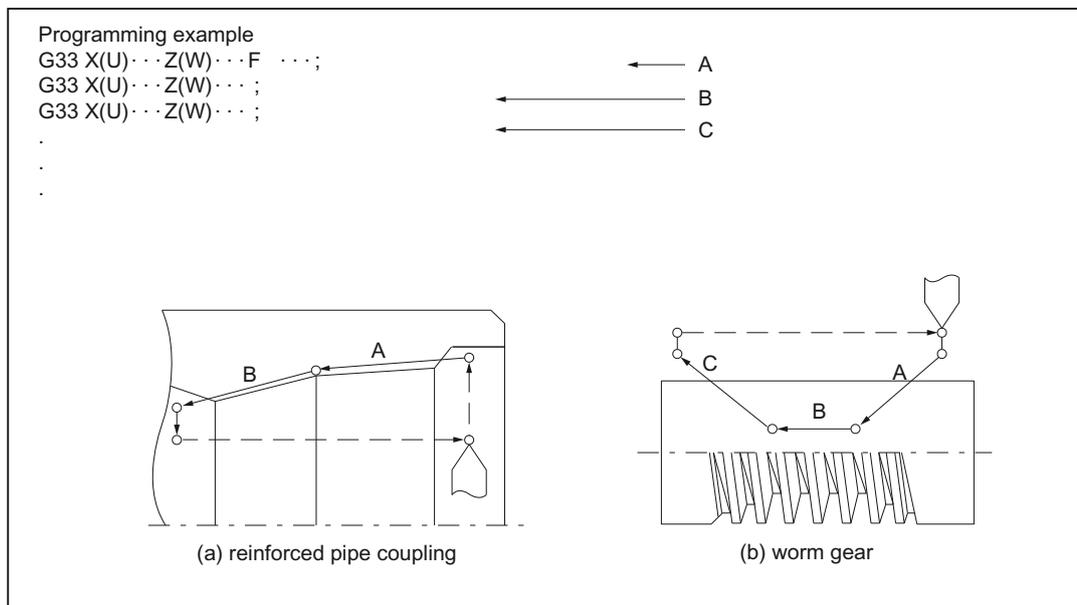


Figure 2-17 Cutting a continuous thread

---

**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 G33 is programmed when operating with G94 (feed per minute), an alarm is given off.

---

### 2.3.3 Cutting of multiple-start threads (G33)

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.

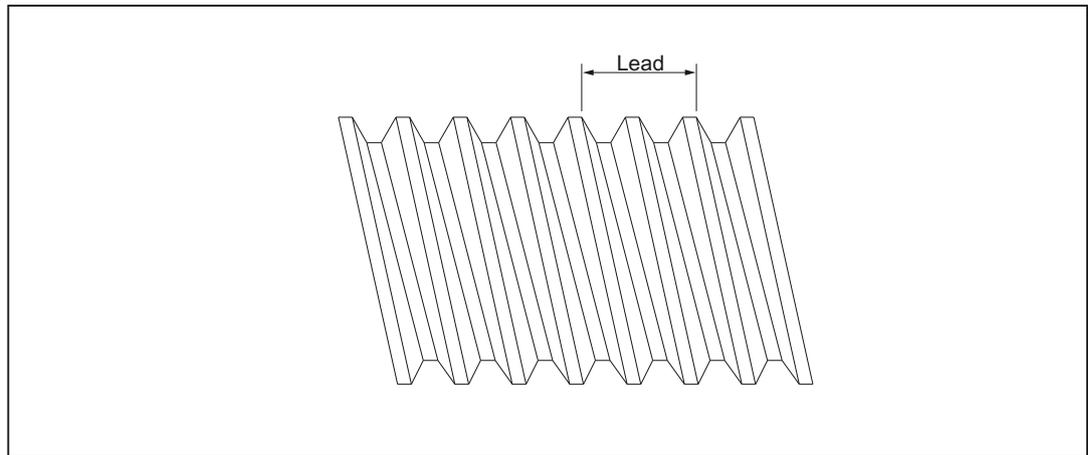


Figure 2-18 Double-start thread

#### Format

With the commands "G33 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:  $0 \leq B < 360.000$

**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.

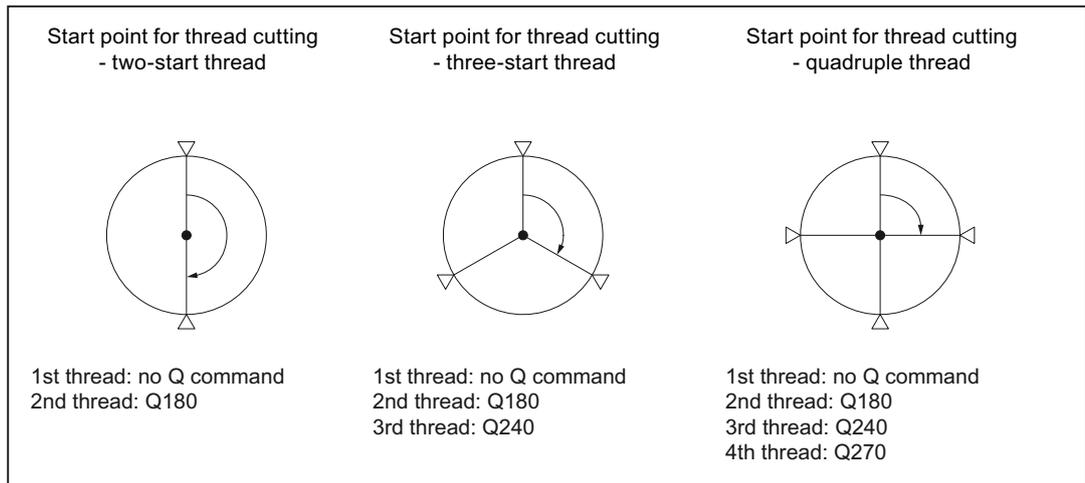


Figure 2-19 Calculation of the start angle in case of multiple-start threads

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

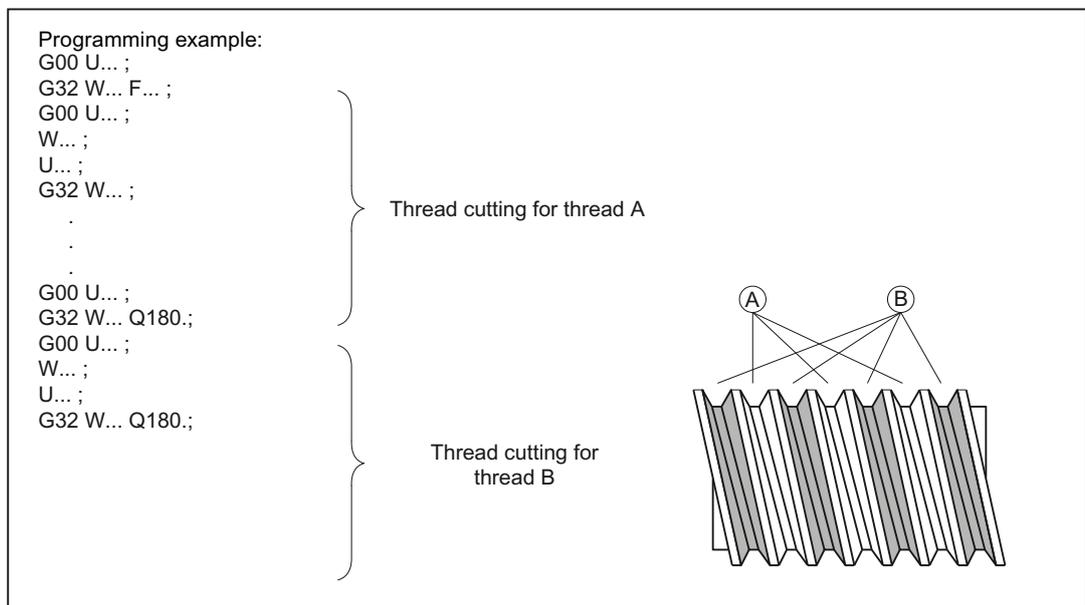


Figure 2-20 Specification of the spindle revolution angles

**Note**

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

### 2.3.4 Cutting of thread 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

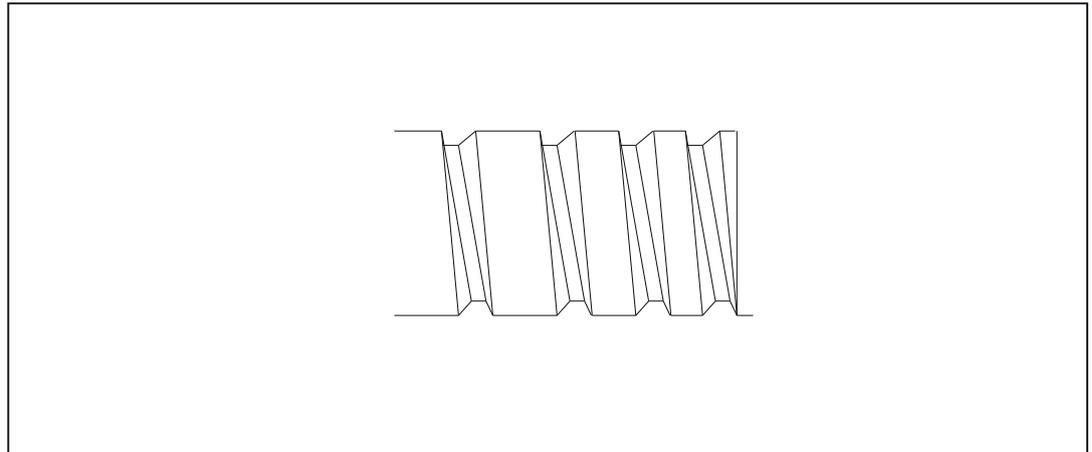


Figure 2-21 Thread with variable thread lead

**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!

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

### 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:

$$F = \frac{|k2e - k2a|}{2 * |G[\text{mm/rev}]}$$

The identifiers have the following meanings:

Ke: Thread lead of axis target point coordinate in [mm/rev]

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

IG: Thread length in [mm]

### 2.3.5 Convex thread with G35 and G36

In ISO dialect T, convex threads are programmed with G codes G35 in the clockwise direction and G36 in the counter-clockwise direction.

These G codes are available for the G code systems A, B and C. The arc can be programmed with specification of the circle center point via the circle parameters I, J, K or via specification of the radius R. Optionally, a thread starting point offset can be specified with Q. If R is programmed together with I, J, K, only R is effective.

Programming:

X.. Y.. Z.. I.. J.. K.. R.. F.. Q..

X Y Z	End point of the arc
I J K	Coordinates for circle center point programming
R	Radius for radius programming
F	Lead in the direction of the main axis
Q	Thread starting point offset (optional)

Convex threads are described in detail in SINUMERIK 840D sl / 828D Fundamentals Programming Manual, Section 9 "Motion commands".

## Traversing command

### 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...

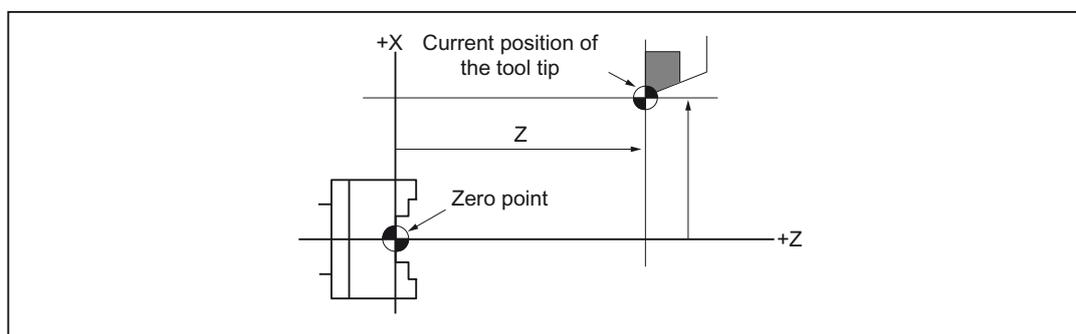


Figure 3-1 Tool position specified with X... Z..

The following coordinate systems are used for specifying the coordinates:

1. Machine coordinate system (G53)
2. Workpiece coordinate system (G92)
3. Local coordinate system (G52)

### 3.1.1 Machine coordinate system (G53)

#### Define machine coordinate system

The machine coordinate system MKS 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.

This is not necessary, if an absolute measuring system is used.

#### Format

(G90) 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.

#### Reference

With MD24004 \$MC\_CHBFRAME\_POWERON\_MASK, for bit 0 it is defined, whether the channel-specific basic frame is reset upon Power On.

### **3.1.2 Workpiece coordinate system (G92)**

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:

1. with G92 (G50 in the G code system A)
2. manually via the HMI operator panel

#### **Format**

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

#### **Explanation**

With G92 a coordinate transformation is programmed from the basic coordinate system (BCS) into basic zero-point system (BZS). G92 acts like a adjustable work offset.

### 3.1.3 Resetting the tool coordinate system (G92.1)

With G92.1 X.. (G Code System A with G50.3 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. 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
```

### 3.1.4 Selection of a workpiece coordinate system

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

#### 1. G92

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

#### 2. 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 and G54 P{1...100}.

Workpiece coordinate systems are setup after the reference point approach after Power On. The closed position of the coordinate system is G54.

### 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.

1. Data inputting at HMI operator panel
2. with the program commands G10 or G92 (setting actual value, spindle speed limitation)

#### 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.

G10 L20 Pp X... Y... Z... ;

p=1 to 93: The value of the workpiece work offset corresponds to the workpiece coordinate system G54 P1 ... P93. The number of work offsets (1 to 93) can be set through MD18601 \$MN\_MM\_NUM\_GLOBAL\_USER\_FRAMES or MD28080 \$MC\_MM\_NUM\_USER\_FRAMES.

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 STPPRE 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.

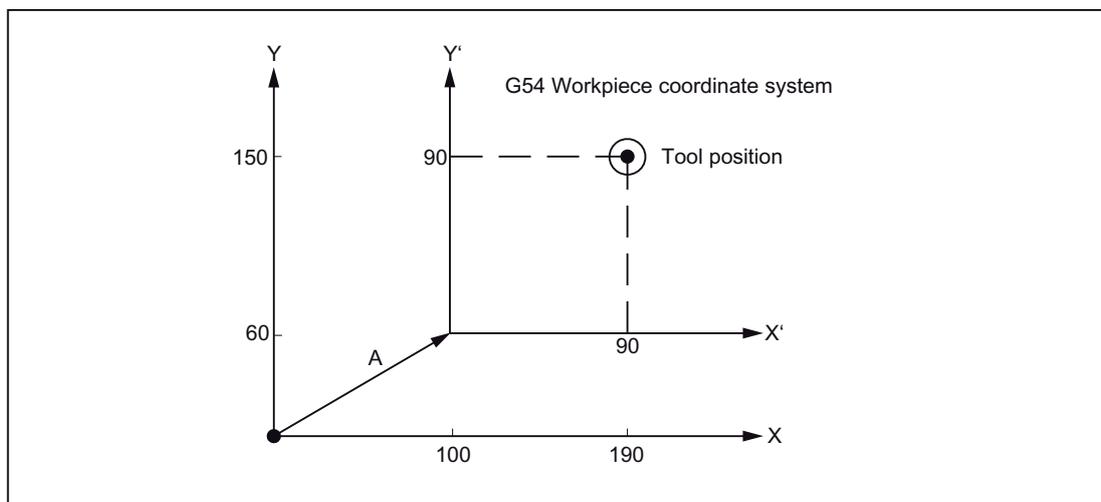


Figure 3-2 Example of setting coordinates

## 3.2 Defining the input modes for the coordinate values

### 3.2.1 Absolute/incremental dimensioning (G90, G91)

With these G commands it is specified, whether the dimensions are entered as absolute or incremental after an axis address.

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

The values programmed with the addresses X, Z, C, Y, U, W, H or V are interpreted in the G code system B and C, depending upon G90 or G91, either as absolute or incremental positions. The G functions G90 and G91 are not present in the G code system A. In this G code system the absolute axis positions are programmed with the addresses X, Y, Z, C and the incremental positions with U, V, W and H.

#### Command format for the G code system A

- Absolute axis positions

Absolute axis positions are programmed with the addresses X, Z and C.

Example: X10 Z100. C20 ;

- Incremental axis positions

Incremental axis positions are programmed with the addresses U, W and H

Example: U5 W3.9 H4 ;

- Incremental and absolute axis positions may be programmed mixed in the same block.

Example: X10 W3 ;

U5 Z100 ;

It is permissible to use the incremental and the absolute values in the one and the same block.

Example: X... W... ;

U... Z... ;

If the addresses that act on the same axis are programmed several times in the block, then the value last programmed, e.g. in the shape "X100 U15 ;" is effective, the X-axis is traversed incrementally by 15 mm, the position is skipped with X100.

Table 3- 1 Absolute/incremental dimensioning and its importance

Address	Command value		Meaning (description)
X	absolute	Diameter value	Position in the direction of X-axis
Z		-	Position in the direction of Z-axis
C		-	Position in the direction of C-axis
Y		-	Position in the direction of Y-axis
U	incremental value	Diameter value	Traversing path in the direction of X-axis
W		-	Position in the direction of Z-axis
H		-	Traversing path in the direction of C-axis
V		-	Traversing path in the direction of Y-axis
I	incremental value	Radius value	Distance of the start point to the center point of the arc in the X-axis
K		-	Distance of the start point to the center point of the arc in the Z-axis
J		-	Distance of the start point to the center point of the arc in the Y-axis
R	incremental value	-	Radius of arc

Since the addresses X and U are programmed for the diameter value, the actual axis movement corresponds to half of the specified values.

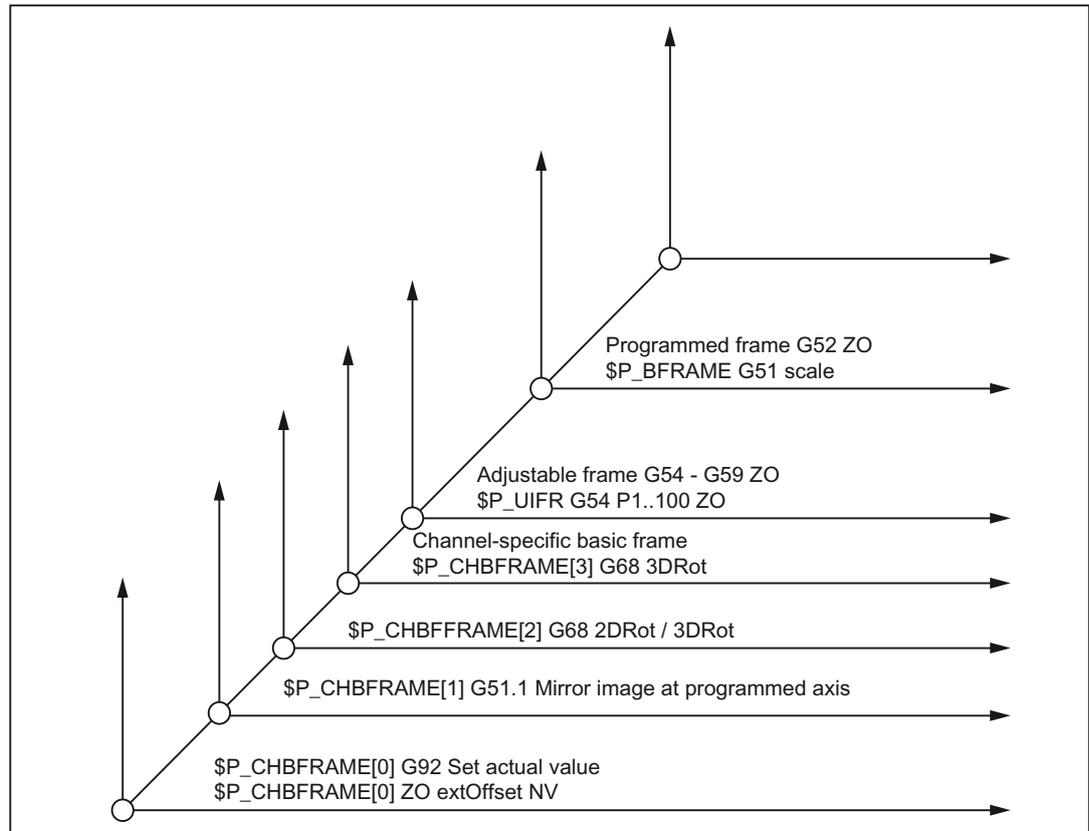


Figure 3-3 Absolute and incremental coordinate values

## Use of G90 and G91 (G code system B and C)

Table 3- 2 Mode of operation of the commands G90 and G91

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

Table 3- 3 Valid addresses for the programming of G90/G91

Address	Command G90	Command G91
	absolute	Incremental
Example: With the commands "G91 G00 X40. Z50.;" the axis positions are traversed incrementally.		

**Circular interpolations parameter**

Circular interpolation parameters I, J, K and the radius R are always interpreted as incremental values.

**Note**

G90 and G91 should not be programmed together in one block; else, only the G function programmed last becomes effective. Thus assumed, if the commands "G01 G90 X80. G91 Z60.;" are programmed in a block, then G91 becomes effective, because it has been programmed last, and all axis positions (X80. and Z60.) are interpreted as incremental paths.

**3.2.2 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.

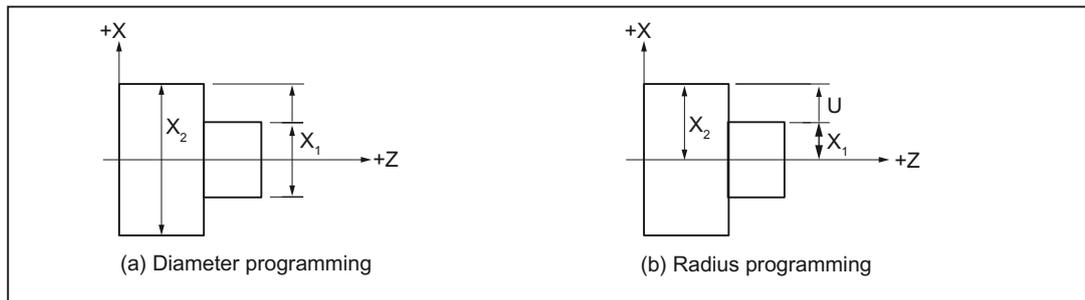


Figure 3-4 Coordinate values

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

**References:**

/PGA/, Programming Manual Advanced, Section "Special movement commands and movement synchronizations"

### 3.2.3 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:

Table 3- 4 G functions for selecting the unit of measurement

G function	Function	G group
G20 (G70, G-Codesyst. C)	Input in "inch"	06
G21 (G71, G-Code syst. 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, Y, Z
- Interpolation parameters I, J, K and circle radius R of the circle radius programming
- Thread lead (G33, 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.

#### References:

/FB1/Function Manual Basic Functions; Speeds, Setpoint/Actual-Value System, Closed-Loop Control (G2), Section "Metric/Inch Measuring System"

Table 3- 5 Tool offset amounts while working with G20 or G21

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 Time-controlled commands

### 3.3.1 Dwell time (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 G94 the dwell time is interpreted in seconds; if G95 is selected, the dwell time is specified as spindle revolutions [rev].

#### 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 (G94 active) or in spindle revolutions (G95 active)

The processing of the next block is delayed in case of G94 (feed per minute) by a certain time (in seconds) and in case of G95 (revolutional feedrate) a certain number of spindle revolutions are awaited.

G04 must be programmed alone in a block.

#### Example

G94 G04 X1000 ;

Standard notation:  $1000 * 0.001 = 1$  second

Pocket calculator notation: 1000 seconds

G95 G04 X1000 ;

Standard notation:  $1000 * 0.001 = 1$  spindle revolution

Pocket calculator notation: 1000 spindle revolutions

## 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.

### 3.4.1 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.

### 3.4.2 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.

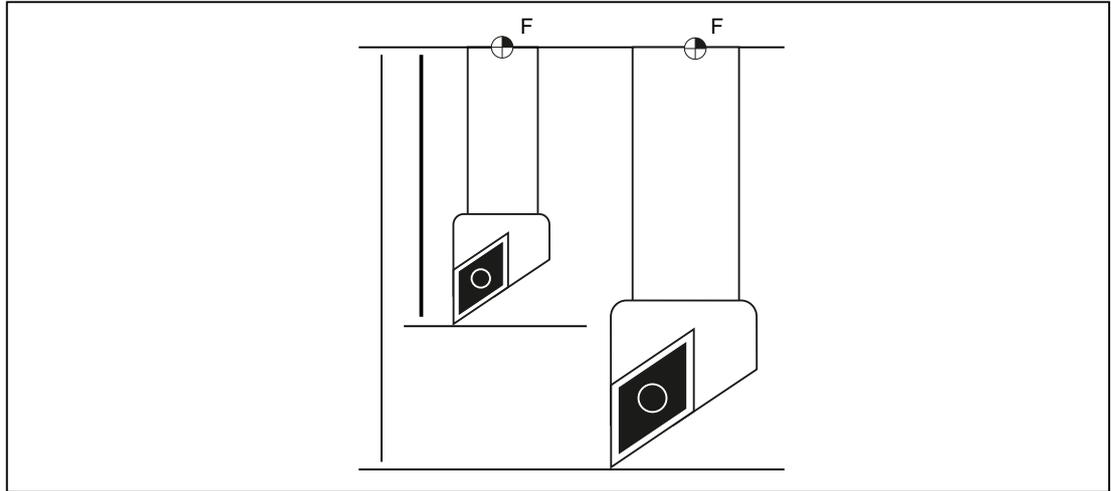


Figure 3-5 Tool length

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.

### 3.4.3 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 with G41 or G42.

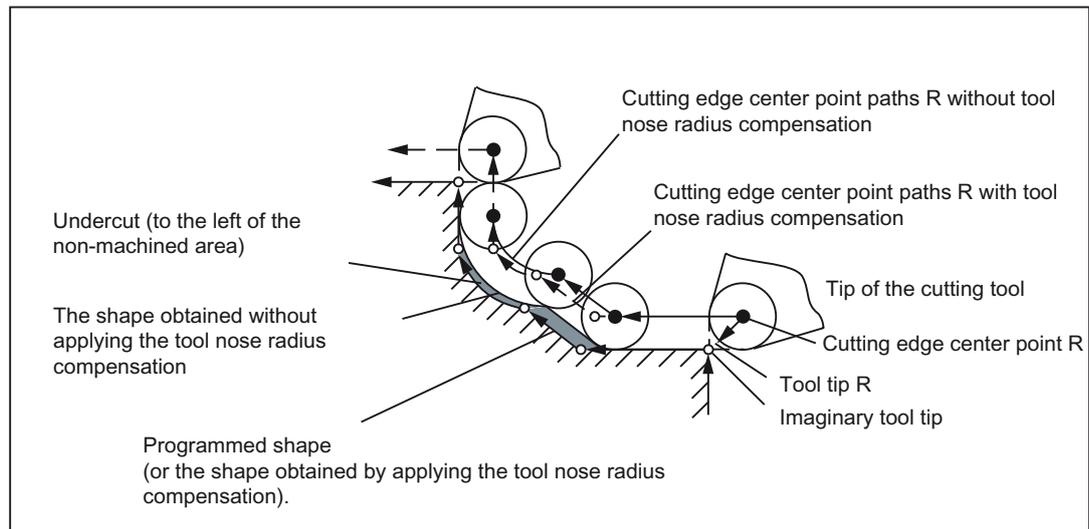


Figure 3-6 Machining without tool nose radius compensation

#### 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.

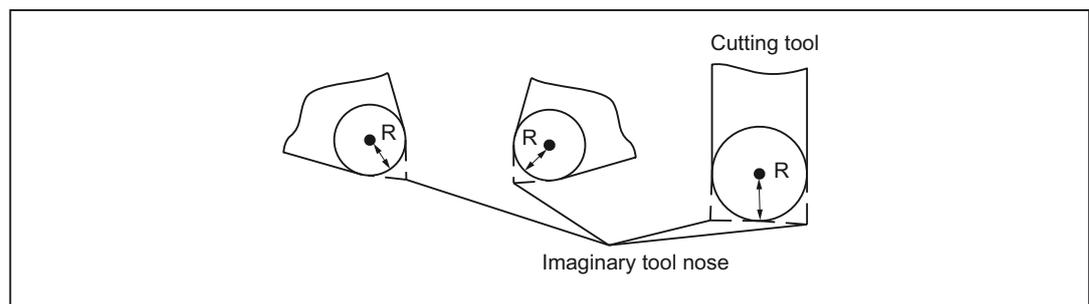


Figure 3-7 Defining the amount of the tool nose radius compensation and an imaginary tool tip

### 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.

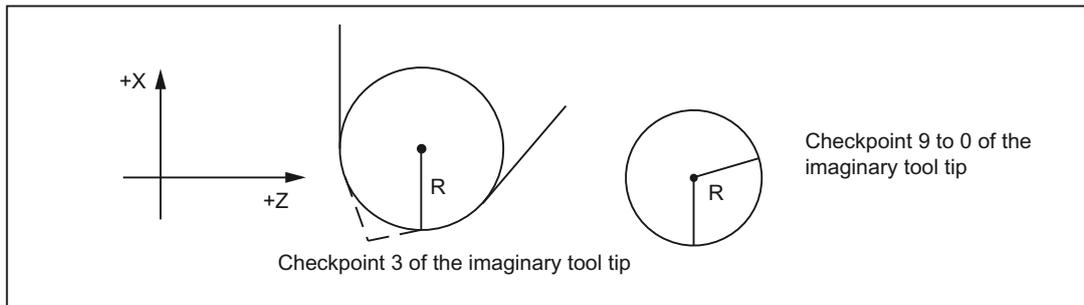


Figure 3-8 Example for defining a checkpoint

### 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.

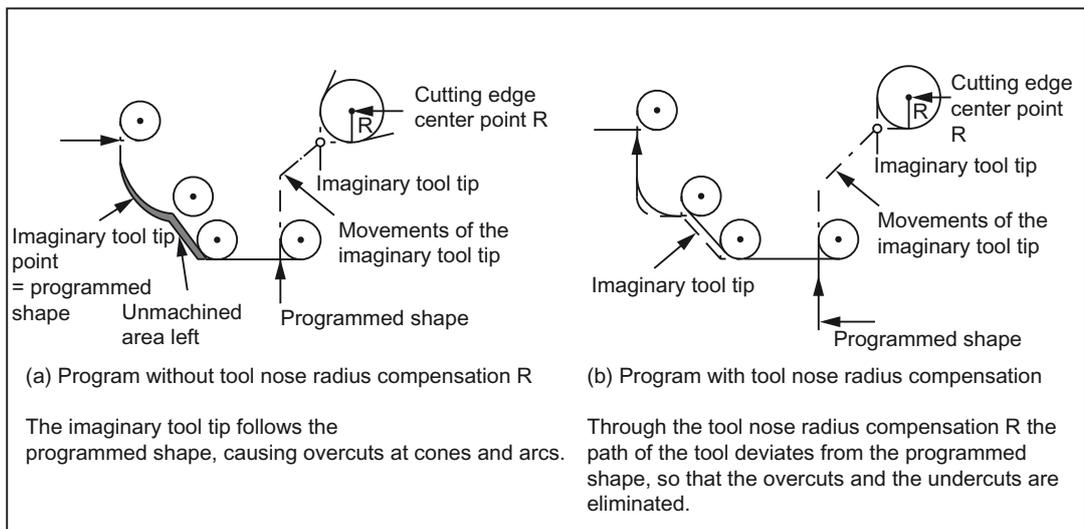


Figure 3-9 Program and tool movements for the checkpoints 1 to 8

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.

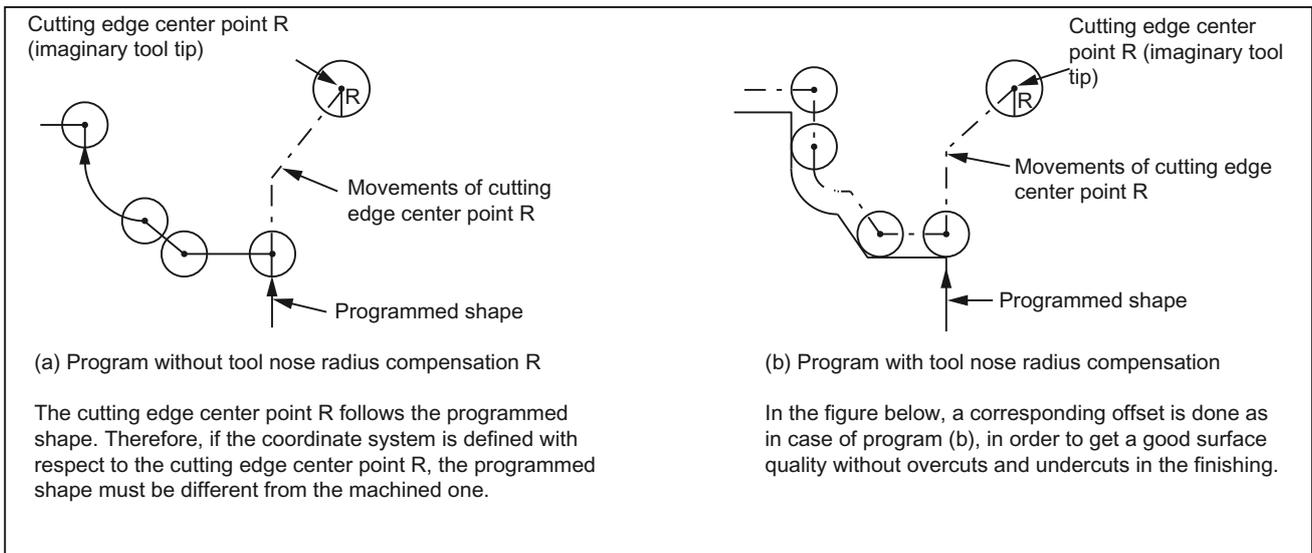


Figure 3-10 Program and tool movements for the checkpoints 0 to 9

### 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.

Table 3- 6 G functions 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.

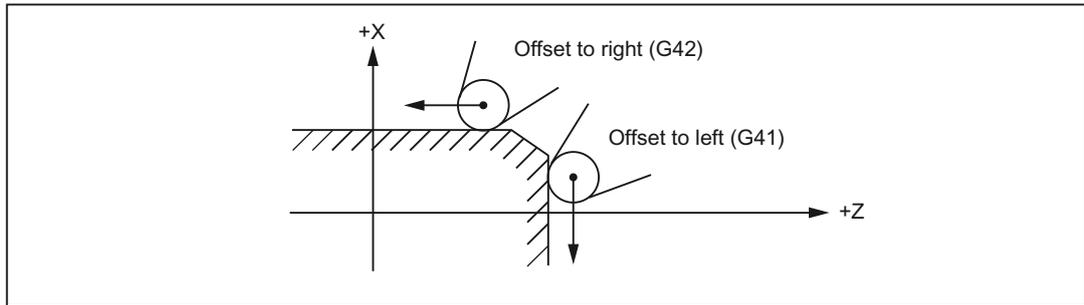


Figure 3-11 Defining the tool nose radius compensation depending upon the machining direction

### 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.

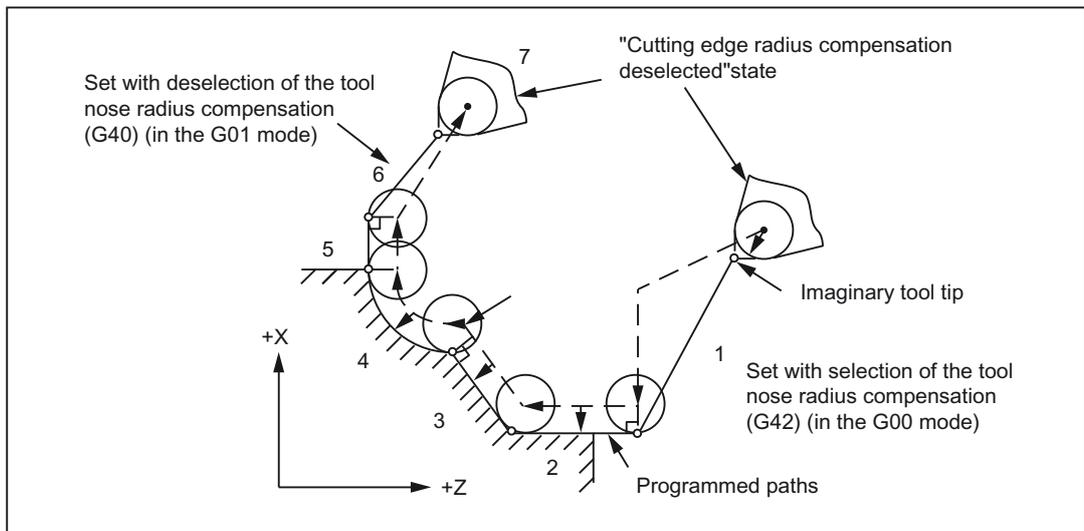


Figure 3-12 Contour of the motions of the tool radius compensation (G42, checkpoint 3)

- 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.

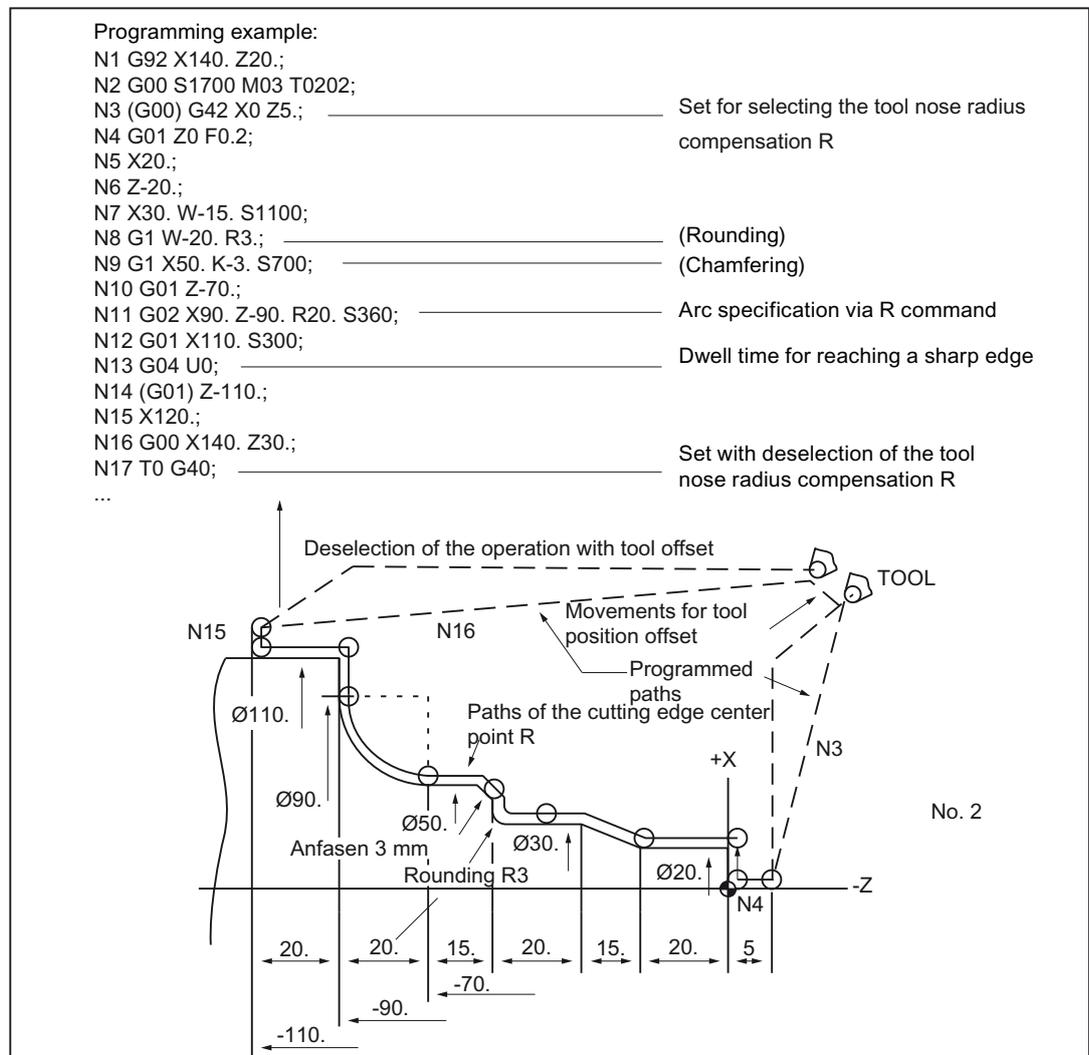


Figure 3-13 Programming example

## 3.5 S-, T-, M- and B functions

### 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.

### 3.5.2 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.

Table 3- 7 G commands for controlling a constant cutting rate

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.

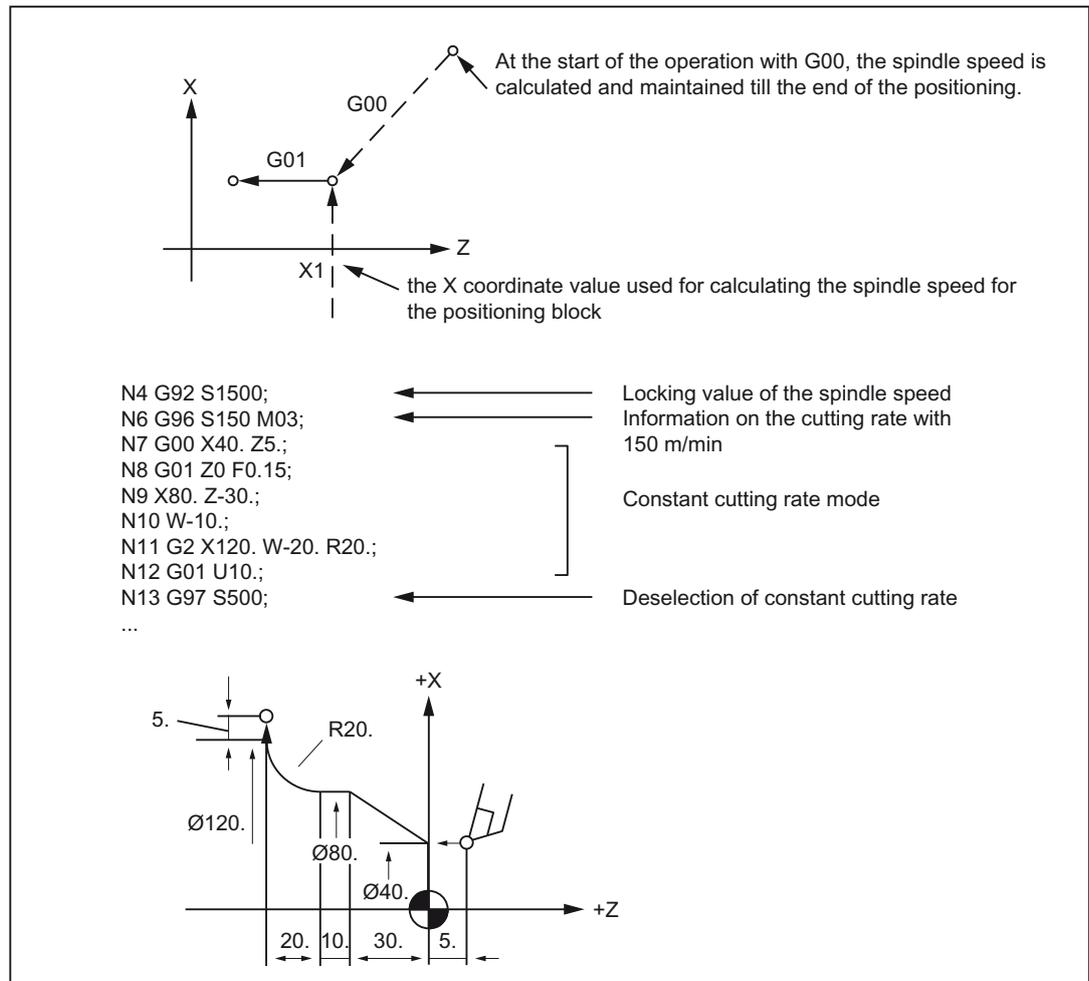


Figure 3-14 Constant cutting rate

### 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; N9 G96 S100 M03; : :	←	M function for selecting the gear stage (Example: Gear stage no. 4)
---	---	--

### 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.

### 3.5.4 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 (max. 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 NC start.

#### **M01 (optional stop)**

M01 can be set via

- 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.

---

### 3.5.5 M functions of spindle control

Table 3- 8 M functions of spindle control

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.

### 3.5.6 M functions for subroutine calls

Table 3- 9 M functions for subroutine calls

M function	Function
M98	Subprogram call
M99	Subprogram end

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

### 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 $C_V == 20 GOTOF N60         ;($C_V)
N0050       SETAL(61001)
N0060       M17
```

### 3.5.8 M functions

#### General M functions

The non-specific M functions are defined by the machine manufacturer. A representative example of the use of 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.

Table 3- 10 Other general M functions

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.

## Additional functions

### 4.1 Program-support functions

#### 4.1.1 Fixed cycles

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.

#### 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.

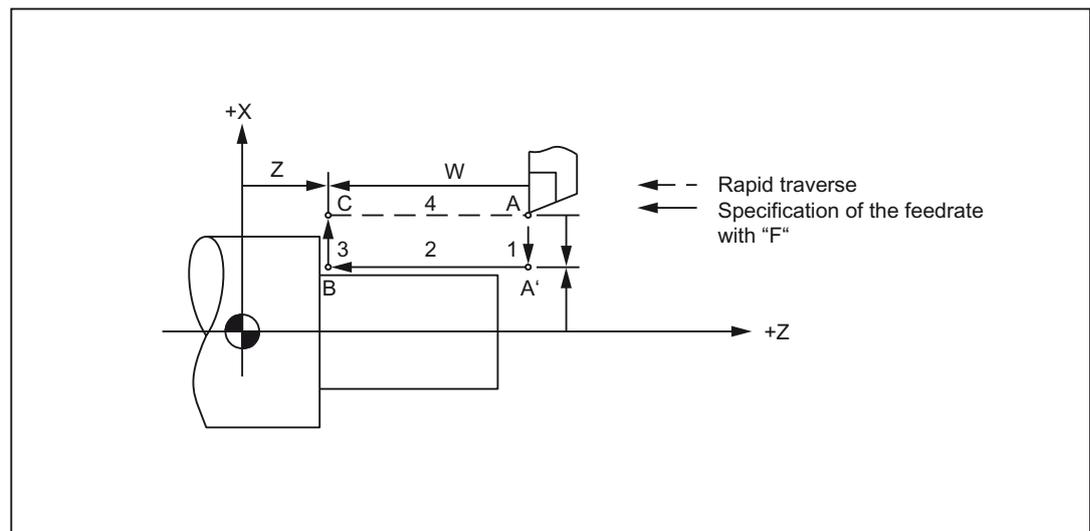


Figure 4-1 Longitudinal turning cycle

4.1 Program-support functions

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

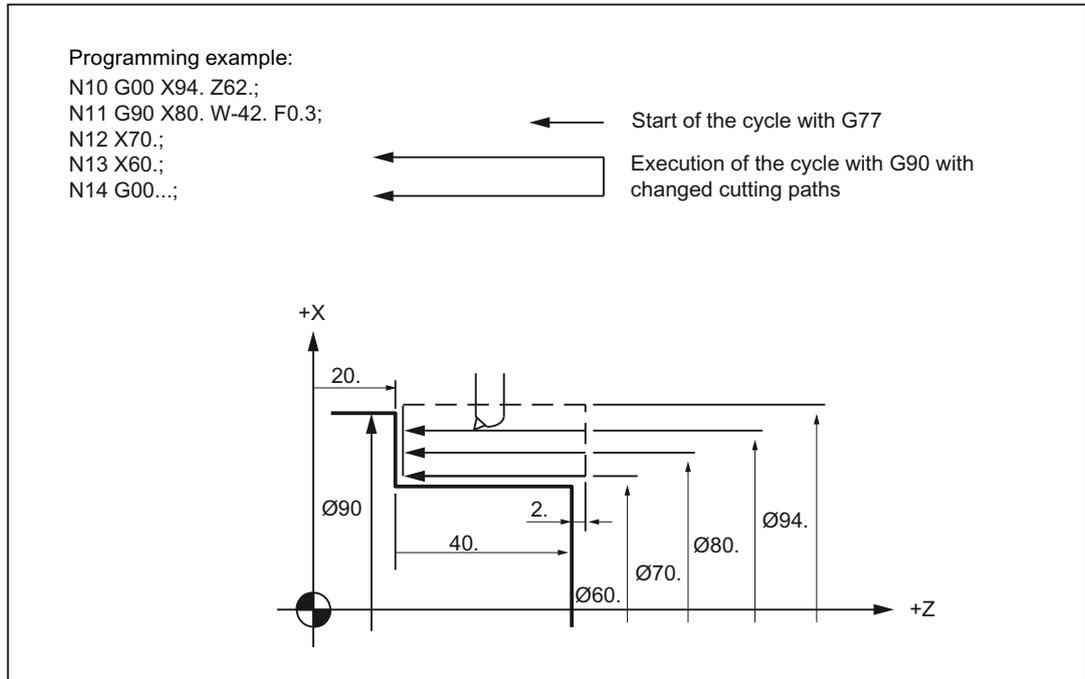


Figure 4-2 Longitudinal turning cycle (G code system A)

**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.

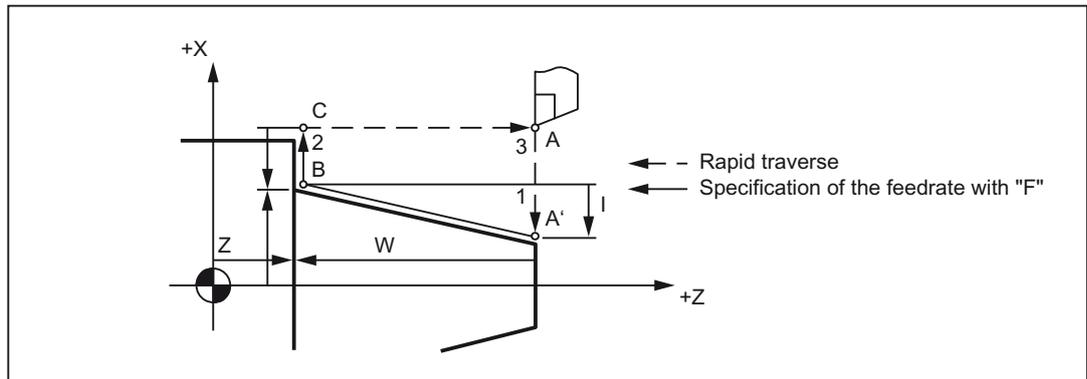


Figure 4-3 Straight cutting cycle

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

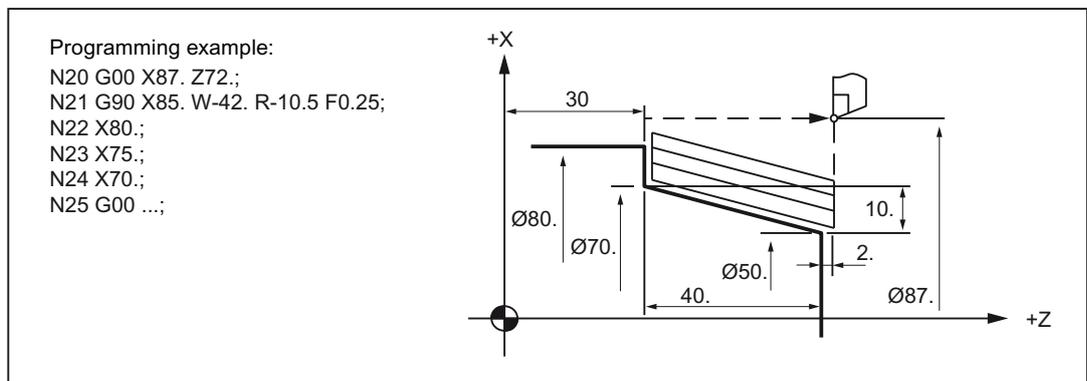


Figure 4-4 Straight cutting cycle (G code system A)

- When the cycle with G77 (G90, 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 G77 (G90, G20), are to be specified in the blocks before the block G77 (G90, 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 G77 (G90, G20).



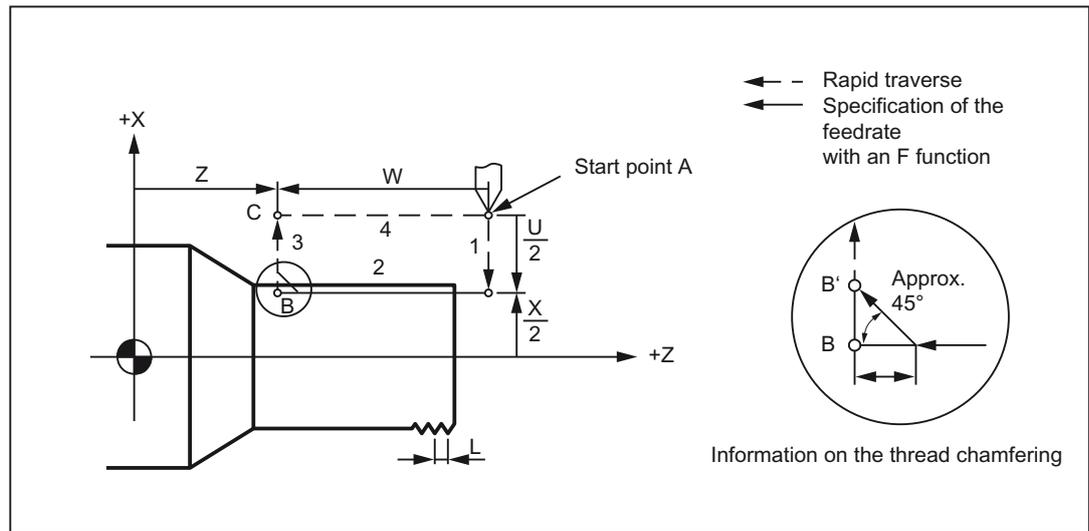


Figure 4-5 Cycle for cutting cylindrical threads

Since G78 (G92, 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 G78 (G92, G21) need not be specified again.

4.1 Program-support functions

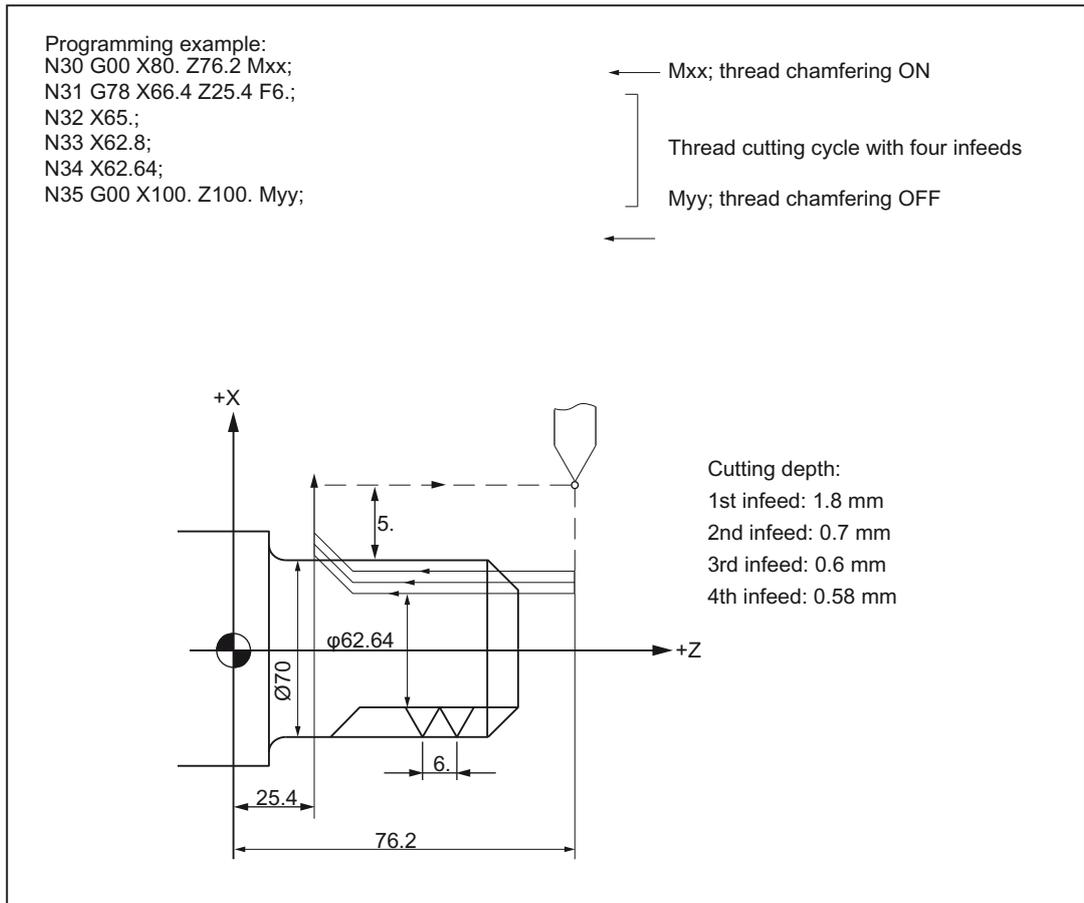


Figure 4-6 Cycle for cutting a cylindrical thread (G code system B)

- When the cycle with G78 (G92, 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 GUD7 \_ZSFI[26]. "L" is thus the specified thread lead.

### Cycle for cutting taper threads

#### Format

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

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... Q...;", a cycle for cutting taper threads is executed as per the sequence 1-4 given in the figure below.

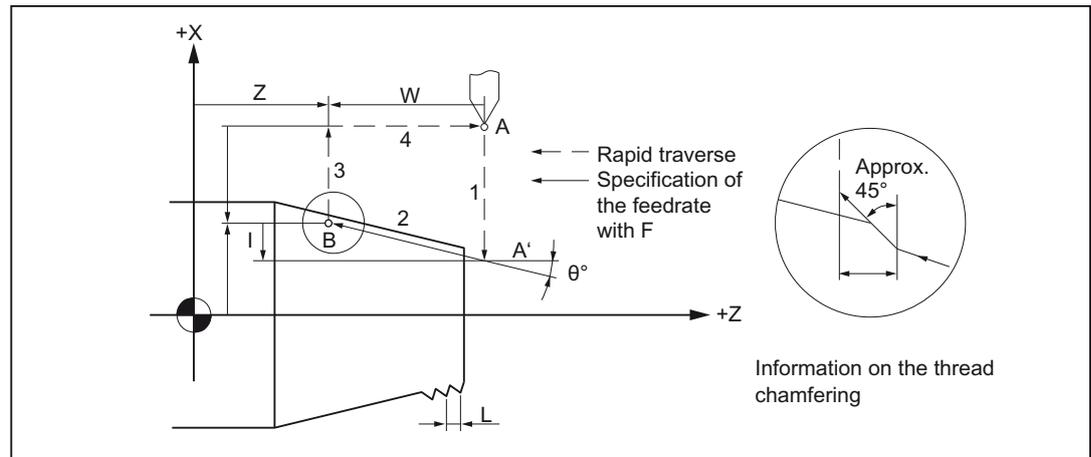


Figure 4-7 Cycle for cutting taper threads

The sign before the address character R depends on point A' of the viewing direction from point B. Since G78 (G92, 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 G78 (G92, G21) need not be specified again.

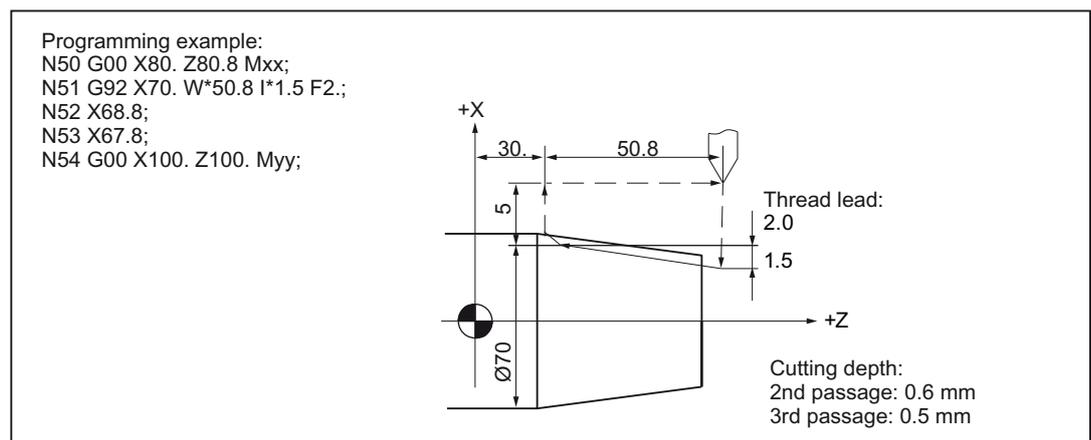


Figure 4-8 Cycle for cutting a taper thread (G code system A)

When the cycle with G78 (G92, 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 G78 (G92, G21), are to be specified in the blocks before the block G78 (G92, 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 G78 (G92, G21).

If at the moment, when the cutting tool is at the starting point A or at the chamfering completion point B, the button ZYKLUSSTART is activated, 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 button VORSCHUB HALT is activated 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.

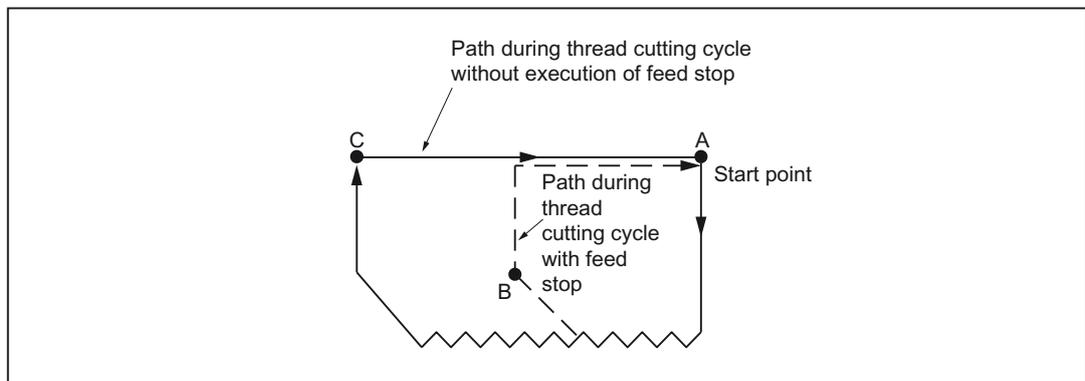


Figure 4-9 Feedrate halt during the execution of the thread-cutting cycle

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

## Straight facing 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.

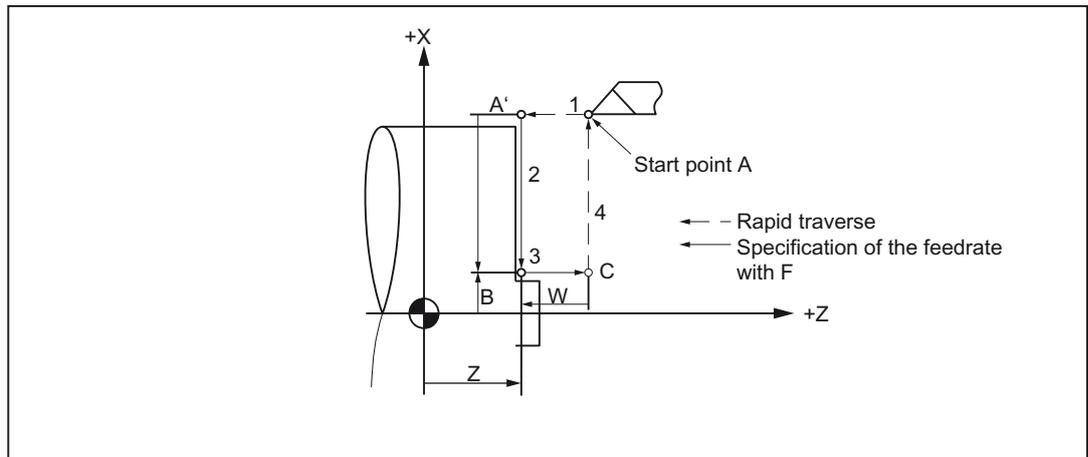


Figure 4-10 Straight facing cycle

Since G79 (G94, 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 G79 (G94, G24) need not be specified again.

4.1 Program-support functions

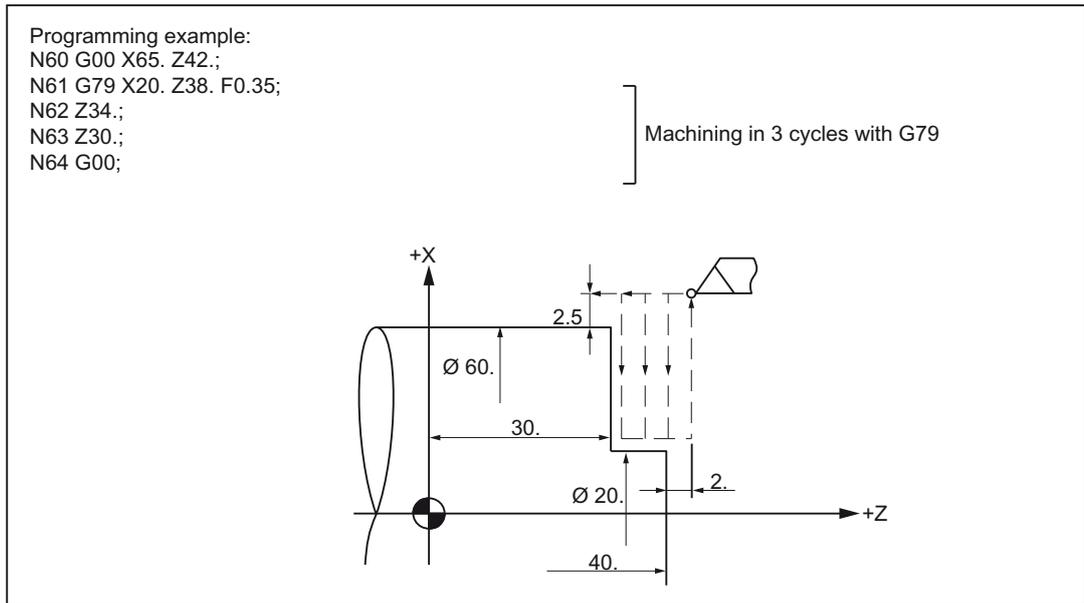


Figure 4-11 Straight facing cycle (G code system B)

Transverse-taper turning cycle

Format

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

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... Q...;", a transverse-taper turning cycle is executed as per the sequence 1-4 given in the figure below.

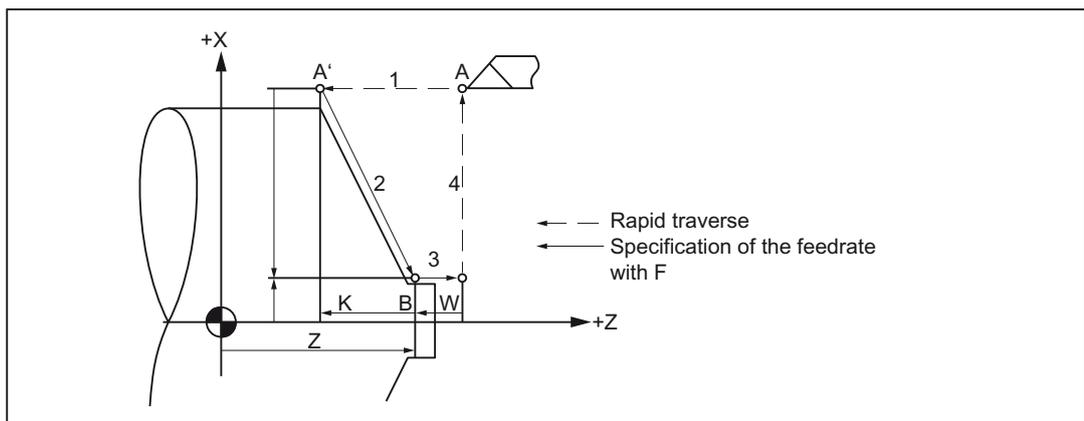


Figure 4-12 Transverse-taper turning cycle

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

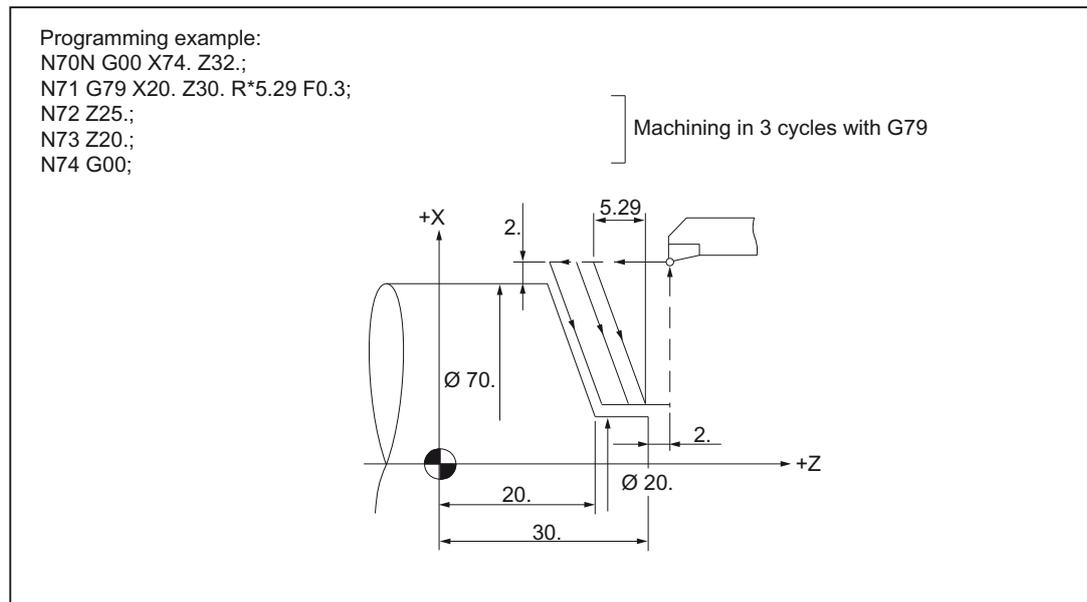


Figure 4-13 Transverse-taper turning cycle (G code system B)

The functions S, T and M, which are used as cutting conditions for the execution of G79 (G94, G24), are to be specified in the blocks before the block G79 (G94, 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 G79 (G94, G24).

When the cycle with G79 (G94, 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.

## 4.1.2 Multiple repetitive cycles

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.

Table 4- 1 Overview of the turning cycles G70 to G76 (G code system A and B)

G code	Description
G70	Finishing cycle
G71	Stock removal cycle, longitudinal axis
G72	Stock removal cycle, transverse axis
G73	Contour repetition
G74	Deep hole drilling and grooving in the longitudinal axis
G75	Deep hole drilling and grooving 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.

Table 4- 2 Overview of the turning cycles G72 to G78 (G code system C)

G code	Description
G72	Finishing cycle
G73	Stock removal cycle, longitudinal axis
G74	Stock removal cycle, transverse axis
G75	Contour repetition
G76	Deep hole drilling and grooving in the longitudinal axis
G77	Deep hole drilling and grooving 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.

**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.



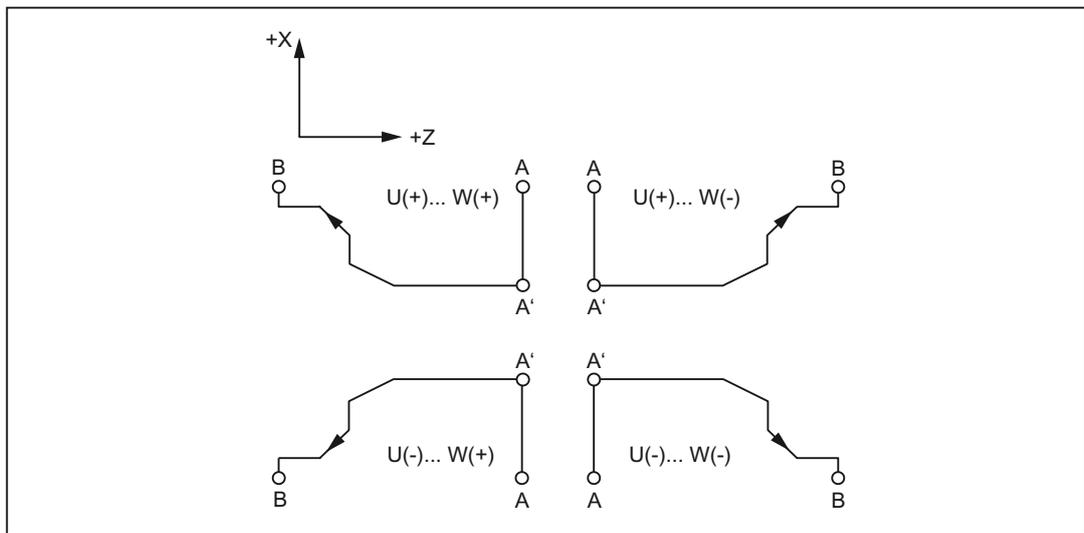
4.1 Program-support functions

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**

- $\Delta d$  as well as  $\Delta u$  are specified with the address character U. When the address characters P and Q are specified,  $\Delta "u"$  is the case.
- There are a total of four different cutting sectors. As shown in the figure below,  $\Delta "u"$  and  $\Delta "w"$  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.

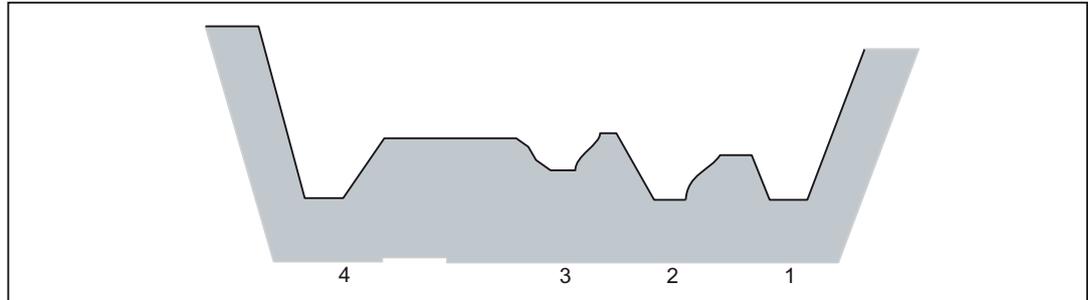


Figure 4-15 Pockets in case of a stock removal cycle (Type II)

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

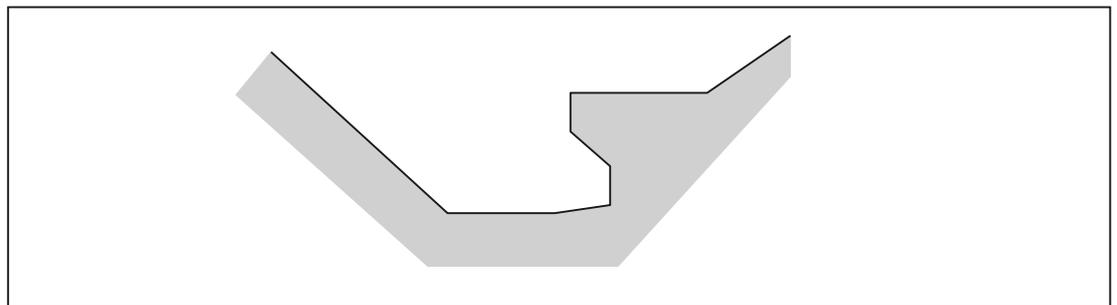


Figure 4-16 A contour, which cannot be machined in a cycle G71

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

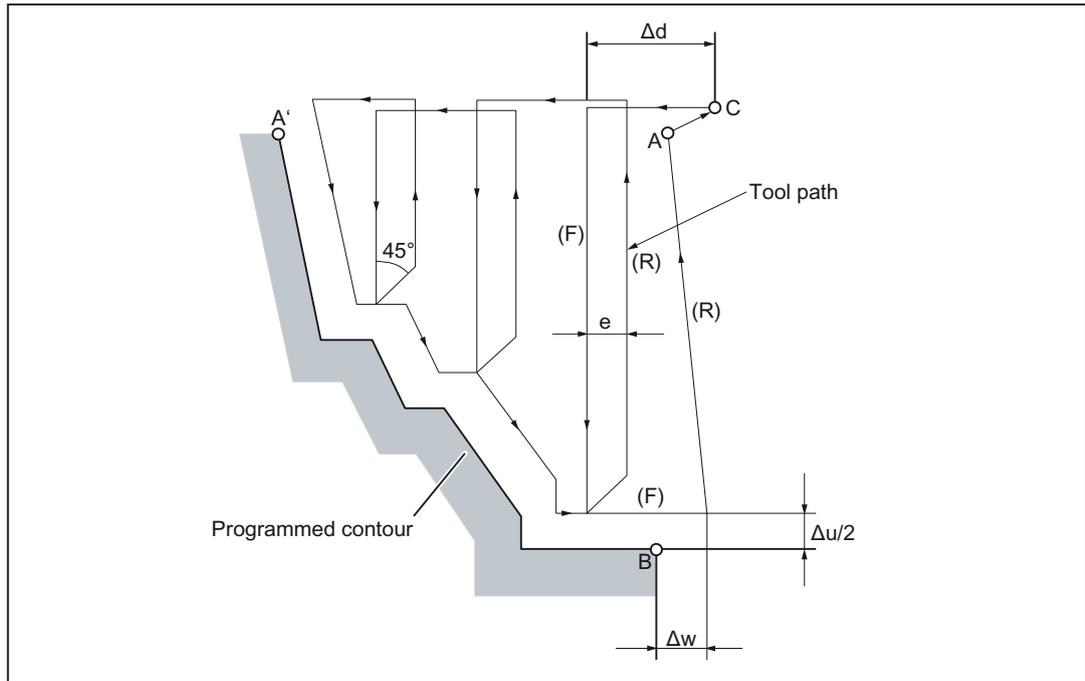


Figure 4-17 Cutting path of a stock removal cycle, transverse axis

### Format

**G72 W... R... ;**

The significance of the addresses W ( $\Delta d$ ) 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 ( $\Delta u$ ), W ( $\Delta w$ ), F, S and T have the same significance as in the cycle G71.

### Note

#### Stock removal cycle transverse axis

- The values  $\Delta "i"$  and  $\Delta "k"$  or  $\Delta "u"$  and  $\Delta "w"$  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  $\Delta "i"$  or  $\Delta "k"$ , when P and Q are not specified in the same block. The address characters U and W refer to  $\Delta "u"$  and  $\Delta "w"$ , 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,  $\Delta "u"$  and  $\Delta "w"$  may have different signs:

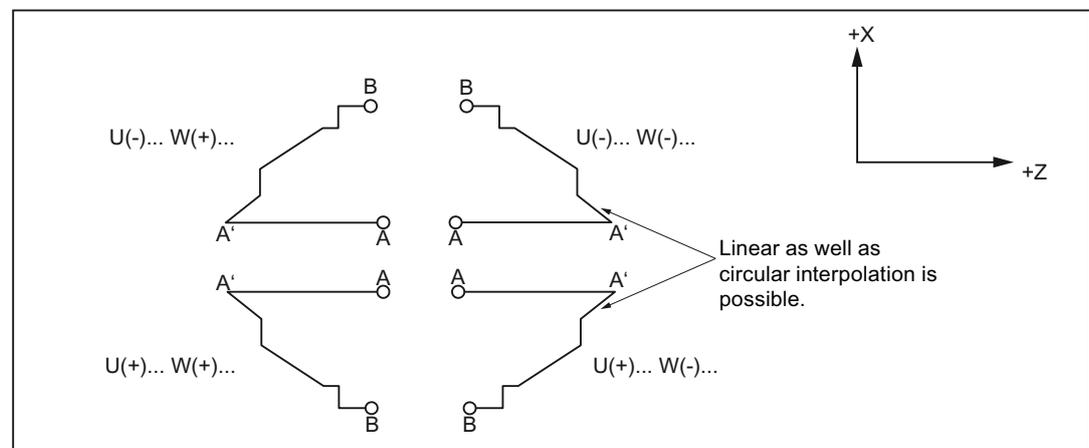


Figure 4-18 Signs of the numbers with U and W in the stock removal during face turning

### 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  $\Delta u$ ,  $\Delta w$ ,  $\Delta k$  and  $\Delta i$ . As soon as the execution cycle has ended, the tool returns to the point A.

**Contour repetition (G73)**

The contour repetition 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.

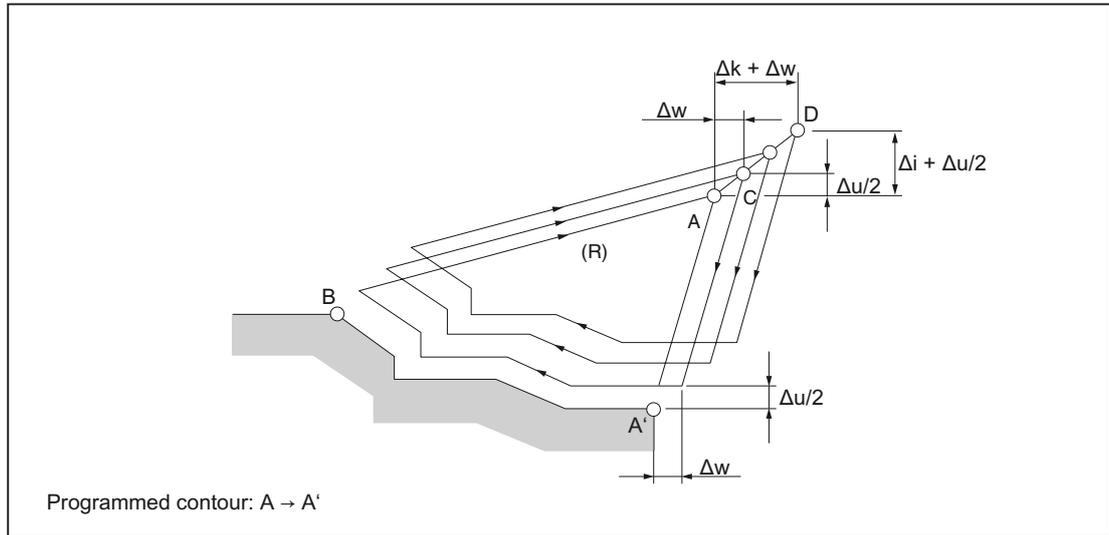


Figure 4-19 Cutting path in the contour repetition

**Format**

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

**U:** Distance ( $\Delta i$ ) 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 GUD7,  $\_ZSFI[32]$ , but this value is overwritten by the value of the program command.

**W:** Distance ( $\Delta k$ ) 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 GUD7,  $\_ZSFI[33]$ , 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 GUD7,  $\_ZSFI[34]$ , 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 ( $\Delta u$ ) (diameter -/radius programming)

**W:** Finishing allowance in the direction of Z axis ( $\Delta w$ )

**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

1. 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.
  2. The tool returns to the starting point and the next block is read, as soon as the execution cycle is completed with G70.
  3. Within the blocks defined with the address characters P and Q it is possible to call subroutines.
-

Examples

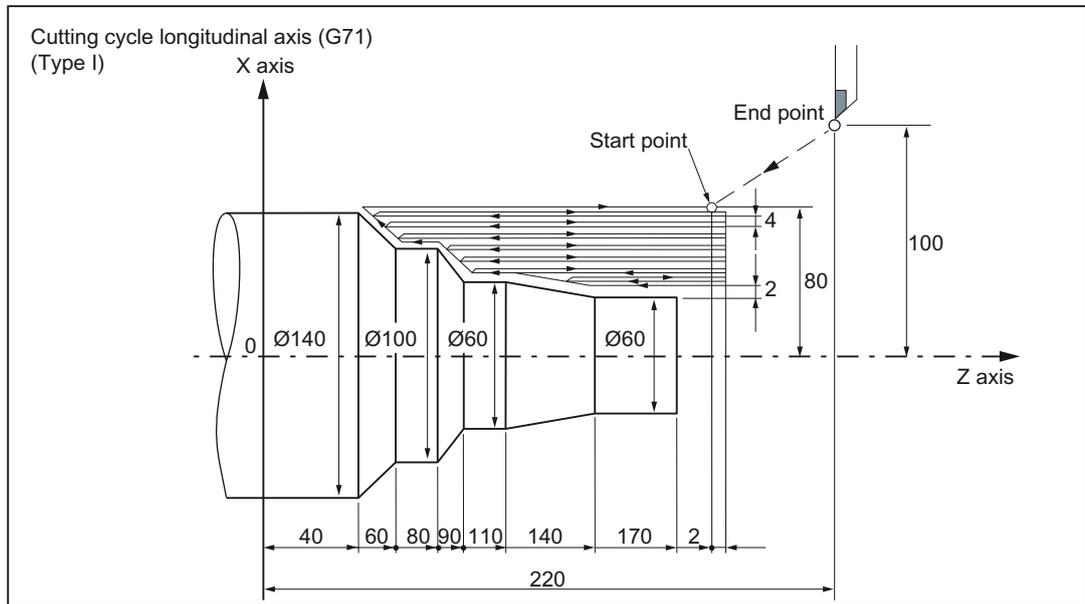


Figure 4-20 Stock removal cycle, longitudinal axis

(Diameter programming, input metric)

```

N010 G00 G90 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
    
```

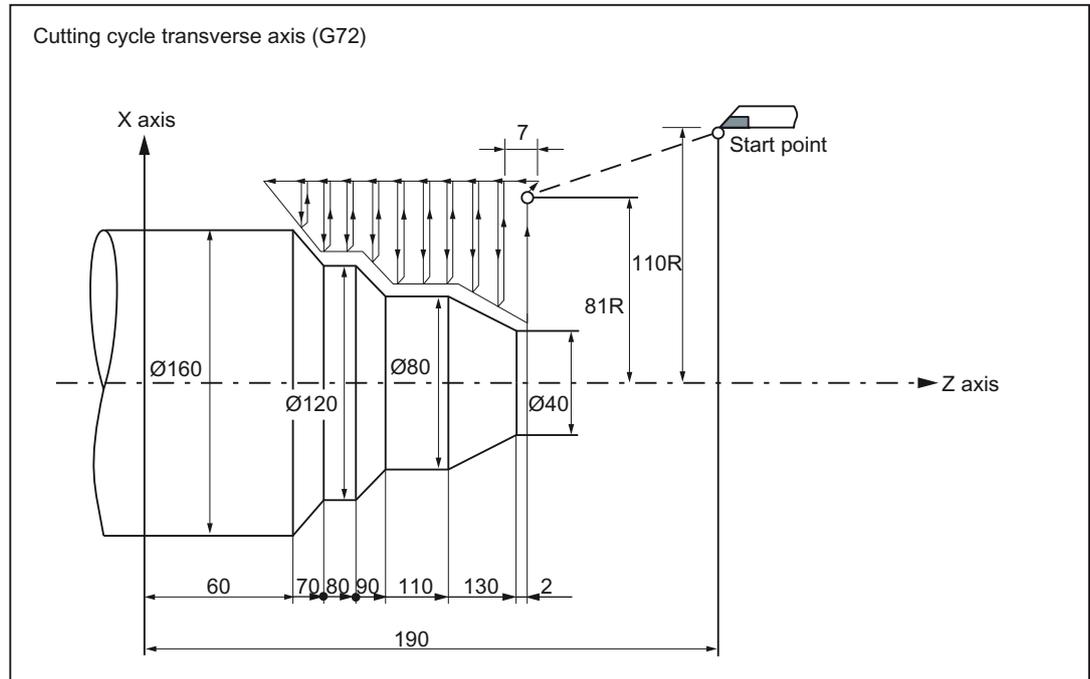


Figure 4-21 Stock removal cycle transverse axis

(Diameter programming, input metric)

```

N010 G00 G90 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
    
```

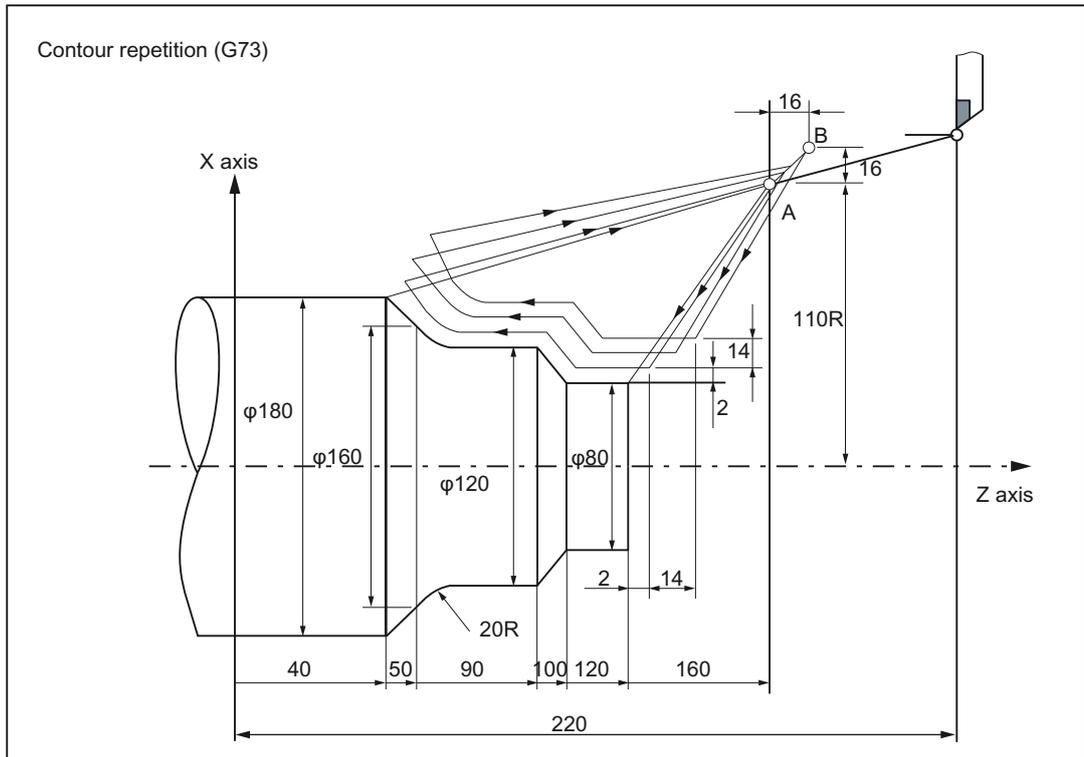


Figure 4-22 Contour repetition

(Diameter programming, input metric)

```

N010 G00 G90 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
    
```

### Deep hole drilling and grooving in the longitudinal axis (G74)

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

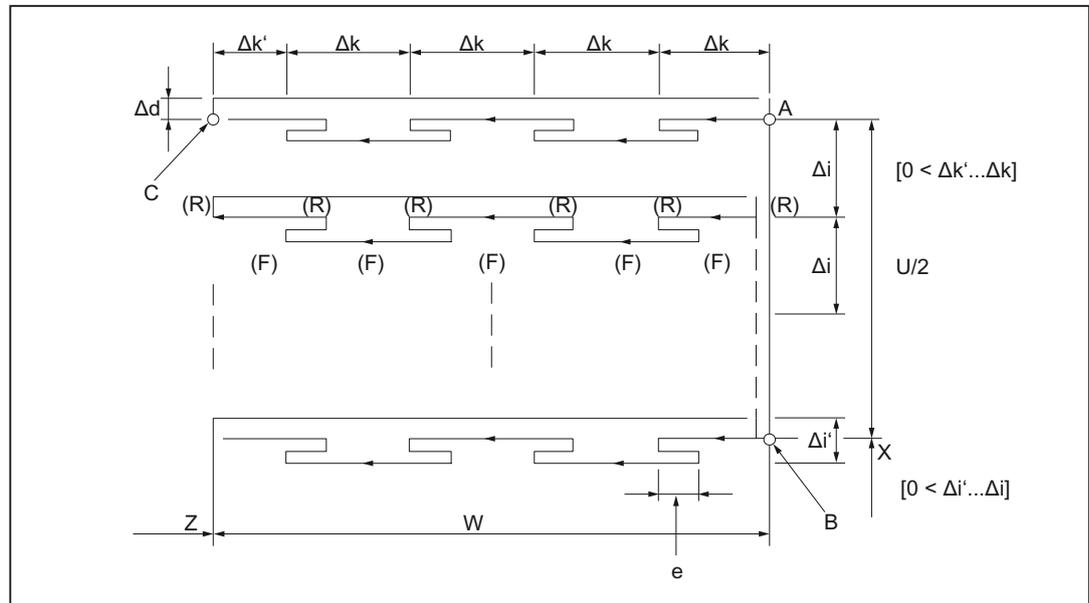


Figure 4-23 Cutting path in case of a deep hole drilling cycle

### 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 GUD7, \_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 ( $\Delta i$ ) in X direction (without sign)

**Q:** Infeed amount ( $\Delta k$ ) in Z direction (without sign)

**R:** Retraction amount ( $\Delta d$ ) at the groove base

**F:** Feedrate

**Note**

**Deep hole drilling and grooving in the longitudinal axis**

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

**Deep hole drilling and grooving in the transverse axis (G75)**

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

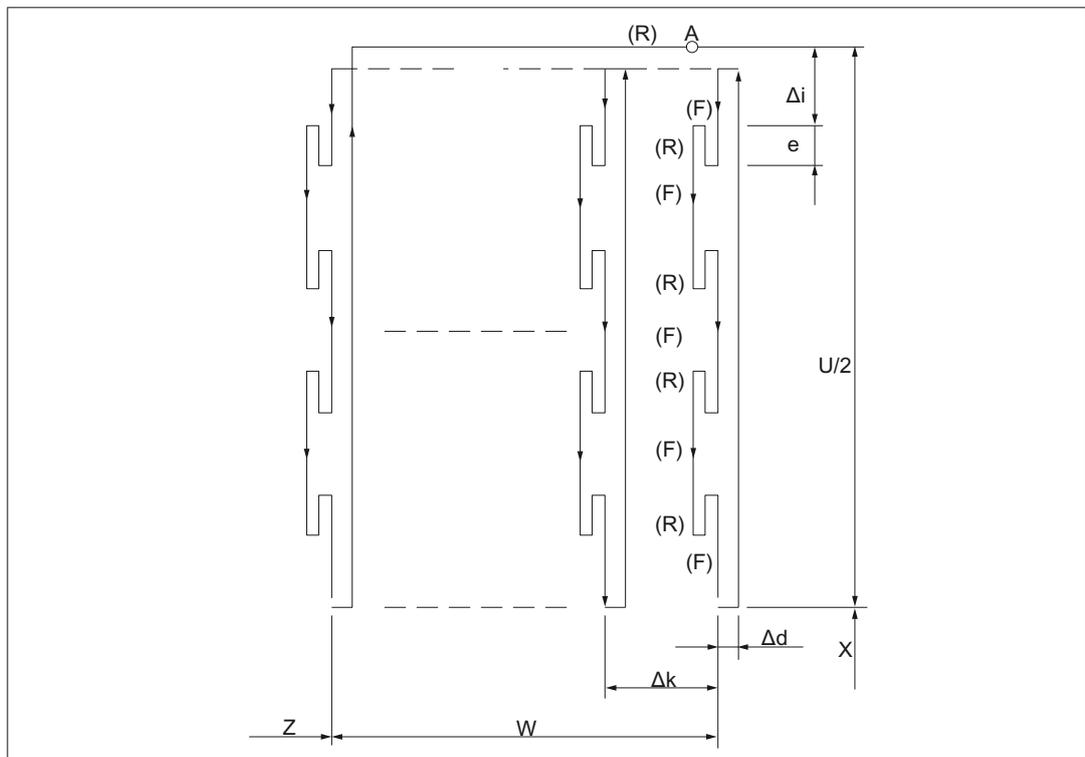


Figure 4-24 Cutting path in deep hole drilling and grooving in the transverse axis (G75)

**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.

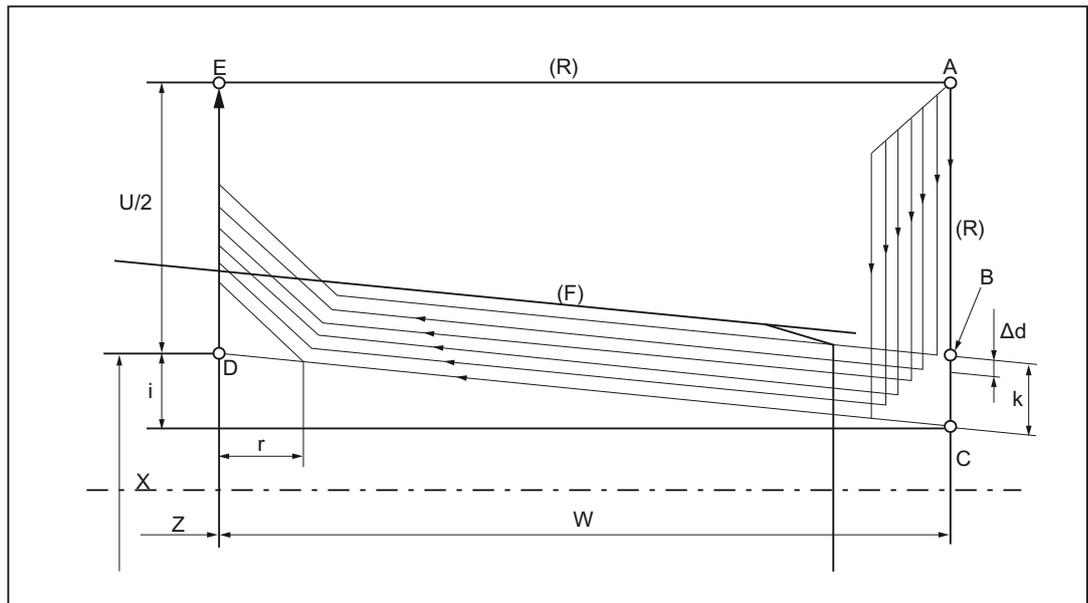


Figure 4-25 Cutting path in case of a cycle for cutting multiple-start threads

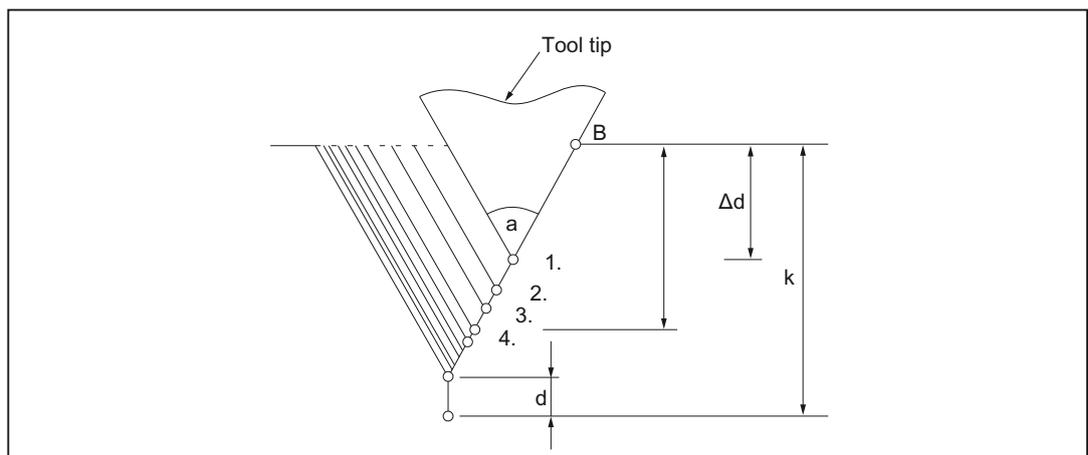


Figure 4-26 Infeed during thread cutting

**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 GUD7, `_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 ( $1/10 \times$  thread lead)

This value is modal and remains effective till another value is programmed. The value can also be entered via GUD7, `_ZSFI[26]`, but this value is overwritten by the value of the program command.

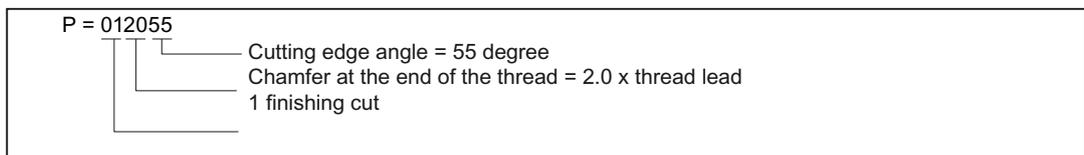
**a:** Facet angle

This value is modal and remains effective till another value is programmed. The value can also be entered via GUD7, `_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 ( $\Delta d_{min}$ ), radius value

Always when the cutting depth during a cycle operation ( $\Delta d - \Delta d_{-1}$ ) 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 GUD7, `_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 GUD7, `_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 ( $\Delta d$ ), radius value

**F:** Thread lead (L)

---

**Note**

**Multiple-thread cutting cycle**

1. The significance of the data specified with the address characters P, Q and R is determined by the appearance of X (U) and X (W).
  2. 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.  $\Delta d$  on the first path and  $\Delta d_n$  on the nth path. Corresponding to the respective sign of the address character, four symmetric sections are considered here.
  3. The same instructions as for thread cutting with G32 or for the thread cutting cycle with G92 are applicable here.
- 

**Examples**

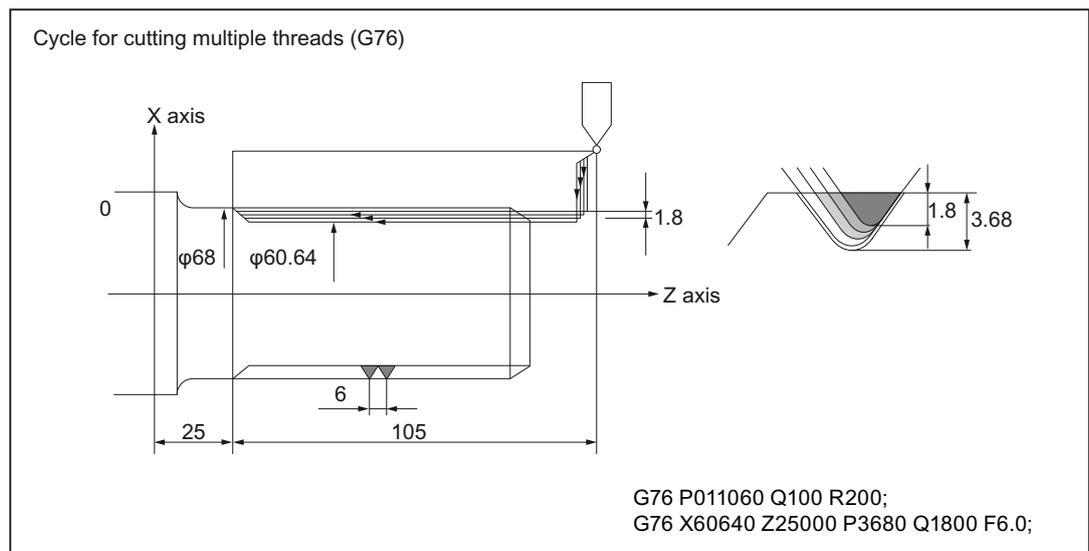


Figure 4-27 Cycle for cutting threads (G76)

---

**Note**

**Supplementary conditions**

1. In the 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 the MDA mode.
  2. 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.
  3. 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
  4. 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.
  5. 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.
  6. In the cycles G71, G72, G73, G74, G75, G76 and G78 no tool nose radius compensation may be performed.
- 

### 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.

Table 4- 3 Drilling cycles

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 feedrate	Dwell time -> left spindle run	Cutting feedrate	Tapping front face
G85	Cutting feedrate	Dwell time	Cutting feedrate	Drilling front face
G87	Interrupted cutting feedrate	Dwell time	Rapid traverse	Deep hole drilling side surface
G88	Cutting feedrate	Dwell time -> left spindle run	Cutting feedrate	Tapping side surface
G89	Cutting feedrate	Dwell time	Cutting feedrate	Drilling side surface

## Explanations

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

- 1st working cycle  
Positioning the X, (Z) and C axis
- 2nd working cycle  
Rapid traverse movement to plane R
- 3rd working cycle  
Drilling
- 4th working cycle  
Machining at drilling base
- 5th working cycle  
Retraction to plane R
- 6th working cycle  
Rapid retraction to positioning plane

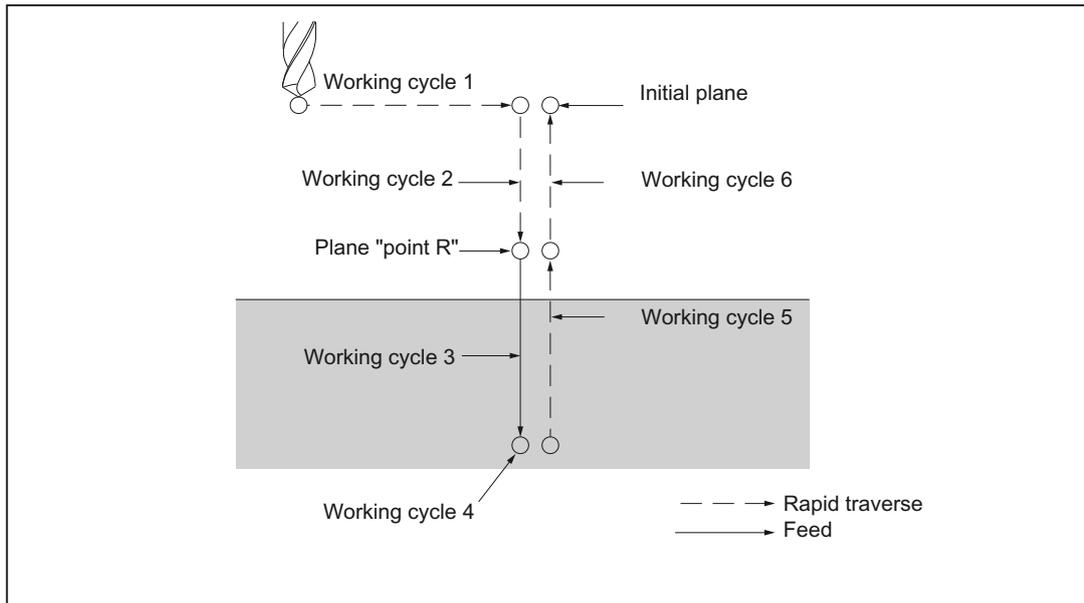


Figure 4-28 Sequence of operations in the drilling cycle

**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.

Table 4- 4 Positioning plane with the corresponding drilling 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 until they are deselected again. As long as these G functions are selected, the drilling mode remains active. The data is retained until 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.

### Reset point level (G98/G99)

In case of an active G code system A, the tool moves away from the bottom of a hole and back to the initial plane. If G98 is programmed in an active G code system B or C, then the tool returns from the bottom of a hole back to the initial plane. If G99 is programmed, the tool returns from the bottom of a hole to the plane R.

G99 is generally used for the first drilling operation, while G98 is always used for the last drilling operation. The initial plane does not change even when the drilling is done in the operation with G99.

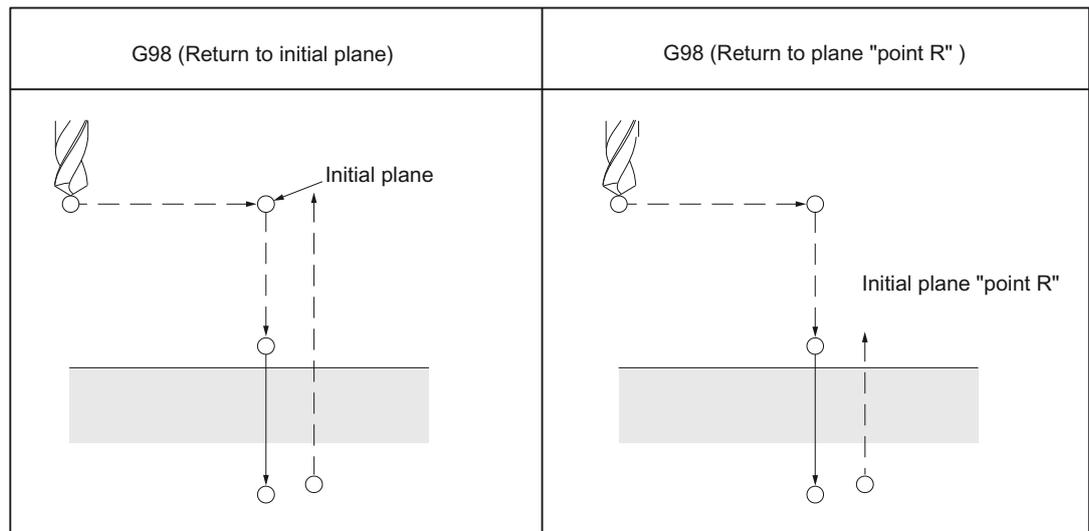


Figure 4-29 Plane for the return point (G98/G99)

### Repetition

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. If the first drilled hole is specified as absolute (G90), then the drilling is done at the same point again; therefore "K" is to be specified incrementally (G91).

The drilling data is 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
M $\alpha$	M function for locking the C axis
M( $\alpha$ +1)	M function for releasing the C axis

Figure 4-30 Symbols and figures

 <b>CAUTION</b>
<b>Address character R</b>
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 on 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 GUD7 \_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) (GUD7 \_ZSFI[20]=0)

In case of a high-speed deep hole drilling cycle, the drill repeats the infeed with the cutting feedrate. This is retracted by a certain amount until the tool has reached the bottom of the 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 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 the hole
- F\_:** Cutting feedrate
- K\_:** Number of repetitions (if required)
- M\_:** M function for locking the C axis (if needed)

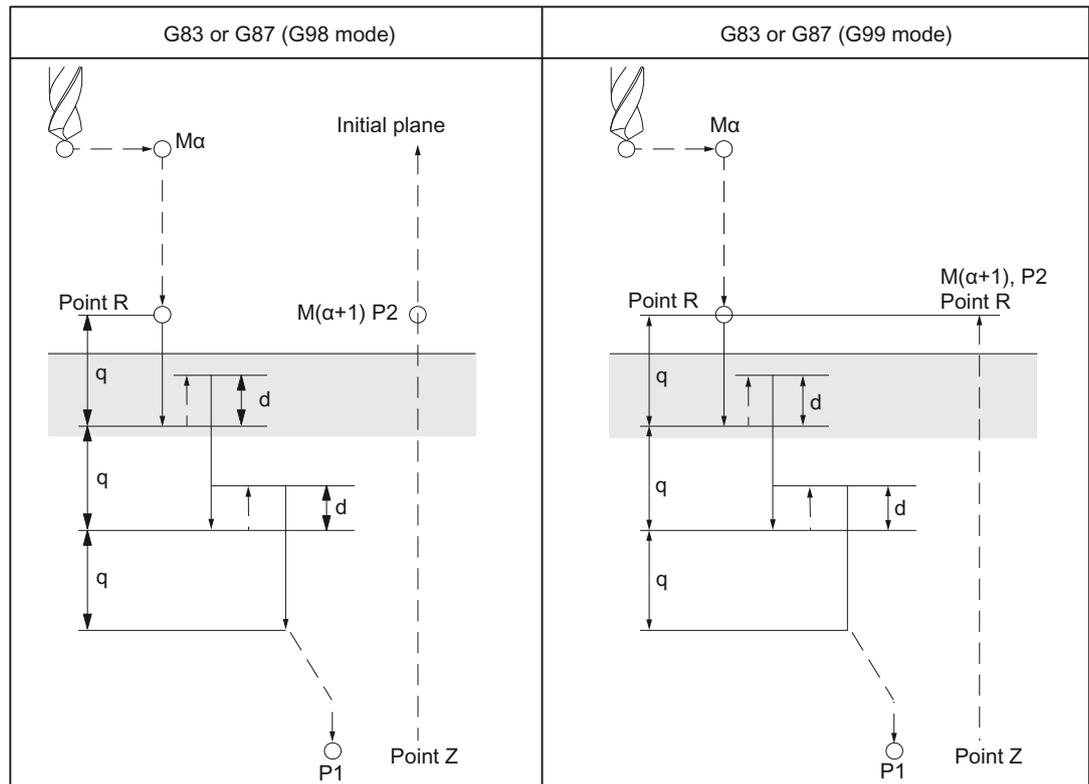


Figure 4-31 Cycle "high-speed deep hole drilling"

- M $\alpha$ :** M function for locking the C axis
- M( $\alpha$ +1):** M function for releasing the C axis
- P1:** Dwell time (Program)
- P2:** Specification of the dwell time in GUD7, `_ZSFR[22]`
- d:** Specification of the retraction amount in GUD7, `_ZSFR[21]`

### Deep hole drilling cycle (G83, G87) (GUD7 `_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 until the tool has reached the bottom of the hole.



**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 the 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 point R to the bottom of the hole

**R<sub>z</sub>:** Distance from the initial plane to the plane R

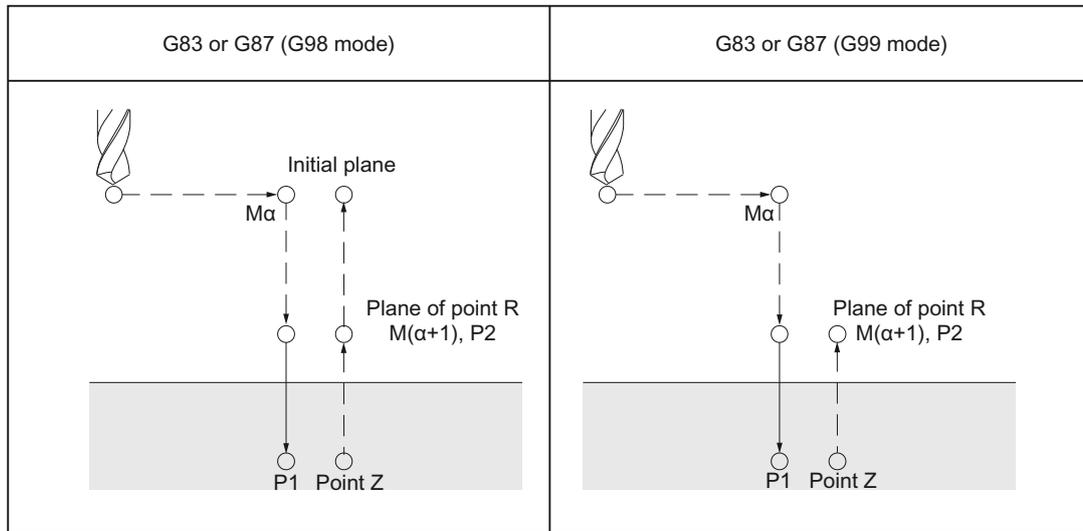
**P<sub>z</sub>:** Dwell time at the bottom of the hole

**F<sub>z</sub>:** Cutting feedrate

**K<sub>z</sub>:** Number of repetitions (if required)

**M<sub>z</sub>:** M function for locking the C axis (if needed)

4.1 Program-support functions



**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 GUD7, \_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 is reached for each cutting feedrate F, the return to the reference plane R takes place in rapid traverse. The approach motion for a new cut is also performed again with rapid traverse, and that by the distance (d) which can be set in GUD7\_ZSFR[10]. The distance d to the programmed cutting depth is traversed with the cutting feedrate F. The infeed Q is incremental 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.
  - The drilling depth/50 formula is always used for larger drilling depths (maximum value 7 mm).

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

In this cycle the spindle direction of rotation at the bottom of the 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 point R to the bottom of the hole

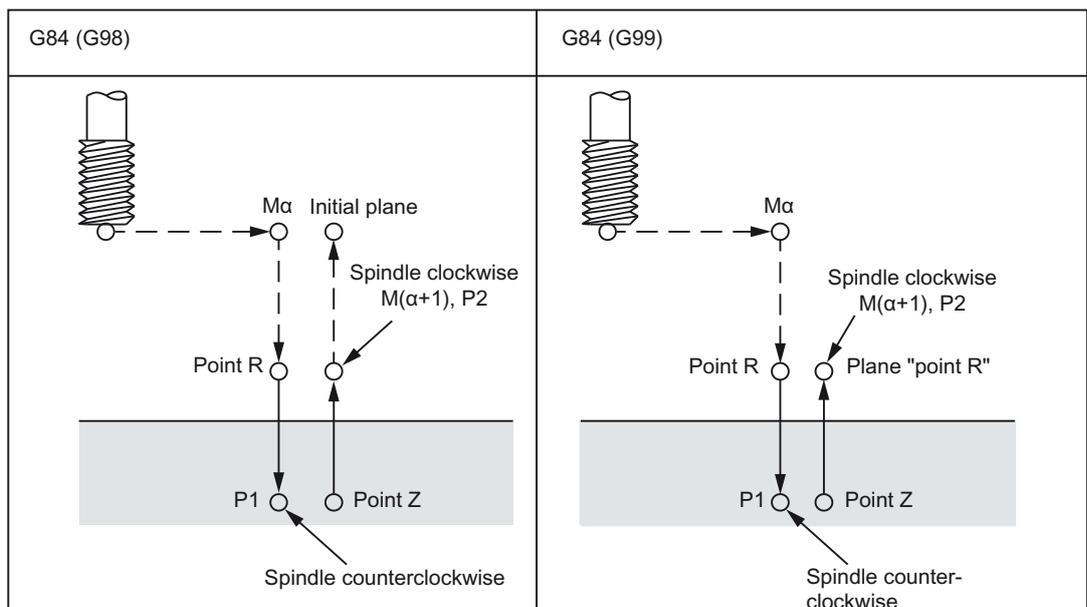
**R<sub>...</sub>:** Distance from the initial plane to the plane R

**P<sub>...</sub>:** Dwell time at the bottom of the hole

**F<sub>...</sub>:** Cutting feedrate

**K<sub>...</sub>:** Number of repetitions (if required)

**M<sub>...</sub>:** M function for locking the C axis (if needed)



P2: Specification of the dwell time in GUD7, \_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 until 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 point R to the bottom of the hole

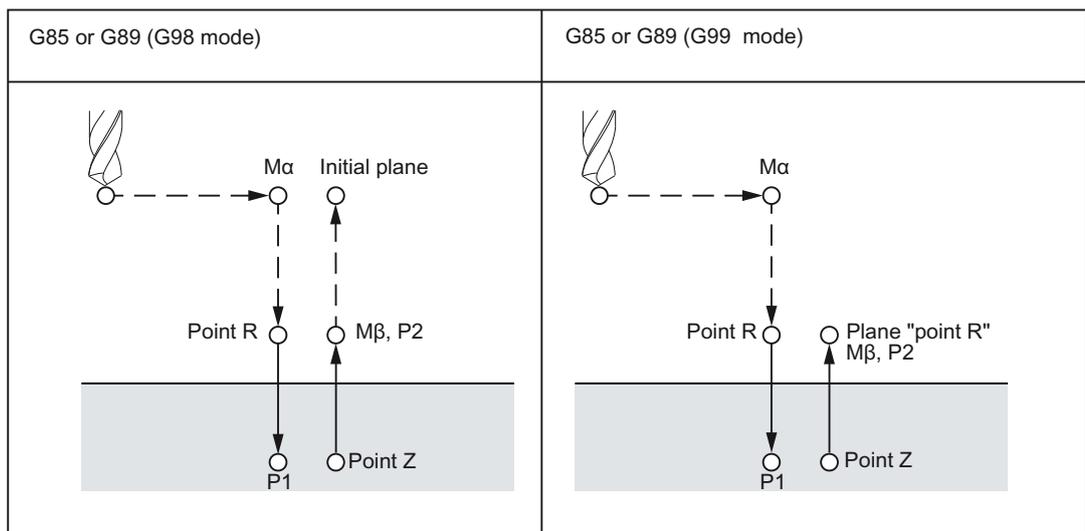
**R:** Distance from the initial plane to the plane R

**P:** Dwell time at the bottom of the 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 GUD7, \_ZSFR[22]

## Explanations

After the positioning at the bottom of the 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.

## 4.2 Programmable data input

### 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.

Table 4- 5 Description of the addresses

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)
U	Tool offset for the X axis (incremental)
V	Tool offset for the X axis (incremental)
W	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

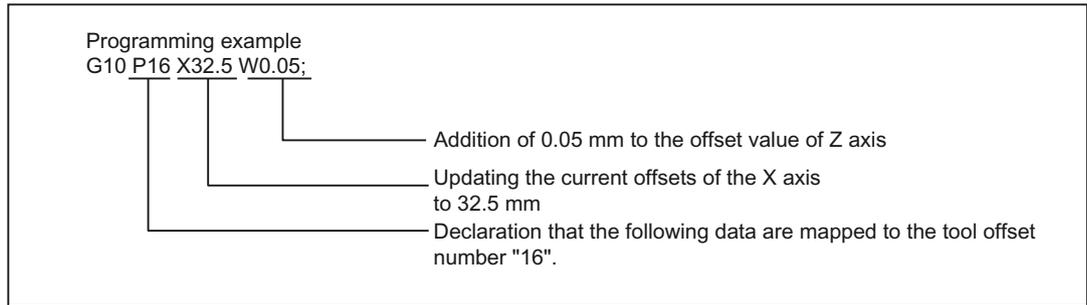
P1 to P99: Writing of tool wear

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

\$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

### 4.2.2 M function for calling subroutines (M98, M99)

This function can be used if subprograms are stored in the part program memory. Subprograms 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 subprograms.

Table 4- 6 M functions for calling subprograms

M function	Function
M98	Subprogram call
M99	End of subprogram

### Subprogram call (M98)

- M98 P nnn mmmm  
m: Program no. (max. 4 digits)  
n: No. of repetitions (max. 4 digits)
- If for example, M98 P21 is programmed, the part program memory is browsed by program name 21.mpf and the subprogram is executed once. To call the subprogram three times, one must program M98 P30021. An alarm is output if the specified program number is not found.
- A nesting of subprograms is possible, up to 16 subprograms are allowed. An alarm is output if more subprogram levels are assigned than is allowed.

### End of subprogram (M99)

A subprogram 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 subprogram 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 start of the part 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.

M99 does not reset the program runtime. An activated workpiece counter is not incremented.

## 4.3 Eight-digit program number

An eight-digit program number selection is activated with the machine data 20734 \$MC\_EXTERN\_FUNCTION\_MASK, Bit 6=1. This function affects M98, G65/66 and M96.

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(Pyxyxxxxx) 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.

### **Interrupt M96**

Does not function in SINUMERIK 802D sl.

\$MC\_EXTERN\_FUNCTION\_MASK, Bit6 = 0

M96 Pxxxx

Addition of program number always to 4 digits with 0

\$MC\_EXTERN\_FUNCTION\_MASK, Bit6 = 1

M96 Pxxxx

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 Measuring functions

### 4.4.1 Rapid lift with G10.6

A retraction position for the rapid lifting of a tool can be activated with G10.6 <Axis position> (e.g., in case of tool breakage). The retraction motion itself is started with a digital signal. The 2nd rapid input of NC is used as the start signal.

Another rapid input (1-8) can also be selected with machine data 10820  
\$MN\_EXTERN\_INTERRUPT\_NUM\_RETRAC (1 - 8).

The interrupt program (ASUB) CYCLE3106.spf must always be available for the rapid retraction with G10.6. If the CYCLE3106.spf is not available in the part program memory, the Alarm 14011 "Program CYCLE3106 not available or not released for processing" is output with G10.6 in a part program block.

The response of the control system after the rapid retraction is defined in ASUB CYCLE3106.spf. If the axes and the spindle are stopped after the rapid retraction, M0 and M5 must be programmed in CYCLE3106.spf. If CYCLE3106.spf is a dummy program that contains only M17, the part program is continued without any interruption after the rapid retraction.

If the rapid retraction is activated with the programming G10.6 <Axis position>, then the change in the input signal of the 2nd NC rapid input from 0 to 1 aborts the current movement and the position programmed in the G10.6 block is moved at rapid traverse. Here, the positions are approached as absolute or incremental, as programmed in the G10.6 block.

The function is deactivated with G10.6 (without position specification). Rapid retraction via the input signal of the 2nd rapid NC input is blocked.

#### Restrictions

Only one axis can be programmed for rapid retraction.

### 4.4.2 Delete measuring with 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 the 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.

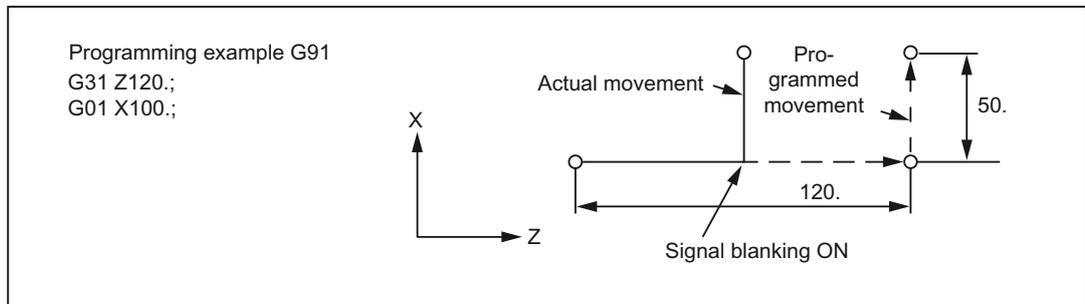


Figure 4-33 Programming example

### 4.4.3 Measuring with G31, P1 - P4

The function G31 P1 (.. P4) is different from G31 in that different inputs for the measuring signal can be selected with P1 to P4. Several inputs can also be monitored on a rising edge of a measuring signal simultaneously. The assignment of the inputs to the addresses P1 to P4 is defined through machine data.

#### Format

G31 X... Y... Z... F... P... ;

X, Y, Z: End point

F...: Feedrate

P...: P1 - P4

#### Explanation

The digital inputs are assigned to the Addresses P1 to P4 via machine data as follows:

P1: \$MN\_EXTERN\_MEAS\_G31\_P\_SIGNAL[0]

P2: \$MN\_EXTERN\_MEAS\_G31\_P\_SIGNAL[1]

P3: \$MN\_EXTERN\_MEAS\_G31\_P\_SIGNAL[2]

P4: \$MN\_EXTERN\_MEAS\_G31\_P\_SIGNAL[3]

Explanations for selection (P1, P2, P3 or P4) can be found in the documentation of your machine manufacturer.

### 4.4.4 Interrupt program with M96/M97 (ASUB)

#### M96

With M96 P<program number> a subprogram can be defined as an interrupt routine.

The start of this program is triggered by an external signal. For starting the interrupt routine always the 1st rapid NC input is used from the 8 inputs available in the Siemens mode. With the machine data 10818 \$MN\_EXTERN\_INTERRUPT\_NUM\_ASUP another rapid input (1-8) can also be selected.

## Format

M96 Pxxxx ;Activation of program interruption  
M97 ;Deactivation of program interruption

With that, the shell cycle CYCLE396 is called first when the interrupt is triggered and this calls the interrupt program programmed with Pxxxx in the ISO mode. At the end of the shell cycle the machine data 10808 \$MN\_EXTERN\_INTERRUPT\_BITS\_M96, bit 1 is evaluated and positioned with REPOS on the interruption point or continued with the next block.

## M97

The start of the interrupt routine is suppressed with M97. Only after the next activation with M96 the interrupt routine can be started with the external signal.

In case the interrupt program programmed with M96 Pxx is to be called directly with the interrupt signal (without the intermediate step with CYCLE396), then the machine data 20734 \$MC\_EXTERN\_FUNCTION\_MASK bit 10 must be set. The subprogram programmed with Pxx is then called in Siemens mode during a signal change from 0 -> 1.

The M function numbers for the interrupt function are set via the machine data. With the 10804 \$MN\_EXTERN\_M\_NO\_SET\_INT the M number is defined for activating an interrupt routine, the machine data 10806 \$MN\_EXTERN\_M\_NO\_DISABLE\_INT defines the M number for suppressing an interrupt routine.

Only such M functions may be used, which are not reserved for the standard M functions. The default setting for the M functions is M96 and M97. In order to activate the functions, bit 0 must be set in the machine data 10808 \$MN\_EXTERN\_INTERRUPT\_BITS\_M96. The M functions are then not output to the PLC. If the bit 0 is not set, the M functions are then interpreted as normal auxiliary functions.

After the end of the interrupt program, the execution leads, by default, to the end position of the part program block following the interruption program. If the part program is to be executed further from the interruption point, an REPOS instruction must be present at the end of the interrupt program e.g. REPOSA. For this, the interrupt program must be written in the Siemens mode.

The M functions for activating and deactivating an interrupt program must be present alone in the block. If other addresses, except "M" and "P", are programmed in the block, the alarm 12080 (syntax error) is output.

## Machine data

The behavior of the function interrupt program can be determined with the following machine data:

MD10808 \$MN\_EXTERN\_INTERRUPT\_BITS\_M96:

- Bit 0 = 0  
No interrupt program possible, M96/M97 are normal M functions.
- Bit 0 = 1  
Activation of an interrupt program allowed with M96/M97.
- Bit 1 = 0  
The part program is processed further with the end position of the next block after the interruption block (REPOSL RMEBL).
- Bit 1 = 1  
The part program is continued after the position of interruption.  
(REPOSL RMEBL)
- Bit 2 = 0  
The interrupt signal interrupts the current block and starts the interrupt routine.
- Bit 2 = 1  
The interrupt routine is started only at the end of the block.
- Bit 3 = 0  
Upon the occurrence of an interrupt signal the execution cycle is interrupted immediately.
- Bit 3 = 1  
The interrupt program is started only at the end of the execution cycle (evaluation in the shell cycle).

Bit 3 is evaluated in the shell cycle and the cycle sequence is adapted accordingly.

Bit 1 is evaluated in the shell cycle CYCLE396.

In case the interrupt program is not called via the shell cycle CYCLE396 (\$MC\_EXTERN\_FUNCTION\_MASK, bit 10 =1), bit 1 must be evaluated; if bit 1 = TRUE, a positioning must be made with REPOSL RMIBL on the interruption point, else positioning must be made with REPOSL RMEBL on the block end point.

Example:

```
N100 M96 P1234      ;Activate ASUB 1234.spf. In the case of a rising edge of
                    ;1st rapid input, the program
                    ;1234.spf is started
"
"
N3000 M97           ;Deactivation of the ASUB
```

Before the call of the interrupt program no rapid lift (LIFTFAST) is performed. With the increasing edge of the interrupt signal, depending on MD10808 \$MN\_EXTERN\_INTERRUPT\_BITS\_M96, the interrupt program is started immediately.

### Restrictions

The interrupt routine is treated as a normal subprogram. This means that at least one subprogram level must be free, in order to execute the interrupt routine. (16 program levels are available plus 2 levels, which are reserved for ASUB, interrupt programs).

The interrupt routine is started only in case of an edge change of the interrupt signal from 0 to 1. If the interrupt signal remains permanently at 1, the interrupt routine is then no longer restarted.

## 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 G68 are modal and are deselected again with G67.

### 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).

### 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.

Table 4- 7 Format for calling a macro program

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 the 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.

Table 4- 8 The P and L command

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 the 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 the 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.

Table 4- 9 Modal call conditions

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 address- and system variables**

Table 4- 10 Interrelation between addresses and variables and addresses that can be used to call commands

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 address- 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]

Table 4- 11 Interrelation between addresses and variables and addresses that can be used to call commands

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]

Interrelation between addresses and variables	
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.

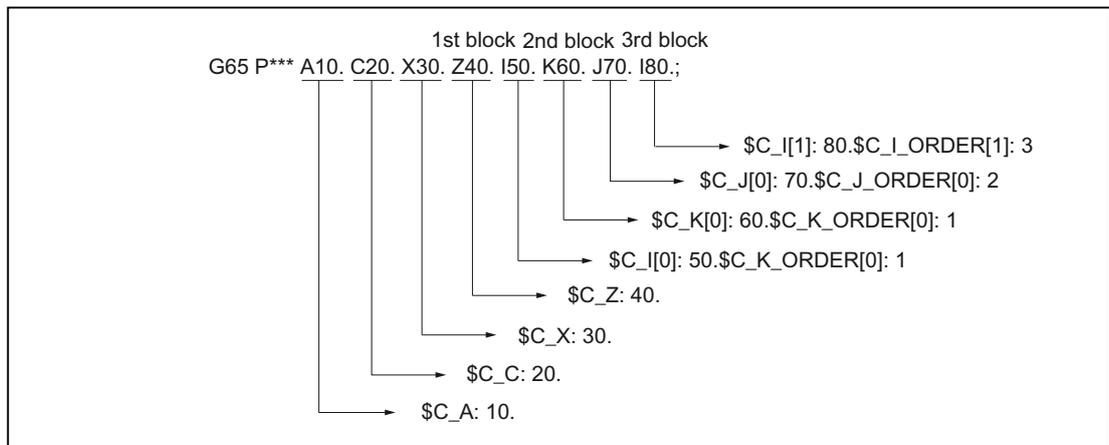


Figure 4-34 Example of entering an argument

**Execution of macro programs in the Siemens and ISO modes**

A called macro program can be called either in the Siemens mode or in the 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 the Siemens mode is conducted. If this instruction is missing, the processing is done in the ISO mode.

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

To read transfer parameters in a macro program executed in the ISO mode, one must first change over to the 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
```

Tool Macro program in the Siemens mode:

```
_N_0010_SPF:
PROC 0010 ; Changeover to the 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 the ISO mode:

```
_N_0010_SPF:
G290; Changeover to the 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 the ISO mode,
N15 M3 G54 T1
N20
...
N80 M99
```

## 4.6 Special functions

### 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 (see Siemens Programming Manual, Chapter Pre-compiling and machine data \$MN\_PREPROCESSING\_LEVEL).

#### 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 the 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.

### 4.6.2 Multi-edge turning

With the polygon turning, workpieces with multiple edges can be produced by coupling two spindles.

With the programming syntax G51.2 Q.. P.. R.. the synchronous spindle coupling is activated. The transformation ratio from leading spindle to the following spindle is defined with the parameters Q and P. If the coupling is to be activated with an angular offset from following spindle and leading spindle, the angular difference is programmed with the address R.

However, no exact edges can be made using multi-edge turning. Typical applications are screw heads of square or hexagonal screws or nuts.

With the programming of G51.2 always the 1st spindle in the channel is defined as the leading spindle and the 2nd spindle as the following spindle. Setpoint linkage is selected as the type of coupling.

**References:**

/FB/ Function Manual - Extended Functions, S3 and

/PGA/, Programming Manual Advanced, Chapter "Synchronous Spindles"

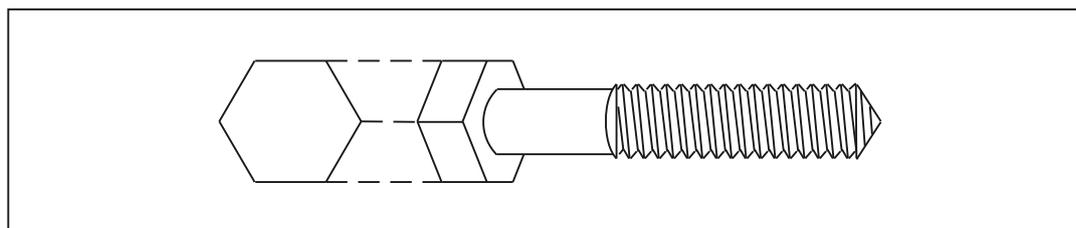


Figure 4-35 Hexagon bolt

**Format**

G51.2 P...Q...;

**P, Q:** Rotation ratio

The direction of the 2nd spindle is specified with the sign before the address character Q.

**Example**

```
G00 X120.0 Z30.0 S1200.0 M03      ; Setting the workpiece rotation speed to
                                   1,200 RPM
G51.2 P1 Q2                        ; Start tool rotation (2,400 RPM)
G01 X80.0 F10.0                    ; Infeed X axis
G04 X2.                             ;
G00 X120.0                          ; Return X axis
G50.2                               ; Stopping the tool rotation
M05                                  ; Spindle stop
```

G50.2 and G51.2 may not be specified together in a block.

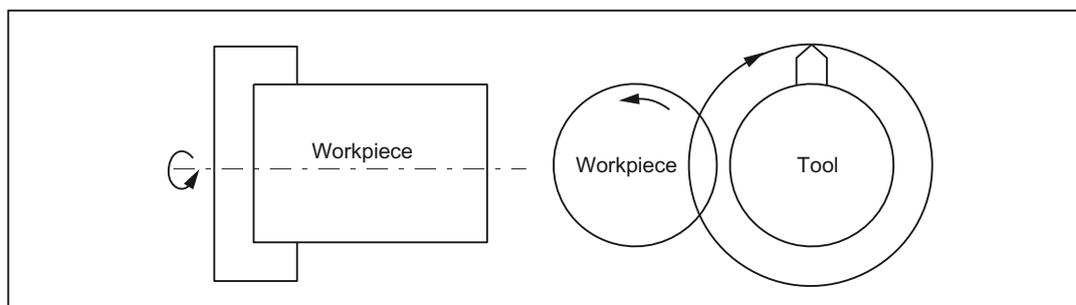


Figure 4-36 Multi-edge turning

### 4.6.3 Compressor in the 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 the Siemens mode, even linear blocks in the 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.), then 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
```

#### 4.6.4 Switchover modes for DryRun and skip levels

Changeover of the skip levels (DB21.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 DB21.DBB0.BIT6) 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 DB21.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

##### Changing the skip levels

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: **Notice! This will also be active "sometime" after the changeover of the DryRun mode!**

### 4.6.5 Interrupt program with M96, M97

#### M96

A subprogram can be defined as an interrupt routine with the M96 P<Program No.>.

The start of this program is triggered by an external signal. To start the interrupt routine, the 1st rapid NC input is used from among the eight inputs available in the Siemens mode.

Another rapid input (1 to 8) can also be selected with MD10818

\$MN\_EXTER\_INTERRUPT\_NUM\_ASUP.

#### Format

M96 Pxxxx	;Activation of program interrupt
M97	;Deactivation of program interrupt

M97 and M96 P\_ must be alone in the block.

So that on triggering the interrupt, the cover cycle CYCLE396 is called first and it calls the interrupt program programmed with Pxxxx in the ISO mode. At the end of the cover cycle, the machine data 10808 \$MN\_EXTERN\_INTERRUPT\_BITS\_M96, Bit 1 is evaluated and either positioned on the interruption point with REPOS or continued with the next block.

#### End of interruption (M97)

The interrupt program is deactivated with M97. Only after the next activation with M96 can the interrupt routine be started with the external signal.

If the interrupt program programmed with M96 Pxx is to be called directly with the interrupt signal (without intermediate step with CYCLE396), then machine data 20734 \$MC\_EXTERN\_FUNCTION\_MASK, Bit 10 must be set. The subprogram programmed with Pxx is then called in the Siemens mode during a signal change from 0 -> 1.

The M function numbers for the interrupt function are set through machine data. Machine data 10804 \$MN\_EXTERN\_M\_NO\_SET\_INT is used to determine the M number for activating an interrupt routine, machine data 10806 \$MN\_EXTERN\_M\_NO\_DISABLE\_INT is used to determine the M number for suppressing an interrupt routine.

Only the M functions not reserved for standard M functions can be used. The default of the M functions is M96 and M97. To activate the function, one must set bit 0 in machine data 10808 \$MN\_EXTERN\_INTERRUPT\_BITS\_M96. The M functions are not output to the PLC. The M functions are interpreted as normal auxiliary functions if Bit 0 is not set.

At the end of the interrupt program, one normally traverses to the end position of the part program block following the interruption block. If the part program is to be processed further from the interruption point, there must be a REPOS instruction at the end of the interrupt program, e.g. REPOSA. For this, the interrupt program must be written in the Siemens mode.

The M function for activating and deactivating an interrupt program must be alone in the block. The system issues Alarm 12080 (syntax error) if addresses other than "M" and "P" are programmed in the block.

## Machine data

The response of the interrupt program function can be determined from the following machine data:

MD10808 \$MN\_EXTERN\_INTERRUPT\_BITS\_M96:

Bit 0 = 0

Interrupt program is not possible as M96/M97 are normal M functions.

Bit 0 = 1

Activation of an interrupt program with M96/M97 is allowed.

Bit 1 = 0

The part program is processed further with the end position of the block immediately after the interruption block (REPOSL RMEBL).

Bit 1 = 1

The part program is continued from the interruption position (REPOSL RMIBL).

Bit 2 = 0

The interrupt signal interrupts the current block immediately and starts the interrupt routine.

Bit 2 = 1

The interrupt routine is started only at the end of the block.

Bit 3 = 0

The execution cycle is interrupted immediately after an interrupt signal arrives.

Bit 3 = 1

The interrupt program is started only at the end of the execution cycle (evaluation in the shell cycles).

Bit 3 is evaluated in the shell cycles and the cycle sequence is adapted accordingly.

Bit 1 is evaluated in cover cycle CYCLE396.

If the interrupt program is not called via the cover cycle CYCLE396, (\$MC\_EXTERN\_FUNCTION\_MASK, Bit 10 = 1) must be evaluated with Bit 1. If Bit 1 = TRUE, REPOSL RMIBL must be used for positioning on the interruption point, otherwise REPOSL RMEBL must be used for positioning on the block end position.

Example:

N100 M96 P1234	;Activate ASUB 1234.spf. In the case of a rising edge of ;1st rapid input, the program ;1234.spf is started
....	
....	
N300 M97	;Deactivation of ASUB

## **Restrictions**

The interrupt routine is treated as a normal subprogram. In other words, to be able to execute interrupt routines, at least one subprogram level must be free. (16 program levels are available, plus two levels that are reserved for the ASUB interrupt programs.)

The interrupt routine is started only during an edge change of the interrupt signal from 0 to 1. If the interrupt signal remains permanently on 1, then the interrupt program is not restarted any more.

# A

## Abbreviations

<b>ASCII</b>	American Standard Code for Information Interchange
<b>ASUB</b>	Asynchronous subprogram
<b>AV</b>	Job planning
<b>BA</b>	Operating mode
<b>BCD</b>	Binary Coded Decimals
<b>BCS</b>	Basic coordinate system
<b>BIN</b>	Binary files
<b>BP</b>	Basic program
<b>C Bus</b>	Communication bus
<b>CAD</b>	Computer-Aided Design
<b>CAM</b>	Computer-Aided Manufacturing
<b>CNC</b>	Computerized Numerical Control
<b>COM</b>	Communication
<b>COR</b>	Coordinate rotation
<b>CPU</b>	Central Processing Unit
<b>CR</b>	Carriage Return
<b>CRC</b>	Cutter radius compensation (tool radius compensation)
<b>CSF</b>	Function plan (PLC programming method)
<b>CTS</b>	Clear To Send (clear to send messages for serial data interfaces)

<b>CUTOM</b>	Cutter radius Compensation: Tool radius compensation
<b>DB</b>	Data Block in the PLC
<b>DBB</b>	Data Block Byte in the PLC
<b>DBW</b>	Data Block Word in the PLC
<b>DBX</b>	Data block bit in the PLC
<b>DC</b>	Direct Control: Movement of the rotary axis via the shortest path to the absolute position within one revolution.
<b>DCE</b>	Data transmission equipment
<b>DDE</b>	Dynamic Data Exchange
<b>DIO</b>	Data Input/Output
<b>DIR</b>	Directory
<b>DLL</b>	Dynamic Link Library: Module which a program can access at runtime. Often contains program sections that are required by different programs.
<b>DOE</b>	Data transmission equipment
<b>DOS</b>	Disk Operating System
<b>DPM</b>	Dual-Port Memory
<b>DPR</b>	Dual-Port RAM: Dual port write/read-only memory
<b>DRAM</b>	Dynamic Random Access Memory: Dynamic write/read-only memory
<b>DRF</b>	Differential Resolver Function (handwheel)
<b>DRY</b>	Dry Run: Dry run feedrate
<b>DSB</b>	Decoding Single Block
<b>DTE</b>	Data Terminal Equipment
<b>DW</b>	Data word

<b>EIA code</b>	Special punched-tape code, number of holes per character always odd
<b>ENC</b>	Encoder: Actual value encoder
<b>EPROM</b>	Erasable Programmable Read Only Memory
<b>FB</b>	Function block
<b>FC</b>	Function Call: Function block in the PLC
<b>FDB</b>	Product database
<b>FDD</b>	Floppy Disk Drive: Diskette drive
<b>FDD</b>	Feed drive
<b>FEPROM</b>	Flash-EPROM: Read and write memory
<b>FIFO</b>	First In First Out: Memory that works without address specification and whose data is read in the same order in which it is stored.
<b>FM</b>	Function module
<b>FM-NC</b>	Function Module - Numerical Control
<b>FPU</b>	Floating Point Unit
<b>FRA</b>	Frame block
<b>FRAME</b>	Data block (FRAME)
<b>FST</b>	Feed Stop
<b>GUD</b>	Global User Data
<b>Hardware</b>	Hardware
<b>HD</b>	Hard Disk
<b>HEX</b>	Abbreviation for hexadecimal number
<b>HMI</b>	Human Machine Interface: Operator functionality of SINUMERIK for operation, programming and simulation.

## Abbreviations

---

<b>I</b>	Input
<b>I/O</b>	Input/output
<b>IBN</b>	Commissioning
<b>ICA</b>	Interpolatory Compensation
<b>IF</b>	Pulse enable for drive modules
<b>IK (GD)</b>	Implicit communication (global data)
<b>IM</b>	Interface Module
<b>IMR</b>	Interface Module Receive: Interface module for receiving data
<b>IMS</b>	Interface Module Send: Interface module for transmit mode
<b>INC</b>	Increment
<b>INI</b>	Initializing Data
<b>IPO</b>	Interpolator
<b>IS</b>	Interface Signal
<b>ISO code</b>	Special punched-tape code, number of holes per character always even
<b>JOG</b>	Jogging: Setup mode
<b>K1 .. K4</b>	Channel 1 to channel 4
<b>Kv</b>	Loop gain factor
<b>LAD</b>	Ladder logic (PLC programming method)
<b>LEC</b>	Leadscrew error compensation
<b>LR</b>	Position controller
<b>LUD</b>	Local User Data
<b>MB</b>	Megabyte

<b>MC</b>	Measuring circuit
<b>MCP</b>	Machine Control Panel
<b>MCS</b>	Machine coordinate system
<b>MD</b>	Machine data
<b>MDA</b>	Manual Data Automatic: Manual input
<b>Mode group</b>	Mode group
<b>MPF</b>	Main Program File: NC part program (main program)
<b>MSD</b>	Main Spindle Drive
<b>NC</b>	Numerical Control
<b>NCK</b>	Numerical Control Kernel: with block preparation, traversing range, etc.
<b>NCU</b>	Numerical Control Unit: Hardware unit of NCK
<b>NURBS</b>	Non-Uniform Rational B-Spline: Rational B-spline curves
<b>O</b>	Output
<b>OB</b>	Organization block in the PLC
<b>OEM</b>	Original Equipment Manufacturer: Manufacturer whose products are marketed under an external company name.
<b>OP</b>	Operator Panel: Operating equipment
<b>OPI</b>	Operator Panel Interface: Operator panel front connection
<b>P-Bus</b>	Peripheral Bus
<b>PC</b>	Personal Computer
<b>PCIN</b>	Name of the SW for data exchange with the control
<b>PCMCIA</b>	Personal Computer Memory Card International Association: Memory plug-in card standardization

<b>PG</b>	Programming device
<b>PLC</b>	Programmable Logic Control: Interface control
<b>PLC</b>	Programmable logic controller
<b>PMS</b>	Position measuring system
<b>RAM</b>	Random Access Memory: Data memory that can be read and written to
<b>RD</b>	Line Feed
<b>REF</b>	"Reference point approach" function
<b>REPOS</b>	"Reposition" function
<b>ROV</b>	Rapid override: Rapid traverse override
<b>RPA</b>	R Parameter Active: Memory area in the NCK for R-NCK for R parameter numbers
<b>RPY</b>	Roll Pitch Yaw: Rotation type of a coordinate system
<b>RTS</b>	Clear To Send (clear to send message for serial data interfaces, activate sending part, control signal from serial data interfaces)
<b>SBL</b>	Single Block
<b>SD</b>	Setting data
<b>SDB</b>	System Data Block
<b>SEA</b>	Setting Data Active: Identifier (file type) for setting data
<b>SFB</b>	System function block
<b>SFC</b>	System function call
<b>SK</b>	Softkey
<b>SKP</b>	Skip block
<b>SM</b>	Stepper motor

<b>SPF</b>	Sub Routine File: Subprogram
<b>SRAM</b>	Static read-only memory (battery buffered)
<b>SRT</b>	Transformation ratio
<b>SS</b>	Interface signal
<b>SSI</b>	Serial Synchronous Interface
<b>STL</b>	Statement list
<b>SW</b>	Software
<b>SYF</b>	System Files
<b>T</b>	Tool
<b>TC</b>	Tool change
<b>TEA</b>	Testing Data Active: Designation for machine data
<b>TLC</b>	Tool length compensation
<b>TNRC</b>	Tool nose radius compensation
<b>TO</b>	Tool Offset
<b>TO</b>	Tool change
<b>TOA</b>	Tool Offset Active: Identifier (file type) for tool offsets
<b>TRANSMIT</b>	Transform Milling Into Turning: Coordinate conversion in turning machine for milling
<b>TRC</b>	Tool radius compensation
<b>UFR</b>	User Frame: Work offset
<b>UI</b>	User interface
<b>UP</b>	Subprogram
<b>V.24</b>	Serial interface (definition of the interchange lines between DTE and DCE)

## Abbreviations

---

<b>WCS</b>	Workpiece coordinate system
<b>WOP</b>	Workshop-oriented programming
<b>WPD</b>	Workpiece Directory
<b>ZO</b>	Work offset
<b>ZOA</b>	Zero Offset Active: Identifier (file type) for work offset data

# G code table

# B

Appendix 1 describes the G codes and their functions.

Table B- 1 G code table

G code		Description	System A	System C
<b>Group 1</b>				
G00 <sup>1)</sup>	1	Rapid traverse	G00	G00
G01	2	Linear motion	G01	G01
G02	3	Circle/helix in clockwise direction	G02	G02
G03	4	Circle/helix in counterclockwise direction	G03	G03
G33	5	Thread cutting with constant lead	G33	G33
G34	9	Thread cutting with variable lead	G34	G34
G35		Convex thread in the clockwise direction	G35	G35
G36		Convex thread in the counter-clockwise direction	G36	G36
G77	6	Longitudinal turning cycle	G90	G20
G78	7	Thread cutting cycle	G92	G21
G79	8	Face turning cycle	G94	G24
<b>Group 2</b>				
G96	1	Constant cutting rate on	G96	G96
G97 <sup>1)</sup>	2	Constant cutting rate off	G97	G97
<b>Group 3</b>				
G90 <sup>1)</sup>	1	Absolute programming	--	G90
G91	2	Incremental programming	--	G91
<b>Group 4</b>				
G68	1	Double slide/turret on	G68	G68
G69 <sup>1)</sup>	2	Double slide/turret off	G69	G69
<b>Group 5</b>				
G94	1	Linear feedrate in [mm/min, inch/min]	G98	G94
G95 <sup>1)</sup>	2	Revolutional feedrate in [mm/rev, inch/rev]	G99	G95
<b>Group 6</b>				
G20 <sup>1)</sup>	1	Inch input system	G20	G70
G21	2	Metric input system	G21	G71
<b>Group 7</b>				
G40 <sup>1)</sup>	1	Deselection of cutter radius compensation	G40	G40
G41	2	Compensation left of contour	G41	G41
G42	3	Offset to right of contour	G42	G42
<b>Group 8</b>				

G code		Description	System A	System C
<b>Group 9</b>				
G22	1	Working area limitation, protection zone 3 on	G22	G22
G23	<sup>1)</sup> 2	Working area limitation, protection zone 3 off	G23	G23
<b>Group 10</b>				
G80	<sup>1)</sup> 1	Drilling cycle off	G80	G80
G83		Front face deep hole drilling	G83	G83
G84	3	Front face tapping	G84	G84
G85	4	Front face drilling cycle	G85	G85
G87	5	Side surface deep hole drilling	G87	GG87
G88	6	Side surface tapping	G88	G88
G89	7	Side surface drilling	G89	G89
<b>Group 11</b>				
G98	<sup>1)</sup> 1	Return to starting point in drilling cycles	--	G98
G99	2	Return to point R in drilling cycles	--	G99
<b>Group 12</b>				
G66	1	Macro module call	G66	G67
G67	<sup>1)</sup> 2	Delete macro module call	G67	G67
<b>Group 13</b>				
<b>Group 14</b>				
G54	<sup>1)</sup> 1	Selecting work offset	G54	G54
G55	2	Selecting work offset	G55	G55
G56	3	Selecting work offset	G56	G56
G57	4	Selecting work offset	G57	G57
G58	5	Selecting work offset	G58	G58
G59	6	Selecting work offset	G59	G59
G54		Extended work offset	x	x
P{1...48}	1			
G54.1	7	Extended work offset	G54.1	G54.1
G54 P0	1	External work offset	x	x
<b>Group 15</b>				
<b>Group 16</b>				
G17	1	XY plane	G17	G17
G18	<sup>1)</sup> 2	ZX plane	G18	G18
G19	3	YZ plane	G19	G19
<b>Group 17</b>				

G code		Description	System A	System C
<b>Group 18 (non-modal effective)</b>				
G04	1	Dwell time in [s] or spindle revolutions	G04	G04
G05	20	High-speed cycle cutting	G05	G05
G05.1	22	High-speed cycle -> Call CYCLE305	G05.1	G05.1
G07.1	18	Cylindrical interpolation	G07.1	G07.1
G10	2	Write work offset / tool offset	G10	G10
G10.6	19	Rapid lift on/off	G10.6	G10.6
G28	3	Approaching 1st reference point	G28	G28
G30	4	Approaching 2nd/3rd/4th reference point	G30	G30
G30.1	21	Reference point position	G30.1	G30.1
G31	5	Measuring with touch-trigger probe	G31	G31
G52	6	Programmable work offset	G52	G52
G53	17	Approach position in machine coordinate system	G53	G53
G60	24	Directed positioning	G60	G60
G65	7	Macro call	G65	G65
G70	8	Finishing cycle	G70	G72
G71	9	Stock removal cycle, longitudinal axis	G71	G73
G72	10	Stock removal cycle transverse axis	G72	G74
G73	11	Contour repetition	G73	G75
G74	12	Deep-hole boring and grooving in the longitudinal axis (Z)	G74	G76
G75	13	Deep-hole boring and grooving in the transverse axis (X)	G75	G77
G76	14	Multiple thread cutting cycle	G76	G78
G92	15	Set actual value, spindle speed limitation	G50	G92
G92.1	23	Delete actual value, reset the WCS	G50.3	G92.1
<b>Group 20</b>				
G50.2 <sup>1)</sup>	1	Multi-edge turning OFF	G50.2	G50.2
G51.2	2	Multi-edge turning ON	G51.2	G51.2
<b>Group 21</b>				
G13.1 <sup>1)</sup>	1	TRANSMIT OFF	G13.1	G13.1
G12.1	2	TRANSMIT ON	G12.1	G12.1
<b>Group 22</b>				
<b>Group 25</b>				
<b>Group 31</b>				
G290 <sup>1)</sup>	1	Select Siemens mode	G290	G290
G291	2	Select ISO dialect mode	G291x	G291

---

**Note**

In general, the G functions mentioned in <sup>1)</sup> are defined by the NC during activation of the control system or during RESET.

---

# Data Description



## C.1 General machine/setting data

### Note

All the machine data described here relates to the SINUMERIK 840D sl. Please use the associated List Manuals for the SINUMERIK 828D control system.

<b>10604</b>	<b>WALIM_GEOAX_CHANGE_MODE</b>		
SD number	Working area limitation during switchover of geometry axes		
Default setting: 0	Min. input limit: 0	Max. input limit: 1	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: BYTE			
Meaning:	<p>This machine data defines whether a possibly active working area limitation is retained or deactivated during a geometry axis interchange.</p> <p>The MD is bit-coded with the following significance:</p> <p>Bit 0=0: Working area limitation is deactivated upon geo axis interchange.</p> <p>Bit 0=1: Active working area limitation remains active upon a geo axis interchange.</p>		

<b>10615</b>	<b>NCBFRAME_POWERON_MASK</b>		
MD number	Delete global basic frames during Power On		
Default setting: 0	Min. input limit: 0	Max. input limit: 0	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: DWORD			
Meaning:	<p>This machine data is used to define whether global basic frames are deleted during Power On Reset.</p> <p>The individual basic frames can be selected separately.</p> <p>Bit 0 corresponds to basic frame 0, bit 1 to basic frame 1, etc.</p> <p>0: Basic frame is retained at Power On</p> <p>1: Basic frame is deleted at Power ON.</p>		

<b>10652</b>	<b>CONTOUR_DEF_ANGLE_NAME</b>		
MD number	Settable name for the angle in the contour brief description		
Default setting: "ANG"	Min. input limit: -	Max. input limit: -	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: STRING			
Meaning:	<p>The setting is effective only for Siemens G code programming, i.e. G290.</p> <p>The name, under which the angle is programmed in the contour brief description, can be set. In this way, for example, identical programming can be implemented in different language modes:</p> <p>If "A" is specified as name, then the angle is specified the same way in Siemens programming as in ISO dialect.</p> <p>The name must be unique, i.e. no axes, variables, macros, etc. may exist with the same name.</p>		

<b>10654</b>	<b>RADIUS_NAME</b>		
MD number	Settable name for the radius non-modal in the contour brief description		
Default setting: "RND"	Min. input limit: -	Max. input limit: -	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: STRING			
Meaning:	<p>The name, under which the radius is programmed in the contour brief description, can be set. In this way, for example, identical programming can be implemented in different language modes:</p> <p>If "R" is specified as name, then the radius is specified the same way in Siemens programming as in ISO dialect.</p> <p>The name must be unique, i.e. no axes, variables, macros, etc. may exist with the same name.</p> <p>The setting is effective only for Siemens G code programming, i.e. G290.</p>		

<b>10656</b>	<b>CHAMFER_NAME</b>		
MD number	Settable name for the chamfer in the contour brief description		
Default setting: "CHR"	Min. input limit: -	Max. input limit: -	
Change valid after POWER ON	Protection level: 2/7	Unit: -	
Data type: STRING			
Meaning:	<p>The name, under which the chamfer is programmed in the contour brief description, can be set. In this way, for example, identical programming can be implemented in different language modes:</p> <p>If "C" is specified as name, then the radius is specified the same way in Siemens programming as in ISO dialect.</p> <p>The name must be unique, i.e. no axes, variables, macros, etc. may exist with the same name.</p> <p>The setting is effective only for Siemens G code programming, i.e. G290.</p> <p>The chamfer acts in the original direction of movement. Alternatively, the chamfer length can be programmed under the CHF identifier.</p>		

<b>10704</b>	<b>DRYRUN_MASK</b>		
MD number	Activation of dry run feedrate		
Default setting:	Min. input limit: -	Max. input limit: -	
Change becomes effective after	Protection level:	Unit: -	
Data type: BYTE			
Meaning:	<p>DRYRUN_MASK == 0 Dry run may be activated and deactivated only at end of block.</p> <p>DRYRUN_MASK == 1 The activation/deactivation of the dry-run feedrate is also possible during a program execution. <b>Notice:</b> After the activation of the dry run feedrate, the axes are stopped for the duration of the reorganization operation.</p> <p>DRYRUN_MASK == 2 Dry run can be switched on and off in each phase, and the axes are not stopped. <b>Notice:</b> The function only becomes effective with a block which comes "later" in the program run! The function is effective from the next (implicit) stop reset block.</p>		

<b>10706</b>	<b>SLASH_MASK</b>		
MD number	Activation of block skip function		
Default setting: 0	Min. input limit: 0	Max. input limit: 2	
Change becomes effective after	Protection level:	Unit: -	
Data type: BYTE			
Meaning:	<p>SLASH_MASK == 0 Switchover of block skip function is possible only at end of block.</p> <p>SLASH_MASK == 1 In case of SLASH_MASK == 1, the activation of block skip function is also possible during a program execution. <b>Notice:</b> After the activation of block skip function, the axes are stopped for the duration of the reorganization operation.</p> <p>SLASH_MASK == 2 Block switchover is possible in each phase. <b>Notice:</b> The function only becomes effective with a block which comes "later" in the program run! The function is effective from the next (implicit) stop reset block.</p>		

<b>10715</b>	<b>M_NO_FCT_CYCLE[0]</b>		
MD number	M function number for cycle call		
Default setting: -1	Min. input limit: -1	Max. input limit: -	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: DWORD			
Meaning:	<p>M number with which the subprogram is called.</p> <p>The name of the subprogram is stored in \$MN_M_NO_FCT_CYCLE_NAME. If the M function defined by \$MN_M_NO_FCT_CYCLE is programmed in a part program, then the subprogram defined in M_NO_FCT_CYCLE_NAME is started at the end of block.</p> <p>If the M function is reprogrammed in the subprogram, then there is no longer any replacement by a subprogram call.</p> <p>\$MN_M_NO_FCT_CYCLE acts in the Siemens mode G290 as well as in the external language mode G291.</p> <p>A subprogram call must not be superimposed on M functions with predetermined significance.</p> <p>Alarm 4150 is generated in the event of a conflict:</p> <ul style="list-style-type: none"> <li>• M0 to M5,</li> <li>• M17, M30,</li> <li>• M40 to M45,</li> <li>• M function for selecting spindle/axis mode according to \$MC_SPIND_RIGID_TAPPING_M_NR (default M70)</li> <li>• M functions for nibbling/punching as configured in \$MC_NIBBLE_PUNCH_CODE if activated via \$MC_PUNCHNIB_ACTIVATION.</li> <li>• For applied external language (\$MN_MM_EXTERN_LANGUAGE) M19, M96-M99.</li> </ul> <p>Exception: The M functions defined by \$MC_TOOL_CHANGE_M_CODE for the tool change.</p> <p>The subprograms configured with \$MN_M_NO_FCT_CYCLE_NAME and \$MN_T_NO_FCT_CYCLE_NAME must not be effective in one block (part program line) at the same time, i.e. max. one M/T function replacement can be effective per block. Neither an M98 nor a modal subprogram call may be programmed in the block with the M function replacement. A subprogram return jump or end of part program is also not permitted.</p> <p>Alarm 14016 is generated in the case of a conflict.</p>		

<b>10716</b>	<b>M_NO_FCT_CYCLE_NAME[0]</b>		
MD number	Subprogram name for M function replacement		
Default setting: -	Min. input limit: -	Max. input limit: -	
Change valid after POWER ON	Protection level: 2/7		Unit: -
Data type: STRING			
Meaning:	<p>The cycle name is stored in the machine data. This cycle is called if the M function was programmed from the machine data \$MN_M_NO_FCT_CYCLE.</p> <p>If the M function is programmed in a traversing block, then the cycle is executed after the motion.</p> <p>\$MN_M_NO_FCT_CYCLE acts in the Siemens mode G290 as well as in the external language mode G291.</p> <p>If a T number is programmed in the call block, the programmed T number can be queried in the cycle under the variable \$P_TOOL.</p> <p>The subprograms configured with \$MN_M_NO_FCT_CYCLE_NAME and \$MN_T_NO_FCT_CYCLE_NAME must not be effective in one block at the same time, i.e. max. one M/T function replacement can be effective per block. Neither an M98 nor a modal subprogram call may be programmed in the block with the M function replacement. A subprogram return jump or end of part program is also not permitted.</p>		

<b>10717</b>	<b>T_NO_FCT_CYCLE_NAME</b>		
MD number	Name of tool change cycle for T function replacement		
Default setting: -	Min. input limit: -	Max. input limit: -	
Changes effective after POWER ON	Protection level: 2/7		Unit: -
Data type: STRING			
Meaning:	<p>Cycle name for tool change routine during call via T function. If a T function is programmed in a part program block, then the subprogram defined in T_NO_FCT_CYCLE_NAME is called at the end of block.</p> <p>The programmed T number can be queried in the cycle via the system variables \$C_T/\$C_T_PROG as decimal value and via \$C_TS/\$C_TS_PROG as string (only with tool management).</p> <p>\$MN_T_NO_FCT_CYCLE_NAME acts in the Siemens mode G290 as well as in the external language mode G291.</p> <p>\$MN_M_NO_FCT_CYCLE_NAME and \$MN_T_NO_FCT_CYCLE_NAME must not be effective in one block at the same time, i.e. max. one M/T function replacement can be effective per block.</p> <p>Neither an M98 nor a modal subprogram call may be programmed in the block with the T function replacement. A subprogram return jump or end of part program is also not permitted. Alarm 14016 is generated in the case of a conflict.</p>		

C.1 General machine/setting data

<b>10760</b>	<b>G53_TOOLCORR</b>		
MD number	Method of functioning with G53, G153 and SUPA		
Default setting: 2	Min. input limit: 2	Max. input limit: 4	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: BYTE			
Meaning:	<p>This MD acts in the Siemens mode and in the external language mode.</p> <p>This machine data defines whether the tool length and the tool radius compensation are to be suppressed for the language commands G53, G153 and SUPA.</p> <p>0 = G53/G153/SUPA is a non-modal suppression of work offsets; active tool length and tool radius compensations are retained.</p> <p>1 = G53/G153/SUPA is a non-modal suppression of work offsets and active tool length and tool radius compensation.</p>		

<b>10800</b>	<b>EXTERN_CHAN_SYNC_M_NO_MIN</b>		
MD number	First M number for channel synchronization		
Default setting: -1	Min. input limit: 100	Max. input limit:	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: DWORD			
Meaning:	Smallest M number of the M range that is reserved for the channel synchronization.		
MD cannot be changed for SINUMERIK 802D sl.			

<b>10802</b>	<b>EXTERN_CHAN_SYNC_M_NO_MAX</b>		
SD number	Last M number for channel synchronization		
Default setting: -1	Min. input limit: 100	Max. input limit:	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: DWORD			
Meaning:	<p>Biggest M number of the M range that is reserved for the channel synchronization.</p> <p>The M range should be a maximum of 10*channel number (2 channels = 20 M numbers). If a larger range is defined, alarm 4170 is output.</p>		

<b>10804</b>	<b>EXTERN_M_NO_SET_INT</b>		
MD number	M function for ASUB activation		
Default setting: 96	Min. input limit: 0	Max. input limit:	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: DWORD			
Meaning:	M function number with which an interrupt program is activated in the ISO_T/M mode (ASUB).		

<b>10806</b>	<b>EXTERN_M_NO_DISABLE_INT</b>		
MD number	M function for ASUB deactivation		
Default setting: 97	Min. input limit: 0	Max. input limit:	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: DWORD			
Meaning:	M function number with which an interrupt program (ASUB) is deactivated in the ISO-T/M mode.		

<b>10808</b>	<b>EXTERN_INTERRUPT_BITS_M96</b>		
MD number	Interrupt program execution (M96)		
Default setting: 0	Min. input limit: 0	Max. input limit: 8	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: WORD			
Meaning:	<p>By setting different bits, the sequence of the interrupt routine activated with M96 P... can be controlled.</p> <p>Bit 0=0: No interrupt program possible, M96/M97 are normal M functions</p> <p>Bit 0=1: Activation of an interrupt program with M96/M97 is allowed</p> <p>Bit 1=0: Process part program with the end position of the next block after the interruption block</p> <p>Bit 1=1: Further process part program from the interruption position</p> <p>Bit 2=0: The interrupt signal interrupts the current block immediately and starts the interrupt routine</p> <p>Bit 2=1: The interrupt routine is started only at the end of the block</p> <p>Bit 3=0: Interrupt execution cycle in case of an interrupt signal</p> <p>Bit 3=1: Start interrupt program only at the end of the execution cycle</p>		

<b>10810</b>	<b>EXTERN_MEAS_G31_P_SIGNAL</b>		
MD number	Assignment of the measuring inputs for G31 P..		
Default setting: 1	Min. input limit: 0	Max. input limit: 3	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: BYTE			
Meaning:	<p>The machine data defines an assignment of the measuring inputs 1 and 2 to the P numbers programmed with G31 P1 (-P4). The MD is bit-coded. Only bit 0 and bit 1 are evaluated. If e.g. in \$MN_EXTERN_MEAS_G31_P_SIGNAL[1] the bit 0=1, the 1st measuring input is activated with G31 P2. The second measuring input is activated with G31 P4 with \$MN_EXTERN_MEAS_G31_P_SIGNAL[3] = 2.</p> <p>Bit 0=0: Do not evaluate measuring input 1 for G31 P1 (-P4)</p> <p>Bit 0=1: Activate measuring input 1 for G31 P1 (-P4)</p> <p>Bit 1=0: Do not evaluate measuring input 2 for G31 P1 (-P4)</p> <p>Bit 1=1: Activate measuring input 2 for G31 P1 (-P4)</p>		

<b>10812</b>	<b>EXTERN_DOUBLE_TURRET_ON</b>		
MD number	Double turret head with G68		
Default setting:	Min. input limit:	Max. input limit:	
Change becomes effective after	Protection level:	Unit: -	
Data type: BOOLEAN			
Meaning:	<p>The machine data is effective only in case of \$MN_EXTER_CNC_SYSTEM = 2.                  This MD defines whether with G68 a double-slide machining (channel synchronization for 1st and 2nd channel) is to be started or whether the second tool of a double turret (= 2, properly connected tool with the distance defined in the setting data \$SC_EXTERN_DOUBLE_TURRET_DIST) is to be activated.</p> <p>FALSE: Channel synchronization for double-slide machining                  TRUE: Changing the 2nd tool of a double turret (= \$SC_EXTERN_DOUBLE_TURRET_DISTANCE as additive work offset and activate mirroring around Z-axis</p>		

<b>10814</b>	<b>EXTERN_M_NO_MAC_CYCLE</b>		
MD number	Macro call via M function		
Default setting:	Min. input limit:	Max. input limit:	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: DWORD			
Meaning:	<p>M number with which a macro is called.</p> <p>The name of the subprogram is stored in \$MN_EXTERN_M_NO_MAC_CYCLE_NAME[n]. If the M function is programmed in a part program block defined with \$MN_EXTERN_M_NO_MAC_CYCLE[n], the subprogram defined in EXTERN_M_NO_MAC_CYCLE_NAME[n] is started, and all the addresses programmed in the block are written to the associated variables. If the M function is reprogrammed in the subprogram, then there is no longer any replacement by a subprogram call.</p> <p>\$MN_EXTERN_M_NO_MAC_CYCLE_NAME[n] acts only in the external language mode G291.</p> <p>A subprogram call must not be superimposed on M functions with predetermined significance. Alarm 4150 is generated in the event of a conflict:</p> <ul style="list-style-type: none"> <li>• M0 to M5,</li> <li>• M17, M30,</li> <li>• M19,</li> <li>• M40 to M45,</li> <li>• M function for switching over spindle mode / axis mode according to \$MC_SPIND_RIGID_TAPPING_M_NR (default: M70).</li> <li>• M functions for nibbling/punching as configured in \$MC_NIBBLE_PUNCH_CODE if activated via \$MC_PUNCHNIB_ACTIVATION.</li> <li>• For applied external language (\$MN_MM_EXTERN_LANGUAGE), M96 to M99 in addition.</li> <li>• M functions defined through \$MN_M_NO_FCT_CYCLE.</li> </ul> <p><b>Exception:</b> The M functions defined with \$MC_TOOL_CHANGE_M_CODE for the tool change.</p> <p>The subprograms configured with \$MN_EXTERN_M_NO_MAC_CYCLE_NAME[n] must not be effective in one block (part program line) at the same time, i.e. max. one M function replacement can be effective per block. Neither an M98 nor a modal subprogram call may be programmed in the block with the M function replacement. A subprogram return jump or end of part program is also not permitted. Alarm 14016 is generated in the case of a conflict.</p>		

<b>10815</b>	<b>EXTERN_M_NO_MAC_CYCLE_NAME</b>		
MD number	Subprogram name for M function macro call		
Default setting:	Min. input limit:	Max. input limit:	
Changes effective after POWER ON	Protection level:	Unit: -	
Data type: STRING			
Meaning:	Cycle name in case of a call via the M function defined with \$MN_EXTERN_M_NO_MAC_CYCLE[n].		

<b>10816</b>	<b>EXTERN_G_NO_MAC_CYCLE</b>		
MD number	Macro call via G function		
Default setting:	Min. input limit:	Max. input limit:	
Changes effective after POWER ON	Protection level:	Unit: -	
Data type: DOUBLE			
Meaning:	<p>G number with which a macro is called.</p> <p>The name of the subprogram is stored in \$MN_EXTERN_G_NO_MAC_CYCLE_NAME[n].</p> <p>If a G function defined with \$MN_EXTERN_M_NO_MAC_CYCLE[n] is programmed in a part program block, the subprogram defined in EXTERN_M_NO_MAC_CYCLE_NAME[n] is started, and all the addresses programmed in the block are written in the related \$C_xx variable.</p> <p>If a subprogram call is already active via an M/G macro or an M substitution, no subprogram call is executed. If in this case a standard G function is programmed, then it is executed, otherwise alarm 12470 is output.</p> <p>\$MN_EXTERN_G_NO_MAC_CYCLE[n] acts only in the external language mode G291.</p> <p>In one block there may only be one subprogram call. That is, in one block only one M/G function replacement may be programmed and there may be no additional subprogram call (M98) or cycle call in the block.</p> <p>A subprogram return jump or end of part program is also not permitted in the same block. Alarm 14016 is generated in the case of a conflict.</p>		

<b>10817</b>	<b>EXTERN_G_NO_MAC_CYCLE_NAME</b>		
MD number	Subprogram name for G function macro call		
Default setting:	Min. input limit:	Max. input limit:	
Changes effective after POWER ON	Protection level:	Unit: -	
Data type: STRING			
Meaning:	Cycle name in case of a call via the G function defined with \$MN_EXTERN_G_NO_MAC_CYCLE[n].		

<b>10818</b>	<b>EXTERN_INTERRUPT_NUM_ASUP</b>		
MD number	Interrupt number for ASUB start (M96)		
Default setting:	Min. input limit:	Max. input limit:	
Changes effective after POWER ON	Protection level:	Unit: -	
Data type: BYTE			
Meaning:	Number of the interrupt input with which an asynchronous subprogram activated in the ISO mode is started. (M96<Program number>)		

<b>10820</b>	<b>EXTERN_INTERRUPT_NUM_RETRAC</b>		
MD number	Interrupt number for rapid retraction (G10.6)		
Default setting:	Min. input limit:	Max. input limit:	
Changes effective after POWER ON	Protection level:	Unit: -	
Data type: BYTE			
Meaning:	Number of the interrupt input with which a rapid retraction is triggered in the ISO mode to the position programmed with G10.6.		

<b>10880</b>	<b>MM_EXTERN_CNC_SYSTEM</b>		
MD number	External control system, whose programs are to be executed		
Default setting: 0	Min. input limit: 0	Max. input limit: 2	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: WORD			
Meaning:	Selecting the external language 1 = ISO-2: Fanuc0 Milling system 2 = ISO-3: Fanuc0 Turning system Thus the function scope defined in the current Siemens documentation is applicable. This data is evaluated only when the machine data \$MN_MM_EXTERN_LANGUAGE is set.		

<b>10881</b>	<b>MM_EXTERN_GCODE_SYSTEM</b>		
SD number	ISO mode T: G code system		
Default setting: 0	Min. input limit: 0	Max. input limit: 2	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: DWORD			
Meaning:	Determination of the G code system which is to be processed actively in ISO dialect T mode: Value = 0: ISO_T: Code system B Value = 1: ISO_T: Code system A Value = 2: ISO_T: Code system C In order for the shell cycle to work in the correct G code system, the corresponding system must be entered in the GUD variable _ZSFI[39]:		

<b>10882</b>	<b>NC_USER_EXTERN_GCODES_TAB [n]:0...59</b>		
MD number	List of user-specific G commands of an external NC language		
Default setting: -	Min. input limit: -	Max. input limit: -	
Changes effective after POWER ON	Protection level: 2/2	Unit: -	
Data type: STRING			
Meaning:	<p>By default, ISO Dialect T Code B is realized for external programming language. Code A and C are differentiated through the names of the G functions.</p> <p>Via \$MN_NC_USER_EXTERN_GCODES_TAB the G functions can be renamed.</p> <p>G functions for external NC languages can be recoded. The G group and the position within the G group are retained. Only the G functions may be recoded.</p> <p>Up to 30 recodings are possible. Example:</p> <p>\$MN_NC_USER_EXTERN_GCODES_TAB[0]="G20"</p> <p>\$MN_NC_USER_EXTERN_GCODES_TAB[1]="G70"</p> <p>--&gt; G20 is recoded in G70</p> <p>If G70 is already available, an error message appears in NCK reset.</p>		

<b>10884</b>	<b>EXTERN_FLOATINGPOINT_PROG</b>		
MD number	Evaluation of programmed values without decimal point		
Default setting: 1	Min. input limit: 0	Max. input limit: 1	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: BOOLEAN			
Meaning:	<p>This machine data is effective for external programming language, i.e. when MD18800 \$MN_MM_EXTERN_LANGUAGE = 1.</p> <p>This machine data defines how the programmed values without decimal point are evaluated.</p> <p>0: Standard notation: Values without the decimal point are interpreted in internal units IS-B, IS-C (see MD EXTERN_INCREMENT_SYSTEM).</p> <p>Values without decimal point are interpreted in internal units, e.g. X1000 = 1 mm (for 0.001 mm input resolution) X1000.0 = 1000 mm</p> <p>1: PocketCalculator notation: Values without decimal point are interpreted as mm, inch or degree.</p> <p>Values without decimal point are interpreted as mm, inch or degree, e.g. X1000 = 1000 mm X1000.0 = 1000 mm</p>		

<b>10886</b>	<b>EXTERN_INCREMENT_SYSTEM</b>		
MD number	Increment system		
Default setting: 0	Min. input limit: 0	Max. input limit: 1	
Changes effective after POWER ON	Protection level: 2/7		Unit: -
Data type: BOOLEAN			
Meaning:	<p>This machine data is effective for external programming language, i.e. when MD18800 \$MN_MM_EXTERN_LANGUAGE = 1.</p> <p>This machine data defines the increment system that is active:</p> <p>0: Increment system IS-B = 0.001 mm/degree = 0.0001 inch</p> <p>1: Increment system IS-C = 0.0001 mm/degree = 0.00001 inch</p>		

<b>10888</b>	<b>EXTERN_DIGITS_TOOL_NO</b>		
MD number	Number of digits for T number in external language mode		
Default setting: 2	Min. input limit: 2	Max. input limit: 4	
Changes effective after POWER ON	Protection level: 2/7		Unit: -
Data type: BYTE			
Meaning:	<p>The machine data is effective only in case of \$MN_EXTERN_CNC_SYSTEM = 2.</p> <p>Number of digits of tool number in the programmed T value.</p> <p>From the programmed T value, the number of leading digits specified via \$MN_EXTERN_DIGITS_TOOL_NO are interpreted as tool number. The following positions address the offset memory.</p>		

<b>10890</b>	<b>EXTERN_TOOLPROG_MODE</b>		
MD number	Tool change programming in case of external programming language		
Default setting: 0	Min. input limit: 0	Max. input limit: 1	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: BYTE			
Meaning:	<p>Configuration of the programming of the tool change for external programming language:</p> <p>Bit 0 = 0: Acts only in case of \$MN_MM_EXTERN_CNC_LANGUAGE = 2:                      The tool number and the offset number are programmed in the T value.                      \$MN_DIGITS_TOOLNO determines the number of leading digits which form the tool number.                      Example:                      \$MN_DIGITS_TOOL_NO = 2                      T=1234                      ; Tool number 12,                      ; Offset number 34</p> <p>Bit 0 = 1: Acts only in case of \$MN_MM_EXTERN_CNC_LANGUAGE = 2:                      Only the tool number is programmed in the T value.                      Offset number = tool number.                      \$MN_DIGITS_TOOL_NO is irrelevant.                      Example:                      T=12                      ; Tool number 12,                      ; Offset number 12</p> <p>Bit 1 = 0: Acts only in case of \$MN_MM_EXTERN_CNC_LANGUAGE = 2:                      If the number of positions programmed in the T value are the same as the number defined in \$MN_EXTERN_DIGITS_TOOL_NO, then leading 0 are added.</p>		

<b>10890</b>	<b>EXTERN_TOOLPROG_MODE</b>
Meaning:	<p>Bit 1 = 1: Acts only in case of \$MN_MM_EXTERN_CNC_LANGUAGE = 2: If the number of positions programmed in the T value is the same as the number of digits specified in \$MN_EXTERN_DIGITS_TOOL_NO, then the programmed number is applicable as offset and tool number</p> <p>Bit 2 = 0: Acts only in case of \$MN_MM_EXTERN_CNC_LANGUAGE = 2: ISO T offset selection only with D (Siemens Cutting edge number)</p> <p>Bit 2 = 1: Acts only in case of \$MN_MM_EXTERN_CNC_LANGUAGE = 2: ISO T Offset selection only with H (\$TC_DPH[t,d])</p> <p>Bit 3 = 0: Acts only in case of \$MN_MM_EXTERN_CNC_LANGUAGE = 2: Each H number is allowed only once per TOA, except H=0. If the bit 3 is set from 1 -&gt; 0, then no H number may appear multiple times in a TO unit. Else an alarm is output upon the next warm restart.</p> <p>Bit 3 = 1: Acts only in case of \$MN_MM_EXTERN_CNC_LANGUAGE = 2: Each H number is allowed several times per TOA.</p> <p>Bit 6 = 0: Acts only in case of \$MN_MM_EXTERN_CNC_LANGUAGE = 1: Selection of tool length is not possible under the address H</p> <p>Bit 6 = 1: Acts only in case of \$MN_MM_EXTERN_CNC_LANGUAGE = 1: Selection of tool length under the address H</p> <p>Bit 7 = 0: Acts only in case of \$MN_MM_EXTERN_CNC_LANGUAGE = 1: Selection of tool length is not possible under the address D</p> <p>Bit 7 = 1: Acts only in case of \$MN_MM_EXTERN_CNC_LANGUAGE = 1: Selection of tool length under the address D</p> <p>If bit 6 and bit 7 are selected, then the selection is possible under address D or H.</p>

<b>18800</b>	<b>MM_EXTERN_LANGUAGE</b>		
MD number	External language active in the control system		
Default setting: 0	Min. input limit: 0	Max. input limit: 1	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: DWORD			
Meaning:	<p>The corresponding NC language must be activated for processing the part programs of other system manufacturers. Only one external language can be selected. The respective instruction set can be taken from the current documentation.</p> <p>Bit 0 (LSB): Execution of part programs ISO_2 or ISO_3. For coding see \$MN_MM_EXTERN_CNC_SYSTEM (10880)</p>		

## C.2 Channel-specific machine data

<b>20050</b>	<b>AXCONF_GEOAX_ASSIGN_TAB</b>		
MD number	Assignment of geometry axis to channel axis		
Default setting: 1, 2, 3	Min. input limit: 0	Max. input limit: 20	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: BYTE			
Meaning:	<p>In this MD, the channel axis is specified to which the geometry axis is assigned. The assignment is to be done channel-specific for all geometry axes. If no assignment is done for a geometry axis, then this geometry axis is not available and cannot be programmed (with the name defined under AXCONF_GEOAX_NAME_TAB).</p> <p>E.g.: Turning machine without transformation:            \$MC_AXCONF_GEOAX_ASSIGN_TAB[ 0 ] = 1 ; 1st geometry axis = 1st channel axis            \$MC_AXCONF_GEOAX_ASSIGN_TAB[ 1 ] = 0 ; 2nd geometry axis not defined            \$MC_AXCONF_GEOAX_ASSIGN_TAB[ 2 ] = 2 ; 3rd geometry axis = 2nd channel axis</p> <p>The assignment made here is applicable if no transformation is active. For active transformation n the transformation-specific assignment table TRAFO_GEOAX_ASSIGN_TAB_n becomes effective.</p>		

<b>20060</b>	<b>AXCONF_GEOAX_NAME_TAB</b>		
MD number	Geometry axis name in the channel		
Default setting: X, Y, Z	Min. input limit: -	Max. input limit: -	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: STRING			
Meaning:	<p>In this MD, the names of the geometry axes are entered separately for the channel. With the names specified here, the geometry axes can be programmed in the part program.</p>		

<b>20070</b>	<b>AXCONF_MACHAX_USED</b>		
MD number	Machine axis number valid in the channel		
Default setting: 1, 2, 3, 4	Min. input limit: 0	Max. input limit: 31	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: BYTE			
Meaning:	<p>In this MD, the machine axis is specified to which the channel/special axis is assigned. The assignment is to be done channel-specific for all channel axes. A machine axis not assigned to any channel is not active, i.e. the axis control is not processed, the axis is not shown on the screen and is not programmed in any channel.</p>		

20080		AXCONF_CHANAX_NAME_TAB	
MD number	Name of channel axis in the channel		
Default setting: X, Y, Z, A, B, C, U, V, X11, Y11, ....	Min. input limit: -	Max. input limit: -	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: STRING			
Meaning:	The name of the channel/special axis is entered in this MD. In the normal case the first three channel axes are filled by the three assigned geometry axes (see also MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB). The remaining channel axes are also known as special axes. The display of the channel axis / special axis on the screen in the WCS (Workpiece Coordinate System) is always done with the names entered in this MD.		

20150		GCODE_RESET_VALUES	
MD number	Reset position of G groups		
Default setting: 2, 0, 0, 1, 0, ...	Min. input limit: -	Max. input limit: -	
Changes effective after RESET	Protection level: 2/7	Unit: -	
Data type: BYTE			
Meaning:	<p>Determination of the G codes which become effective during run-up and reset or part program end and part program start.</p> <p>The G code must be specified in the respective groups as preset value.</p> <p>Naming - Group - Default value:</p> <p>GCODE_RESET_VALUES[0] - Group 1 - Default value 2 (G01)</p> <p>GCODE_RESET_VALUES[1] - Group 2 - Default value 0 (inactive)</p> <p>GCODE_RESET_VALUES[2] - Group 3 - Default value 0 (inactive)</p> <p>GCODE_RESET_VALUES[3] - Group 4 - Default value 1 (START FIFO)</p> <p>GCODE_RESET_VALUES[4] - Group 5 - Default value 0 (inactive)</p> <p>GCODE_RESET_VALUES[5] - Group 6 - Default value 1 (G17) in case of turning</p> <p>GCODE_RESET_VALUES[6] - Group 7 - Default value 1 (G40)</p> <p>GCODE_RESET_VALUES[7] - Group 8 - Default value 1 (G500)</p> <p>GCODE_RESET_VALUES[8] - Group 9 - Default value 0 (inactive)</p> <p>GCODE_RESET_VALUES[9] - Group 10 - Default value 1 (G60)</p> <p>GCODE_RESET_VALUES[10] - Group 11 - Default value 0 (inactive)</p> <p>GCODE_RESET_VALUES[11] - Group 12 - Default value 1 (G601)</p> <p>GCODE_RESET_VALUES[12] - Group 13 - Default value 2 (G71)</p> <p>GCODE_RESET_VALUES[13] - Group 14 - Default value 1 (G90)</p> <p>GCODE_RESET_VALUES[14] - Group 15 - Default value 2 (G94)</p> <p>GCODE_RESET_VALUES[15] - Group 16 - Default value 1 (CFC)</p> <p>...</p>		

<b>20154</b>	<b>EXTERN_GCODE_RESET_VALUES[n]: 0, ..., 30</b>		
MD number	Defining the G code to become effective during power-up, when the NC channel does not run in the Siemens mode.		
Default setting: -	Min. input limit: -	Max. input limit: -	
Changes effective after POWER ON	Protection level: 2/2	Unit: -	
Data type: BYTE			
Meaning:	<p>The following external programming languages are possible:</p> <ul style="list-style-type: none"> <li>• ISO dialect milling</li> <li>• ISO dialect turning</li> </ul> <p>The G group classification to be used results from the current SINUMERIK documentation. The following groups within the MD EXTERN_GCODE_RESET_VALUES can be written:</p> <p><b>ISO Dialect M:</b>  G group 2: G17/G18/G19  G group 3: G90/G91  G group 5: G94/G95  G group 6: G20/G21  G group 13: G96/G97  G group 14: G54-G59</p> <p><b>ISO Dialect T:</b>  G group 2: G96/G97  G group 3: G90/G91  G group 5: G94/G95  G group 6: G20/G21  G group 16: G17/G18/G19</p>		
MD cannot be changed for SINUMERIK 802D sl.			

20380		TOOL_CORR_MODE_G43/G44	
MD number	Handling the tool length offset G43/G44		
Default setting: 0	Min. input limit: 1	Max. input limit: 2	
Changes effective after RESET	Protection level: 2/7		Unit: -
Data type: BYTE			
Meaning:	<p>The machine data is effective only for \$MN_MM_EXTERN_CNC_LANGUAGE = 1;            If G43/G44 is active, it determines how length compensations programmed with H are processed.</p> <p>0: Mode A            The tool length H always acts on the Z-axis, regardless of the current plane</p> <p>1: Mode B            The tool length H acts, regardless of the current plane, on one of the three geometry axes and on the 3rd geometry axis (as a rule Z) in case of G17            G18 on the 2nd geometry axis (as a rule Y)            G19 on the 1st geometry axis (as a rule X)</p> <p>Offsets can be built up in this mode in all three geometry axes through multiple programming i.e. by activating a component the length offset possibly active in another axis is not deleted.</p> <p>2: Mode C            Regardless of the active plane, the tool length acts in the axis which has been programmed simultaneously with H. The behavior is the same as in the case of variant B</p>		
MD cannot be changed for SINUMERIK 802D sl.			

20382		TOOL_CORR_MOVE_MODE	
MD number	Retracting the tool length offset		
Default setting: FALSE	Min. input limit: -	Max. input limit: -	
Changes effective after RESET	Protection level: 2/7		Unit: -
Data type: BOOLEAN			
Meaning:	<p>The machine data defines how the tool length compensations are retracted.</p> <p>FALSE: A tool length component is retracted only when the related axis has been programmed (behavior as in the earlier software versions).</p> <p>TRUE: Tool lengths are always retracted immediately, regardless of whether the related axes are programmed or not.</p>		
MD cannot be changed for SINUMERIK 802D sl.			

<b>20732</b>	<b>EXTERN_G0_LINEAR_MODE</b>		
MD number	Interpolation behavior with G00		
Default setting: 1	Min. input limit: 0	Max. input limit: 1	
Changes effective after POWER ON	Protection level: 2/4	Unit: -	
Data type: BOOLEAN			
Meaning:	This MD defines the interpolation behavior in case of G00. 0: Axes are traversed as positioning axes 1: Interpolate axes with one another		

<b>20734</b>	<b>EXTERN_FUNCTION_MASK</b>		
MD number	Function mask for external language		
Default setting:	Min. input limit: 0	Max. input limit: 16	
Changes effective after RESET	Protection level: 2/7	Unit: -	
Data type: DWORD			
Meaning:	This machine data controls the functions in the ISO mode. Bit 0=0: ISO mode T: "A" and "C" are interpreted as axes. If control definition is programmed, then a comma must be present before "A" or "C". Bit 0=1: "A" and "C" are always interpreted as contour definition in the part program. There may be no axis A or C. Bit 1=0: ISO Mode T G10 P<100 tool geometry P>100 tool wear Bit 1=1: G10 P<10 000 tool geometry P>10 000 tool wear Bit 2=0: G04 Dwell time: always [s] or [ms] Bit 2=1: If G95 is active, dwell time is in spindle revolutions Bit 3=0: Errors in ISO Scanner lead to alarms Example: N5 G291 ; ISO Dialect Mode N10 WAIT ; Alarm 12080 "WAIT unknown" N15 G91 G500 ; Alarm 12080 "G500 unknown" Bit 3=1: Error in the ISO Scanner is not output, the block is forwarded to the Siemens translator Example: N5 G291 ; ISO Dialect Mode N10 WAIT ; Block is processed by the Siemens translator N15 G91 G500 ; Block is processed by the Siemens translator N20 X Y ; Block is processed by the ISO translator owing to G291, G91 from N15 is active		

<b>20734</b>	<b>EXTERN_FUNCTION_MASK</b>
Meaning:	<p>Bit 4=0: G00 is traversed in the exact stop function Example: In case of G64, G00 blocks with G64 are also traversed</p> <p>Bit 4=1: G00 blocks are always traversed with G09, even when G64 is active</p> <p>Bit 5=0: Movements of the rotary axis are executed on the shortest path</p> <p>Bit 5=1: Depending on the sign, the movements of the rotary axis are executed in positive or negative direction of rotation</p> <p>Bit 6=0: Only four-digit program number allowed</p> <p>Bit 6=1: Eight-digit program number allowed. If number of digits less than 4, then extended by 4 digits.</p> <p>Bit 7=0: Axis programming in case of geo-axis interchange / parallel axes is compatible in ISO mode</p> <p>Bit 7=1: Axis programming in case of geo-axis interchange / parallel axes is compatible in ISO mode to the Siemens mode</p> <p>Bit 8=0: In case of cycles, the F value is always interpreted as feedrate</p> <p>Bit 8=1: In case of threading cycles, the F value is always interpreted as lead</p> <p>bit9=0: For ISO Mode T in case of G84, G88 in the standard mode F for G95 is multiplied by 0.01 mm or 0.0001 inch</p> <p>Bit 9=1: For ISO Mode T in case of G84, G88 in the standard mode F for G95 is multiplied by 0.01 mm or 0.0001 inch</p> <p>Bit 10=0: For M96 Pxx in case of interrupt is always with Pxx progr. program is called</p> <p>Bit 10=1: For M96 Pxx for the interrupt always CYCLE396.spf is called</p> <p>Bit 11=0: In the programming of G54 Pxx, G54.1 is shown</p> <p>Bit 11=1: In the programming of G54 Pxx or G54.1 Px, always G54Px is shown</p> <p>Bit 12=0: Upon the call of the subprogram defined with M96 Pxx, \$P_ISO_STACK is not changed</p> <p>Bit 12=1: Upon the call of the subprogram defined with M96 Pxx, \$P_ISO_STACK is incremented</p>

<b>22420</b>	<b>FGROUP_DEFAULT_AXES[n]: 0, ..., 7</b>		
MD number	Default value for FGROUP command		
Default setting: 0	Min. input limit: 0	Max. input limit: 8	
Changes effective after POWER ON	Protection level: 7/7	Unit: -	
Data type: BYTE			
Meaning:	<p>One can specify up to eight channel axes, whose resulting speed corresponds to the programmed path feedrate. If all the eight values are zero (predefined), then the geo-axes entered as default setting for the FGROUP command in \$MC_AXCONF_GEOAX_ASSIGN_TAB always becomes active.</p> <p>Example: The first four axes in the channel are relevant for the path feedrate:</p> <p>\$MC_FGROUP_DEFAULT_AXES[0] = 1  \$MC_FGROUP_DEFAULT_AXES[2] = 2  \$MC_FGROUP_DEFAULT_AXES[3] = 3  \$MC_FGROUP_DEFAULT_AXES[4] = 4</p>		
MD cannot be changed for SINUMERIK 802D sl.			

<b>22512</b>	<b>EXTERN_GCODE_GROUPS_TO_PLC[n]: 0, ..., 7</b>		
MD number	Specification of the G groups, which are printed at the NCK-PLC interface when an external NC language is active		
Default setting: -	Min. input limit: -	Max. input limit: -	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: BYTE			
Meaning:	<p>Via the channel machine data \$MC_EXTERN_GCODE_GROUPS_TO_PLC the user can select the G groups of an external NC language, whose active G command is to be reported from NCK to PLC.</p> <p>Default 0: no output</p> <p>The interface NCK_PLC is updated upon each block change and after reset. A block-synchronous relationship is not present in each case between NC block and reported G functions (e.g. in case of short blocks in the continuous-path mode).</p> <p>Analog \$MC_GCODE_GROUPS_TO_PLC</p>		
MD cannot be changed for SINUMERIK 802D sl.			

<b>22515</b>	<b>GCODE_GROUPS_TO_PLC_MODE</b>		
MD number	Behavior of the G group transfer to PLC		
Default setting: -	Min. input limit: -	Max. input limit: -	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: DWORD			
Meaning:	<p>For setting the behavior, how are the G groups to be interpreted in the PLC as data. In case of current behavior (bit 0=0) the G group is the array-index of an array having the size 64 bytes (DBB 208 - DBB 271). At the most, the 64th G group can therefore be reached.</p> <p>In case of a new behavior (Bit 0=1), the data storage in the PLC has the maximum size of eight bytes (DBB 208 - DBB 215). In this behavior, the array index of this byte array is identical to the index of MD \$MC_GCODE_GROUPS_TO_PLC[Index] and \$MC_EXTERN_GCODE_GROUPS_TO_PLC[Index]. Here each index (0-7) may be used only in one of the two machine data, for the other MD the value 0 must be entered.</p> <p>Bit 0 (LSB) = 0: Behavior as before, the 64-byte array is used to display the G code.</p> <p>Bit 0 (LSB) = 1: The user sets the G groups for which the first eight bytes are to be used.</p>		
MD cannot be changed for SINUMERIK 802D sl.			

<b>22900</b>	<b>STROKE_CHECK_INSIDE</b>		
MD number	Direction (inside/outside) in which the protection zone is effective		
Default setting: 0	Min. input limit: 0	Max. input limit: 1	
Changes effective after POWER ON	Protection level: 2/7		Unit: -
Data type: BYTE			
Meaning:	<p>This machine data is valid in connection with external programming languages. It is effective with \$MN_MM_EXTERN_LANGUAGE = 1.</p> <p>It is defined whether protection zone 3 is an inside protection zone or an outside protection zone.</p> <p>Meaning:</p> <p>0: Protection zone 3 is an inside protection zone, i.e. the protection zone inside may not be overrun</p> <p>1: Protection zone 3 is an outside protection zone</p>		
MD cannot be changed for SINUMERIK 802D sl.			

<b>22910</b>	<b>WEIGHTING_FACTOR_FOR_SCALE</b>		
MD number	Input resolution for scaling factor		
Default setting: 0	Min. input limit: 0	Max. input limit: 1	
Changes effective after POWER ON	Protection level: 2/7		Unit: -
Data type: BOOLEAN			
Meaning:	<p>This machine data is valid in connection with external programming languages. It is effective with \$MN_MM_EXTERN_LANGUAGE = 1.</p> <p>Determination of the unit for the scaling factor P and for the axial scaling factors I, J, K</p> <p>Meaning:</p> <p>0: Scaling factor in 0.001</p> <p>1: Scaling factor in 0.00001</p>		

<b>22914</b>	<b>AXES_SCALE_ENABLE</b>		
MD number	Activation for axial scaling factor (G51)		
Default setting: 0	Min. input limit: 0	Max. input limit: 1	
Changes effective after POWER ON	Protection level: 2/7		Unit: -
Data type: BOOLEAN			
Meaning:	<p>With this MD, the axial scaling is released.</p> <p>Meaning:</p> <p>0: Axial scaling is not possible</p> <p>1: Axial scaling is possible, i.e. MD DEFAULT_SCALE_FACTOR_AXIS is effective</p>		

<b>22920</b>	<b>EXTERN_FIXED_FEEDRATE_F1_ON</b>		
SD number	Activate fixed feedrates F1 - F9		
Default setting: FALSE	Min. input limit:	Max. input limit:	
Changes effective after POWER ON	Protection level: 2/7	Unit:	
Data type: BOOLEAN			
Meaning:	<p>This MD releases the fixed feedrates from the setting data \$SC_EXTERN_FIXED_FEEDRATE_F1_F9 [ ].</p> <p>0: No fixed feedrates with F1 - F9</p> <p>1: The feedrates from the setting data \$SC_EXTERN_FIXED_FEEDRATE_F1_F9 become effective with the programming of F1 - F9</p>		

<b>22930</b>	<b>EXTERN_PARALLEL_GEOAX</b>		
SD number	Assignment of parallel channel geometry axis		
Default setting: 0	Min. input limit: 0	Max. input limit: 3	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: BYTE			
Meaning:	<p>Assignment table of the axes which lie parallel to the geometry axes. Via this table the geometry axes can be assigned to the parallel channel axes. The parallel axes can then be activated in the ISO Dialect with the G functions of the plane selection (G17 - G19) and with the axes names of the parallel axis as geometry axis. An axis interchange is then performed with the axes defined via \$MC_AXCONF_GEOAX_ASSIGN_TAB[ ].</p> <p>Precondition: The channel axes used must be active (filled list place in AXCONF_MACHAX_USED).</p> <p>Entry of a zero deactivates the corresponding parallel geometry axis.</p>		

<b>24004</b>	<b>CHBFRAME_POWERON_MASK</b>		
MD number	Reset channel-specific basic frame after power-on		
Default setting: 0	Min. input limit: 0	Max. input limit: 0xFF	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: DWORD			
Meaning:	<p>This machine data specifies whether the channel-specific basic frames are reset upon a Power On Reset in the data storage, i.e. displacements and turnings are set to 0, scalings to 1. Mirroring is disabled. The individual basic frames can be selected separately.</p> <p>Bit 0 corresponds to basic frame 0, bit 1 to basic frame 1, etc.</p> <p>0: Basic frame is retained at Power On</p> <p>1: Basic frame is reset in the data management at Power ON.</p>		
MD cannot be changed for SINUMERIK 802D sl.			

### C.3 Axis-specific setting data

<b>43120</b>	<b>DEFAULT_SCALE_FACTOR_AXIS</b>		
MD number	Default axial scaling factor with active G51		
Default setting: 1	Min. input limit: -99999999	Max. input limit: 99999999	
Change valid IMMEDIATELY	Protection level: 7/7		Unit: -
Data type: DWORD			
Meaning:	<p>This machine data is valid in connection with external programming languages. It is effective with \$MN_MM_EXTERN_LANGUAGE = 1.</p> <p>If no axial scaling factor I, J or K is programmed in the G51 block, the DEFAULT_SCALEFAKTOR_AXIS is effective. The MD AXES_SCALE_ENABLE must be set so that the scaling factor is effective.</p>		

<b>43240</b>	<b>M19_SPOS</b>		
MD number	Spindle position in degrees for spindle positions with M19		
Default setting: 0	Min. input limit: -359.999	Max. input limit: 359.999	
Change valid IMMEDIATELY	Protection level: 7/7		Unit: -
Data type: DOUBLE			
Meaning:	The setting data is also effective in Siemens mode.		

## C.4 Channel-specific setting data

42110		DEFAULT_FEED	
SD number	Default value for path feed		
Default setting: 0	Min. input limit: 0	Max. input limit: -	
Change valid IMMEDIATELY	Protection level: 7/7		Unit: -
Data type: DOUBLE			
Meaning:	<p>If no path feed is programmed in the part program, then the value stored in \$SC_DEFAULT_FEED is used.</p> <p>The evaluation of the setting data takes place at the start of the part program taking into consideration the feed type active at this time (see \$MC_GCODE_RESET_VALUES or \$MC_EXTERN_GCODE_RESET_VALUES).</p>		

42140		DEFAULT_SCALE_FACTOR_P	
SD number	Default scaling factor for address P		
Default setting: 0	Min. input limit: -99999999	Max. input limit: 99999999	
Change valid IMMEDIATELY	Protection level: 7/7		Unit: -
Data type: DWORD			
Meaning:	<p>This machine data is valid in connection with external programming languages. It is effective with \$MN_MM_EXTERN_LANGUAGE = 1.</p> <p>If no scaling factor P is programmed in a block, the value from this machine data becomes effective.</p>		

42150		DEFAULT_ROT_FACTOR_R	
SD number	Default angle of rotation R		
Default setting: 0	Min. input limit: 0	Max. input limit: 360	
Change valid IMMEDIATELY	Protection level: 2/7		Unit: degree
Data type: DOUBLE			
Meaning:	<p>If no factor for rotation R is programmed with the selection of the rotation G68, the value from this setting data becomes effective.</p>		

<b>42160</b>	<b>EXTERN_FIXED_FEEDRATE_F1_F9</b>		
SD number	Fixed feeds with F1 - F9		
Default setting: 0	Min. input limit: 0	Max. input limit:	
Change valid IMMEDIATELY	Protection level: 2/7	Unit: VELO	
Data type: DOUBLE			
Meaning:	Fixed feed values for the programming of F1 - F9. If the machine data \$MC_FEEDRATE_F1_F9_ON=TRUE is set, then with the programming of F1 - F9, the feed values are read from the setting data \$SC_EXTERN_FIXED_FEEDRATE_F1_F9[0] - \$SC_EXTERN_FIXED_FEEDRATE_F1_F9[8] and activated as machining feed. In \$SC_EXTERN_FIXED_FEEDRATE_F1_F9[0] the rapid traverse feed must be entered.		

<b>42162</b>	<b>EXTERN_DOUBLE_TURRET_DIST</b>		
SD number	Tool distance of double turret head		
Default setting:	Min. input limit:	Max. input limit:	
Change effective	Protection level:	Unit:	
Data type: DOUBLE			
Meaning:	The machine data is effective only in case of \$MN_EXTER_CNC_SYSTEM = 2. Turning distance of both tools of a double slide turret. The distance is activated with G68 as additive work offset if \$MN_EXTERN_DOUBLE_TURRET_ON = TRUE is set.		



## Data lists

### D.1 Machine data

Number	Identifier	Name
<b>General (\$MN_ ... )</b>		
10604	WALIM_GEOAX_CHANGE_MODE	Working area limitation during switchover of geometry axes
10615	NCFRAME_POWERON_MASK	Delete global basic frames after Power On
10652	CONTOUR_DEF_ANGLE_NAME	Settable name for the angle in the contour brief description
10654	RADIUS_NAME	Settable name for the radius non-modal in the contour brief description
10656	CHAMFER_NAME	Settable name for the chamfer in the contour brief description
10704	DRYRUN_MASK	Activation of dry run feedrate
10706	SLASH_MASK	Activation of block skip function
10715	M_NO_FCT_CYCLE[n]: 0, ..., 0	M function number for tool-changing cycle call
10716	M_NO_FCT_CYCLE_NAME[ ]	Name of tool-changing cycle with M functions from MD \$MN_MFCT_CYCLE
10717	T_NO_FCT_CYCLE_NAME	Name of tool-changing cycle for T function
10760	G53_TOOLCORR	Mode of operation G53, G153 and SUPA
10800	EXTERN_CHAN_SYNC_M_NO_MIN	First M number for channel synchronization
10802	EXTERN_CHAN_SYNC_M_NO_MAX	Last M number for channel synchronization
10804	EXTERN_M_NO_SET_INT	M function for ASUB activation
10806	EXTERN_M_NO_DISABLE_INT	M function for ASUB deactivation
10808	EXTERN_INTERRUPT_BITS_M96	Interrupt program execution (M96)
10810	EXTERN_MEAS_G31_P_SIGNAL	Assignment of the measuring inputs for G31 P..
10812	EXTERN_DOUBLE_TURRET_ON	Double turret head with G68
10814	EXTERN_M_NO_MAC_CYCLE	Macro call via M function
10815	EXTERN_M_NO_MAC_CYCLE_NAME	Subprogram name for M function macro call
10816	EXTERN_G_NO_MAC_CYCLE	Macro call via G function
10817	EXTERN_G_NO_MAC_CYCLE_NAME	Subprogram name for G function macro call
10818	EXTERN_INTERRUPT_NUM_ASUP	Interrupt number for ASUB start (M96)
10820	EXTERN_INTERRUPT_NUM_RETRAC	Interrupt number for rapid retraction (G10.6)
10880	EXTERN_CNC_SYSTEM	External control system, whose programs are to be executed
10881	EXTERN_GCODE_SYSTEM	ISO mode T: G Code System
10882	NC_USER_EXTERN_GCODES_TAB[n]: 0-59	List of user-specific G commands of an external NC language
10884	EXTERN_FLOATINGPOINT_PROG	Evaluation of programmed values without decimal point

Number	Identifier	Name
10886	EXTERN_INCREMENT_SYSTEM	Defining the increment systems
10888	EXTERN_DIGITS_TOOL_NO	Number of digits for T number in external language mode
10890	EXTERN_TOOLPROG_MODE	Tool change programming in case of external programming language
18190	MM_NUM_PROTECT_AREA_NCK	Number of files for machine related protection zones (SRAM)
18800	MM_EXTERN_LANGUAGE	Activation of external NC languages
<b>Channel-specific (\$MC_ ... )</b>		
20050	AXCONF_GEOAX_ASSIGN_TAB[ ]	Assignment of geometry axis to channel axis
20060	AXCONF_GEOAX_NAME_TAB[ ]	Geometry axis in channel
20070	AXCONF_MACHAX_USED[ ]	Machine axis number valid in the channel
20080	AXCONF_CHANAX_NAME_TAB[ ]	Name of channel axis in the channel
20094	SPIND_RIGID_TAPPING_M_NR	M function for switching over to controlled axis mode
20095	EXTERN_RIGID_TAPPING_M_NR	M function number in the external language mode for switching over of spindle in the controlled spindle mode
20100	DIAMETER_AX_DEF	Geometry axis with cross axis functionality
20150	GCODE_RESET_VALUES[n]: 0 to max. number of G codes	Reset position of G groups
20154	EXTERN_GCODE_RESET_VALUES[n]: 0-30	Defining the G code to become effective during power on, when the NC channel does not run in the Siemens mode
20380	TOOL_CORR_MODE_G43G44	Handling the tool length offset G43/G44
20382	TOOL_CORR_MOVE_MODE	Retracting the tool length offset
20732	EXTERN_G0_LINEAR_MODE	Determine interpolation behavior with G00
20734	EXTERN_FUNCTION_MASK	Function mask for external language
22420	FGROUP_DEFAULT_AXES[ ]	Default value for FGROUP command
22512	EXTERN_GCODE_GROUPS_TO_PLC[n]: 0-7	Send G command to an external NC language at PLC
22515	GCODE_GROUPS_TO_PLC_MODE	Behavior of the G group transfer to PLC
22900	STROKE_CHECK_INSIDE	Direction (inside/outside) in which the protection zone is effective
22910	WEIGHTING_FACTOR_FOR_SCALE	Unit of the scaling factors
22914	AXES_SCALE_ENABLE	Activation for axial scaling factor (G51)
22920	EXTERN_FEEDRATE_F1_F9_ACTIV	Permit fixed feeds with F0 - F9
22930	EXTERN_PARALLEL_GEOAX	Assignment of parallel channel geometry axis
24004	CHBFRAME_POWERON_MASK	Reset channel-specific basic frame after power-on
28080	NUM_USER_FRAMES	Number of work offsets
29210	NUM_PROTECT_AREA_ACTIVE	Activate protection zone
34100	REFP_SET_POS[0]	Reference point value / with distance-coded system without any significance
35000	SPIND_ASSIGN_TO_MACHAX	Assign spindle/machine axis

## Channel-specific cycle machine data

Number	Identifier	Name
52802	ISO_ENABLE_INTERRUPTS	Interrupt processing
52804	ISO_ENABLE_DRYRUN	Processing skip during DRYRUN
52806	ISO_SCALING_SYSTEM	Basic system
52808	ISO_SIMULTAN_AXES_START	Simultaneous drill position approach for all programmed axes
52810	ISO_T_DEEPHOLE_DRILL_MODE	Deep-hole drilling with chip breaking / removal of chips

## D.2 Setting data

Table D- 1

Number	Identifier	Name
<b>Axis-specific</b>		
43120	DEFAULT_SCALE_FACTOR_AXIS	Default axial scaling factor with active G51
43240	M19_SPOS	Position of the spindle on programming M19
42890	M19_SPOSMODE	Positioning mode of the spindle on programming M19
<b>Channel-specific</b>		
42110	DEFAULT_FEED	Default value for path feed
42140	DEFAULT_SCALE_FACTOR_P	Default scaling factor for address P
42150	DEFAULT_ROT_FACTOR_R	Default angle of rotation R
42160	EXTERN_FIXED_FEEDRATE_F1_F9	Fixed feeds with F1-F9
42162	EXTERN_DOUBLE_TURRET_DIST	Tool distance of double turret head

### Channel-specific cycle setting data

Number	Identifier	Name
55808	ISO_T_RETRACTION_FACTOR	Factor for retraction speed (0...200%)
55810	ISO_T_DWELL_TIME_UNIT	Evaluation of the dwell time

## D.3 Variables

Identifier	Type	Description
\$C_A	REAL	Value of the programmed address A in the ISO Dialect mode for cycle programming
\$C_B	REAL	Value of the programmed address B in the ISO Dialect mode for cycle programming
....	....	.....
\$C_G	INT	G number for cycle calls in the external mode
\$C_H	REAL	Value of the programmed address H in the ISO Dialect mode for cycle programming
\$C_I[ ]	REAL	Value of the programmed address I in the ISO Dialect mode for cycle programming and macro technology with G65/G66. Maximum 10 entries are possible in the block for the macro programming. The values are present in an array in the programmed sequence.
\$C_I_ORDER[ ]	REAL	For description see \$C_I[ ], used to determine the programming sequence
\$C_J[ ]	REAL	For description see \$C_I[ ]
\$C_J_ORDER[ ]	REAL	For description see \$C_I[ ], used to determine the programming sequence
\$C_K[ ]	REAL	For description see \$C_I[ ]
\$C_K_ORDER[ ]	REAL	For description see \$C_I[ ], used to determine the programming sequence
\$C_L	INT	Value of the programmed address L in the ISO Dialect mode for cycle programming
....	....	....
\$C_Z	INT	Value of the programmed address Z in the ISO Dialect mode for cycle programming
\$C_TS	STRING	String of tool identifier programmed under the address T
\$C_A_PROG	INT	Address A is programmed in a block with cycle call 0 = not programmed 1 = programmed (absolute) 3 = programmed (incremental)
\$C_B_PROG	INT	Address B is programmed in a block with cycle call 0 = not programmed 1 = programmed (absolute) 3 = programmed (incremental)
....	....	....
\$C_G_PROG	INT	Shell cycle is programmed via a G function
\$C_Z_PROG	INT	Address Z is programmed in a block with cycle call 0 = not programmed 1 = programmed (absolute) 3 = programmed (incremental)
\$C_TS_PROG	INT	A tool identifier has been programmed under the address T TRUE = programmed, FALSE = not programmed

Identifier	Type	Description
\$C_ALL_PROG	INT	Bit pattern of all programmed addresses in a block with cycle call Bit 0 = address A Bit 25 = address Z Bit = 1 address programmed Bit = 0 address not programmed
\$P_EXTGG[n]	INT	Active G code of the external language
\$C_INC_PROG	INT	Bit pattern of all incremental programmed addresses in a block with cycle call Bit 0 = address A Bit 25 = address Z Bit = 1 address incremental programmed Bit = 0 address absolute programmed
\$C_I_NUM	INT	Cycle programming: Value is always 1, if the bit 0 is set in \$C_I_PROG. Macro programming: Number of I addresses programmed in the block (max. 10).
\$C_J_NUM	INT	For description see \$C_I_NUM
\$C_K_NUM	INT	For description see \$C_I_NUM
\$P_AP	INT	Polar coordinates 0 = Off 1 = On
\$C_TYP_PROG	INT	Bit pattern of all programmed addresses in a block with cycle call Bit 0 = A Bit 25 = Z Bit = 0 axis is programmed as INT Bit = 1 axis is programmed as REAL
\$C_PI	INT	Program number of the interrupt routine programmed with M96

# Alarms

## E.1 Alarms

When error states are detected in the cycle, an alarm is generated and the cycle execution is interrupted.

The cycles output the messages in the message line of the control system. These messages do not interrupt the machining.

The alarms with the numbers from 61000 to 62999 are generated in the cycles. This large range is further subdivided according to alarm responses and cancel criteria.

Table E- 1 Alarm number and alarm description

Alarm No.	Product brief	Cause	Explanation/remedy
<b>General alarms</b>			
61001	Incorrect thread lead	CYCLE376T	The thread lead is not specified correctly
61003	No feed programmed in the cycle	CYCLE371T, CYCLE374T, CYCLE383T, CYCLE384T, CYCLE385T, CYCLE381M, CYCLE383M, CYCLE384M, CYCLE387M	No F word programmed in the calling block before the cycle call, see Siemens standard cycles.
61004	Configuration of geometry axis not correct	CYCLE328	The order of the geometry axes is false, see Siemens standard cycles
61101	Reference plane defined incorrectly	CYCLE375T, CYCLE81, CYCLE83, CYCLE84, CYCLE87	See Siemens standard cycles
61102	No spindle direction programmed	CYCLE371T, CYCLE374T, CYCLE383T, CYCLE384T, CYCLE385T, CYCLE381M, CYCLE383M, CYCLE384M, CYCLE387M	Spindle direction M03 or M04 missing; see Siemens standard cycles
61107	First drilling depth defined incorrectly		First drilling depth is opposite to total drilling depth
61603	Groove shape defined incorrectly	CYCLE374T	Value for groove depth equal to 0
61607	Start point programmed incorrectly	CYCLE376T	The start point lies outside the machining area.
61610	No infeed depth programmed	CYCLE374T	Infeed value = 0

## Alarms

### E.1 Alarms

Alarm No.	Product brief	Cause	Explanation/remedy
<b>ISO alarms</b>			
61800	External CNC system missing	CYCLE300, CYCLE328, CYCLE330, CYCLE371T, CYCLE374T, CYCLE376T, CYCLE383T, CYCLE384T, CYCLE385T, CYCLE381M, CYCLE383M, CYCLE384M, CYCLE387M	Machine data for external language MD18800 \$MN_MM_EXTERN_LANGUAGE or options bit 19800 \$MN_EXTERN_LANGUAGE is not set.
61801	Incorrect G code selected	CYCLE300, CYCLE371T, CYCLE374T, CYCLE376T, CYCLE383T, CYCLE384T, CYCLE385T	In the program call CYCLE300<Value> an impermissible value has been programmed or an incorrect value has been specified in the cycle setting data for the G code system.
61802	Incorrect axis type	CYCLE328, CYCLE330	The programmed axis is assigned to a spindle.
61803	Programmed axis not available	CYCLE328, CYCLE330	The programmed axis is not available at the control system. Check MD20050-20080.
61804	Programmed position exceeds the reference point	CYCLE328, CYCLE330	The programmed intermediate position or the current position lies behind the reference point.
61805	Value programmed absolute and incremental	CYCLE328, CYCLE330, CYCLE371T, CYCLE374T, CYCLE376T, CYCLE383T, CYCLE384T, CYCLE385T	The intermediate position is programmed with absolute as well as incremental values.
61806	Incorrect axis assignment	CYCLE328	The sequence of the axes is incorrect.
61807	Incorrect spindle direction programmed	CYCLE384M	The programmed spindle direction contradicts the spindle direction intended for the cycle.
61808	Final drilling depth or single drilling depth missing	CYCLE383T, CYCLE384T, CYCLE385T, CYCLE381M, CYCLE383M, CYCLE384M, CYCLE387M	Total drilling depth Z or single drilling depth Q missing in the G8x-block (first call of cycle)
61809	Drilling position not permitted	CYCLE383T, CYCLE384T, CYCLE385T	
61810	ISO G Code not possible	CYCLE383T, CYCLE384T, CYCLE385T	
61811	ISO axis identifier not permitted	CYCLE328, CYCLE330, CYCLE371T, CYCLE374T, CYCLE376T, CYCLE383T, CYCLE384T, CYCLE385T	The calling NC block contains an impermissible ISO axis identifier.

Alarm No.	Product brief	Cause	Explanation/remedy
61812	Value(s) in the external cycle call defined incorrectly	CYCLE371T, CYCLE376T,	The calling NC block contains an impermissible numeric value.
61813	GUD value defined incorrectly	CYCLE376T	An incorrect numeric value has been entered in the cycle setting data.
61814	Polar coordinates not possible with cycle	CYCLE381M, CYCLE383M, CYCLE384M, CYCLE387M	
61815	G40 not active	CYCLE374T, CYCLE376T	G40 was not active before cycle call.



# Glossary

## A spline

The Akima spline runs tangentially through the programmed interpolation points (3rd order polynomial).

## Absolute dimensions

A specification for the target of axis motion is defined by a dimension that refers to the origin of the currently active coordinate system. See also -> incremental dimension.

## AC (adaptive control)

A process variable (e.g. path- or axis-specific feedrate) can be controlled as a function of another measured process variable (e.g. depending on spindle current). Typical application: Maintenance of constant chip removal volume during grinding.

## Acceleration and jerk limit

To achieve an optimal acceleration factor for the machine while preserving the mechanical parts, the machining program offers the option of switching between instantaneous (inertia-free) and constant (jerk-free) acceleration.

## Access rights

The blocks of a CNC program are protected via the following 7-level system of access-authorization restrictions:

- Three password levels, one each for the system manufacturer, the machine manufacturer and the user;
- Four key-operated switch settings which can be evaluated via the PLC.

## Activation / deactivation

Working area limitation is a means of limiting the axis motion over the restrictions set by the limit switch. A pair of values can be specified for each axis with which the protection zone is delimited.

## Address

Addresses are fixed or variable identifiers for axes (X, Y, ...), spindle speed (S), feedrate (F), circle radius (CR), etc.

## Analog input and output modules

Analog input and output modules are sensors for analog process signals.

Analog input modules are used to convert measured analog values into digital values, so that they can be processed in the CPU. Analog input modules convert digital values into manipulated variables.

## Archiving

Exporting files or directories to an external memory medium.

## Asynchronous subprogram

- A part program that can be started with an interrupt signal (e.g. "Rapid NC input signal") asynchronously (i.e. independently) while a part program is active.
- A part program that can be started with an interrupt signal (e.g. "Rapid NC input signal") asynchronously (i.e. independent of the current program status).

## AUTOMATIC or Automatic mode

Operating mode of the control system (block sequence according to DIN): Operating mode of NC controls in which a -> part program is selected and continuously executed.

## Auxiliary functions

Auxiliary functions can be used to transfer -> parameters in part programs to the -> PLC, whereupon reactions defined by the machine manufacturer are triggered.

## Axes

CNC axes are set according to their functionality as follows:

- Axes: Interpolatory path axes
- Positioning axes: Non-interpolating infeed and positioning axes with axis-specific feedrates; these axes can be traversed beyond the block limits. Positioning axes need not be involved in the machining of the tool and they contain, e.g. tool feeder, tool magazine etc.

**Axis identifier**

According to DIN 66217, axes are denoted using X, Y, and Z for a clockwise, rectangular -> coordinate system.

-> Rotary axes that rotate around X, Y and Z, to which the identifiers A, B and C are assigned. Additional axes parallel to the mentioned ones can be denoted with other letters.

**Axis/spindle replacement**

An axis/spindle is permanently assigned to a specific channel via machine data. This assignment via machine data can be reset with program commands and the axis/spindle can be assigned to another channel.

**B spline**

The points programmed for the B spline are not interpolation points but simply "checkpoints". The generated curve only runs near the checkpoints, not directly through them (optional 1st, 2nd or 3rd order polynomials).

**Backlash compensation**

Compensation of the mechanical backlash of the machine, e.g. the backlash compensation of the spindles. The backlash compensation can be input separately for each axis.

**Backup**

A copy of the memory contents (hard disk) stored on an external device for the purpose of data backup/archiving.

**Base axis**

Axis whose setpoint value or actual value is used in the calculation of the offset value.

**Basic coordinate system**

Cartesian coordinate system which is mapped through a transformation onto the machine coordinate system.

The programmer works in the -> part program with the axis names of the basic coordinate systems. The basic coordinate system exists in parallel to the -> machine coordinate system if no -> transformation is active. The difference between the two systems lies only in the axis identifiers.

**Baud rate**

The speed at which the data transfer takes place (bits/s).

**Block search**

The block search function can be used to go to any convenient point in a part program at which the machining is to start or continue. This function is intended for testing the part programs or for continuing the machining after an interruption.

**Booting**

Loading the system program after power on.

**C axis**

An axis which the tool uses to describe a controlled rotary or positioning movement.

**C spline**

The C spline is the most well-known and widely used spline. The spline runs along a tangent and along the axis of curvature through all interpolation points. 3rd order polynomials are used for this.

**Channel structure**

The -> programs of the individual channels can be processed simultaneously or asynchronously with the channel structure.

**Circular interpolation**

In circular interpolation, the -> tool traverses between defined contour points at a definite feedrate during the machining of the workpiece on a circular path.

**Clearance control (3D), sensor-driven**

The positional shift for a certain axis can be controlled as a function of a measured process variable (e.g. analog input, spindle current...). This function can be used to maintain a fixed distance automatically so as to fulfill certain technological requirements of the relevant machining.

**CNC programming language**

The CNC programming language is based on DIN 66025 with high-level language extensions. The CNC programming language and the high-level language extensions support the definition of macros (sequenced statements).

**COM**

Part of the numerical control system for the implementation and coordination of communication.

**Command axis**

Command axes are started from synchronized actions as a reaction to an event (command). Command axes can be positioned, started and stopped fully asynchronous to the part program.

**Compensation axis**

Axis with a setpoint or actual value modified by a compensation value.

**Compensation memory**

Data storage area in the control system, in which the tool offset data is stored.

**Compensation table**

Table of interpolation points. It delivers the offset values of the offset axis for the selected positions of the basic axis.

**Connecting cables**

Connecting cables are either pre-assembled cables or cables that are pre-assembled, ready-to-use two-wire cables that have a connector at both ends. Connecting cables are used to connect the -> CPU to a -> programming device or another CPU with use of an -> "interface".

**Continuous-path mode**

The purpose of the continuous-path mode is to avoid an excessive acceleration of the -> path axes at the block limits of part programs, which may affect the operator, the machine or other assets of the plant. The continuous-path mode is expected to influence the transition to the next block in the NC program and to maintain the path velocity as uniformly as possible.

## Contour

Contours of a workpiece.

## Contour monitoring

The following error is monitored as a measure for contour precision within a defined tolerance range. Thus, for instance, an overloading of the drive can lead to a sequential fault, which is no longer acceptable. An alarm is output and the axes are stopped in this case.

## Correction value

The distance between the axis position and the desired programmed axis position measured with a position encoder.

## CPU

Central Processor Unit -> programmable logic controller

## Cycle

Protected subprogram for executing machining operations which are always repeated at the -> workpiece.

## Cycle support

The available cycles are listed in the "Cycle support" menu in the "Program" operating area. The required parameters for the assignment of the values are displayed in plain text after the desired execution cycle has been selected.

## Data block

- The unit of data used in the -> PLC which can be accessed via -> HIGHSTEP program.
- Unit of data in the -> NC: Data blocks that contain data definitions for global user data. This data can be initialized directly during its definition.

## Data word

Data unit inside a -> PLC data block having a size of 2 bytes

### **Deletion of distance-to-go**

Command in a part program with which the machining is stopped and the distance-to-go is deleted.

### **Diagnostics**

- Operating area of the control.
- The control system contains a self-diagnostics program and test routines for service: Status, alarm and service displays.

### **DRF**

Differential Resolver Function. This is a function of an NC in which an incremental work offset is created in the automatic mode in conjunction with the electronic handwheel.

### **Dynamic feedforward control**

The "Dynamic acceleration-dependent feedforward control" function can be used to fully overcome the contour inaccuracies that result from secondary faults. The feedforward control function enables an extremely high degree of machining accuracy at high tool path velocities. The feedforward control function allows selection or deselection of all axes with just one part program.

### **Editor**

The editor is used to create, change, add, connect and insert programs/texts/blocks of a program.

### **Electronic handwheel**

The selected axes can be operated simultaneously in manual mode with an electronic handwheel. The handwheel clicks are evaluated by the increment analyzer.

### **Exact stop**

If exact stop is programmed, the position specified in the block is approached accurately and - if necessary - very slowly. To reduce the startup speeds, -> exact stop limits are defined for rapid traverse and feedrate.

### **Exact stop limit**

After all the path axes reach their exact stop limits, the control system reacts as if they have reached the end point exactly. The -> part program continues with the machining from the next block.

### **External work offset**

Work offset specified by the -> PLC.

### **Fast retraction from contour**

When an interrupt occurs, a motion can be initiated via the CNC machining program, enabling the tool to be quickly retracted from the workpiece contour that is currently being machined. The retraction angle and the retraction path can also be parameterized. An interrupt routine can be carried out after a rapid retraction.

### **Feedrate override**

In feedrate override, the current feedrate entered in the operator panel or specified by the PLC is overlaid on the programmed feedrate (0 - 200%). A feedrate override is also possible by means of a programmable percentage value (1 - 200%) in the machining program. A feedrate correction can also be applied by synchronous actions regardless of the program currently running.

### **Fixed machine point**

Point that is defined uniquely by the machine tool, e.g. the reference point.

### **Fixed-point approach**

The machine tool can be used to approach defined fixed points such as tool change points, loading points, palette change points etc. The coordinates of these points are stored in the control system. If possible, the control system approaches these axes at -> rapid traverse.

### **Frame**

A frame is an arithmetic rule, with which one Cartesian coordinate system is converted to another Cartesian coordinate system. A frame consists of the components -> work offset -> rotation -> scaling and -> mirroring.

## **Geometry**

Description of a -> workpiece in the -> workpiece coordinate system.

## **Geometry axis**

Geometry axes are used to describe a 2- or 3-dimensional range in the workpiece coordinate system.

## **Global main program / subprogram**

Each global main program / subprogram can be stored only once under its name in the directory. However, the same name can be used several times in one and the same directory.

## **Ground**

The term "Ground" is used for all electrically inactive, interconnected parts of one part of the system or a resource which cannot result in dangerous touch voltage in the event of errors.

## **Helical interpolation**

The "Helical interpolation" function is especially suitable for machining female or male threads with form cutters and for milling lubricating grooves. The helix comprises two movements:

Circular movement in the plane  
Linear movement vertical to this plane

## **High-speed digital inputs/outputs**

Example: High-speed CNC program routines (interrupt routines) that can be started via digital inputs. Digital CNC outputs (SINUMERIK 840D sl) can be used to trigger switching functions operated by high-speed programs.

## **HIGHSTEP**

Combination of different programming features for the -> PLC in the range S7-300/400.

## **I/O module**

I/O modules are used to establish a connection between the CPU and the process.

I/O modules are:

Digital input and output modules  
Analog input and output modules  
Simulator modules

## **Identifiers**

According to DIN 66025, identifiers (names) of variables (arithmetic variables, system variables, user variables), of subprograms, of vocabulary words and of words can contain several address characters. These characters have the same meaning as the words in the block syntax. Identifiers must always be unique. Different identifiers should always be used for different objects.

## **Inch measuring system**

Measuring system with which the distance traversed and fractions thereof are specified in inches.

## **Inclined axis**

Fixed angular interpolation with allowance for an inclined infeed axis or grinding wheel by specifying the angle. The inclined axes are programmed and displayed in the Cartesian coordinate system.

## **Inclined surface machining**

The "Inclined surface machining" function supports drilling and milling operations on workpiece surfaces that are inclined with reference to the coordinate planes of the machine. The position of the inclined surface can be defined via the oblique position of the coordinate system (refer to FRAME programming).

## **Increment**

The end point for the traversing of axes is defined by the distance to be covered and a direction that refers to a point already reached. See also -> Absolute dimension.

Specification of the length of the travel path in increments. The number of increments can be stored either in the -> setting data or it can be selected with the keys 10, 100, 1000 and 10 000.

**Initialization block**

Initialization blocks are special -> program blocks. These contain values that must be assigned before executing the program.  
Initialization blocks are the preferred choice for initializing previously defined data or global user data.

**Initialization files**

An initialization file can be created for each -> workpiece. Different instructions for variable values that are valid exclusively for a workpiece, can be stored in the initialization file.

**Intermediate blocks**

Movements with a selected tool offset (G41/G42) can be interrupted by a limited number of intermediate blocks (blocks without traversing movements at the compensation plane). The tool offset can be calculated correctly while using intermediate blocks. The number of intermediate blocks that can be read in advance by the control system can be set in the system parameters.

**Interpolation cycle**

The interpolation cycle is a multiple of the basic system cycle. The interpolation cycle is used to specify the cycle time that is required to update the setpoint interface with the position controls. The resolution of the velocity profiles is determined with the interpolation cycle.

**Interpolator**

Logical unit of the -> NCK which can be used to determine the intermediate values for the movements of the individual axes to be executed on the basis of the target positions specified in the part program.

**Interpolatory compensation**

Interpolatory compensation is a means of compensating the leadscrew error (SSFK) and measuring system error (MSF) that result from the production process.

**Interrupt routine**

Interrupt routines are special -> subprograms which can be started by events (external signals) in the machining process. Here, the block of the part program just processed is interrupted and the axis position at the interruption point is stored automatically. See -> ASUP

## Interrupts

All -> messages and alarms are displayed in plain text on the operator panel. The alarm text contains the date, time and a suitable symbol for the cancel criterion.

Alarms and messages are displayed separately according to the following criteria:

- 1. Alarms and messages in the part program:  
Alarms and messages can be displayed directly from the program as plain text.
- 2. PLC alarms and messages that concern the machine can be displayed directly from the PLC in plain text format. No additional function blocks are required for this.

## Inverse-time feedrate

In the SINUMERIK 840D sl controls, the speed of the axis motion can be specified (G93) instead of the feedrate for traversing the distance stored in a block.

## JOG

Operating mode of the CNC (in the setup operation): The machine can be setup in the JOG mode. The individual axes and spindles can be traversed in steps (in the JOG mode) with direction keys. Other functions offered by the JOG mode are -> Reference point approach, -> REPOS (reposition) -> Preset -> (actual value specification).

## Keyswitch

S7-300: In the S7-300, the key-operated switch is the mode selector switch in the -> CPU. The key-operated switch is operated with a removable key.

840D sl: The key-operated switch on the -> machine control panel has four positions to which the operating system of the control system has assigned appropriate functions. Each key-operated switch has three differently colored keys which can be removed in the specified positions.

## Languages

The texts of the user interface, system messages and alarms are available in five system languages: English, German, French, Italian and Spanish. The user can always select two of the languages listed in the control system.

## Leadscrew error compensation

Compensation for the mechanical inaccuracies of a spindle involved in the feed movement. Errors are compensated by the control system on the basis of the measured deviations that are stored in the control system.

### **Linear axis**

In contrast to a rotary axis, a linear axis describes a straight line.

### **Linear interpolation**

In the linear interpolation, the tool moves along a straight line to the end point during machining of the workpiece.

### **LookAhead**

The "LookAhead" function is a means of optimizing the machining speed through looking ahead using a parameterizable number of traversing blocks.

### **LookAhead for contour violations**

The control system recognizes and reports the following types of collision:  
The travel path is shorter than the tool radius.  
The width of the internal corner is smaller than the tool diameter.

### **Machine control panel**

A control panel in the machine tool having operator controls such as keys, rotary switches, etc., as well as simple displays such as LEDs. The machine control panel is used for direct control of the machine tool via the PLC.

### **Machine coordinate system**

Coordinate system based on the axes of the machine tool.

### **Machine zero**

A fixed point on the machine tool to which all measuring systems (derived from it) refer.

### **Machining channel**

The channel structure can be used to reduce downtime by executing motional sequences simultaneously. Thus, for instance, the gantry of a loader can execute its movements during the machining. In this case, the CNC functions as an autonomous control system which performs operations like decoding, block preparation and interpolation independently.

## Macros

Several instructions in different programming languages can be combined with each other in one instruction. This abbreviated sequence of instructions is called in the CNC program under a user-defined name. The instructions are executed one after the other with the macro.

## Main block

A block that is preceded by a ":" sign and that contains all the parameters necessary for starting the processing of a -> part program.

## Main program

A -> part program that is denoted by a number or a name in which other main programs, subprograms or -> cycles can be called.

## Main run

Part program blocks that have been decoded and prepared via block preparation, are processed in the "main run".

## MDA

Control operating mode: Manual Data Automatic = manual data input in automatic mode. Individual program blocks or block sequences with no reference to a main program or subprogram can be input in MDA mode; these are processed immediately on actuating the NC start key.

## Memory reset

The following -> CPU memories are erased by a memory reset operation:

- -> Work memory
- Read/write area of the -> load memory
- -> System memory
- -> Backup memory

## Metric or inch dimensions

Position and thread lead values can be programmed in the machining program in inches. The control is always set to the basic system independently of the programmed unit of measurement (G70/G71).

**Metric units system**

Standardized system of length units in millimeters, meters, etc.

**Mirroring**

Mirroring changes the sign of the coordinate value of a contour with reference to an axis. Mirroring can be executed simultaneously for several axes.

**Mode group**

All axes/spindles are assigned at a randomly specified time to a single channel. Each channel is assigned to a mode group (BAG). The same -> mode is always assigned to the channels in a mode group.

**Motion synchronization**

This function can be used to trigger actions that run simultaneously (synchronously) with the machining. The starting point of the actions is defined by a condition (e.g. the status of a PLC input, the time that has elapsed since the start of a block). The start of motion-synchronized actions is not linked to block limits.

Examples of typical motion-synchronized actions are: Transfer of M- and H-(auxiliary) functions at the PLC or distance-to-go delete for certain axes.

**NC**

"Numerical Control" = numerical control system; contains all components of the control system of the machine tool: -> NCK, -> PLC, -> HMI, -> COM.

**NCK**

Numerical Control Kernel: Component of the NC control system, which executes -> part programs and essentially coordinates the movements at the machine tool.

**Net, Network**

A net or network is the connection of several S7-300 and other automation or HMI devices such as programming devices via -> connecting cables. The interconnected devices exchange data across the network.

**NURBS**

Motion control and path interpolation are executed in the control system internally on the basis of NURBS (non-uniform rational B splines). Thus, there is a standard procedure (SINUMERIK 840D sl) for all operating modes as an internal control function.

## **OEM**

The scope of the implementation of individual solutions (OEM applications) for the SINUMERIK 840D sl was developed for machine manufacturers, who want to create their own user interfaces or who want to integrate process-oriented functions in the control system.

## **Online tool offset**

This function can be used for grinding tools only.

The reduction in the size of the grinding wheel through dressing is transferred to the relevant active tool as tool offset and is immediately effective.

## **Operating mode**

Operator control concept for SINUMERIK control systems. The following modes exist: -> JOG, -> MDA and -> AUTOMATIC.

## **Oriented spindle stop**

Stops the spindle at a defined orientation angle, e.g. for the execution of additional machining operation at the specified position.

## **Oriented tool retraction**

RETOOL: If the machining is interrupted (e.g. in case of tool breakage), the tool can be retracted with a program command along a fixed path in an orientation defined by the user.

## **Override**

Manually adjustable or programmable feature of the control system, with which the users can superimpose the programmed feedrates and speeds to adjust them to their special tool or material.

## **Part program**

A sequence of instructions to the NC control whose combination is supposed to create a certain -> workpiece by executing certain DO operations on a specified -> raw part.

**Part program management**

The "part program management" function can be organized according to the -> workpieces. The number of programs and data to be managed depends on the capacity of the control memory and can also be configured via machine data settings. A name consisting of a maximum of 16 alphanumeric characters can be assigned to each file (programs and data).

**Path axis**

Path axes are all the machining axes of a -> channel that is controlled by the -> interpolator in such a way that they start together, accelerate at the same time and reach their end position at the same time.

**Path feedrate**

The path feedrate acts on the -> path axes. It represents the geometrical sum of the feedrates of the participating -> path axes.

**Path velocity**

The maximum programmable path velocity depends on the input resolution. The maximum programmable path velocity at a resolution of, say, 0.1 mm is 1.000 m/min.

**PCIN data transfer program**

PCIN is a routine for transfer and reception of CNC user data, such as part programs, tool offsets etc. over a serial interface. The PCIN program runs on standard commercial PCs under MS-DOS.

**PLC**

Programmable Logic Control -> programmable logic controller. Component of -> NC: Programmable control system for executing the control logic of the machine tool.

**PLC program memory**

The PLC user program, the user data and the PLC main program are stored together in the PLC user memory of the PLC. The PLC user memory can be expanded to 128 KB.

**PLC programming**

The PLC is programmed using the STEP 7 software. The STEP 7 programming software is based on the WINDOWS standard operating system and contains the functionality of STEP 5 programming with innovative enhancements and developments.

### **Polar coordinates**

A coordinate system, in which the position of a point in the plane is defined with a defined axis from the perspective of its distance from the coordinate origin and the angle formed by the radius vector.

### **Polynomial interpolation**

Polynomial interpolation is a means with which a very large range of curve characteristics including straight line, parabola and exponential functions can be created.

### **Positioning axis**

An axis that executes auxiliary movement in the machine (e.g. tool magazine, palette transport). Positioning axes are axes that do not interpolate with -> path axes.

### **Preprocessing memory, dynamic**

The traversing blocks are prepared before their execution (pre-processed) and stored in a "preprocessing" memory. Block sequences can be executed from this memory at a very high speed. Blocks can be loaded in the preprocessing memory continuously during the machining.

### **Preprocessing stop**

Program command The next block in a part program is executed only if all the earlier prepared blocks stored in the preprocessing memory have been processed.

### **Preset**

The machine zero point can be redefined in the machine coordinate system by means of the preset function. No axis is traveled during preset; instead, a new position value is input for the current axis position.

### **Programmable frames**

New starting points of a coordinate system can be defined dynamically with programmable -> frames when the program is running. One differentiates between absolute definitions in which new frames are used, and additive definitions in which the definition is undertaken with reference to an existing starting point.

**Programmable logic controller**

Programmable logic controllers (PLC) are electronic control systems whose functions are stored in the control system as a program. Hence, the structure and wiring do not depend on the control functions. Programmable logic controllers are designed exactly like a computer, i.e. they consist of a CPU with memory, I/O modules and an internal bus system. The selection of the I/O modules and the programming language depends on the technology used.

**Programmable working area limitation**

Limiting the traversing range of the tool to defined, programmable limits.

**Programming key**

Characters and character strings with an accurately defined meaning within the programming language for -> part programs (refer to programming manual).

**Protection zone**

Three-dimensional zone inside a -> working area which the tool cannot reach (can be programmed via MD).

**Quadrant error compensation**

Contour errors in quadrant transitions that are caused by frictional losses on the guide rails can be corrected to a large extent with the quadrant error compensation. A circularity test is used to define the parameters of the quadrant error compensation.

**Rapid traverse**

The maximum rapid traverse rate of an axis is, for instance, used to traverse a tool from the idle position to the -> workpiece contour or to retract the tool from the workpiece contour.

**Reference point**

Point on the machine to which the measuring system of the -> machine axis refers.

**Reference point approach**

If the position measuring system used is not an absolute encoder, then reference point approach must be started so that the actual values delivered by the measuring system match the machine coordinate system values.

## REPOS

1. Repositioning to the contour, triggered by the operator

REPOS can be used to reset the tool to the interruption point with direction keys.

2. Programmed repositioning to the contour

A selection of approach strategies is available in the form of program commands:  
Approaching the interruption point, approaching the starting block, approaching the last block, approaching a point on the path between block start and interruption point.

## Revolutional feedrate

The axis feedrate is set in the channel as a function of the main spindle speed (programming with G95).

## Rigid tapping

This function is used for tapping without compensating chuck. In this case, the spindle is controlled as an interpolatory rotary axis with the effect that the threads are drilled up to the final drilling depth, e.g. in tapping of blind holes (precondition: Spindle is operated as axis).

## Rotary axis

Rotary axes are used to rotate the tool or workpiece by a certain angle.

## Rotary axis, continuously turning

The traversing range of a rotary axis can be set either to a modulo value (can be adjusted via machine data) or as endlessly turning in both directions, depending on the relevant application. Endlessly turning rotary axes are used, e.g. for non-circular machining, grinding and winding jobs.

## Rotation

Component of a -> frame which defines a rotation of the coordinate system through a specific angle.

## Rounding axis

Rounding axes enable the workpiece or tool to turn by a certain angle that is stored in the indexing grid. The rounding axis is "in position" when the latching is reached.

## R-parameters

R-parameters. The programmer can assign or query the values of the R-parameters, if necessary in the -> part program.

**S7 configuration**

"S7 configuration" is a tool for defining the parameters of modules. "S7 configuration" can be used to set different -> parameter sets of the -> CPU and the I/O modules in the -> programming device. These parameters are loaded to the CPU.

**S7-300 bus**

The S7-300 bus is a serial data bus that supplies the appropriate voltage to the modules, and that is used by these modules for data interchange. The modules are interconnected via plug-in bus connectors.

**Safety functions**

The control system has constantly active monitoring functions with which errors in the -> CNC, the programmable control system (-> PLC) and the machine can be detected early enough to avoid damages to the workpiece, tool or machine to a large extent. The machining is interrupted and the drives are stopped when a fault or error occurs. The cause of the error is recorded and an alarm is output. At the same time, the PLC is notified that a CNC alarm has been triggered.

**Safety Integrated**

Effective protection according to the EU Guideline >>89/392/EEC<<, >>Safety Integrity Level 3<< according to EN-954-1 (the B 1-4 classes are defined in this standard) for the safety of operators and machines, integrated in the control system for safe installation and testing.

Failure safety is guaranteed. This safety function is also effective during individual faults.

**Scaling**

Component of a -> frame through which axis-specific changes are made.

**Set**

All files that are required for the programming and execution of a program are denoted as sets.

A section of a -> part program which ends with "LineFeed" (line break). A distinction is made between -> main blocks and -> subblocks.

## Setting data

Data that supplies information on the features of the machine to the control system; the method is defined in the system software. Unlike -> machine data, setting data can be modified by the user.

## Softkey

A key whose name is displayed in one part of the screen. The selection of the softkeys that are displayed is adjusted automatically to the relevant operating state. The freely programmable function keys (softkeys) are assigned to certain functions that are defined via the software.

## Software limit switch

The limits of the traversing range of an axis is defined with software limit switches and this prevents the carriage from coming in contact with the hardware limit switches. Two pairs of values can be assigned per axis, and these can be activated separately via the -> PLC.

## Speed limitation

Maximum/minimum (spindle) speed: The maximum spindle speed is limited by the values that are specified either in the machine data by the-> PLC or the -> setting data.

## Spindles

The spindle functionality is a construct with two levels:

Spindles: Speed or position-controlled spindle drives, analog digital (SINUMERIK 840D sl)

Auxiliary spindles: Speed-controlled spindle drives without encoder, e.g. for power tools.

## Spline interpolation

The control system can generate a smooth curve with spline interpolation, for which a small number of interpolation points are necessary along a programmed contour.

## Standard cycles

Standard cycles are used to program DO operations that are repeated frequently:

- For drilling/milling
- For measuring tools and workpiece

The available cycles are listed in the "Cycle support" menu in the "Program" operating area. The required parameters for the assignment of the values are displayed in plain text after the desired execution cycle has been selected.

**Station number**

The station number represents the "contact address" of a -> CPU or a -> programming device or any other intelligent I/O module if these communicate with each other via a -> network. The station number is assigned to the CPU or the programming device with the S7 tool -> "S7 configuration".

**Subblock**

Block preceded by "N" containing information about a machining step, e.g. a position specification.

**Subprogram**

A sequence of instructions of a -> part program which can be called multiple times with various output parameters. Subprograms are always called from the main programs. Subprograms can be locked to prevent unauthorized export and impermissible viewing. -> Cycles are subprograms from the perspective of type.

**Synchronization**

Instructions in -> part programs for coordinating working cycles in different -> channels at certain machining points.

**Synchronized actions**

- Auxiliary function output

Technological functions (-> auxiliary functions) can be output by the CNC program to the PLC during the machining of a workpiece. These auxiliary functions can, for instance, be used to control ancillary equipment on the machine (quill, gripper, chuck etc.).

- Fast auxiliary function output

The acknowledgement times of -> auxiliary functions can be reduced and unnecessary machining process for executing less critical switching functions can be avoided.

Synchronized actions can be combined to build programs (technology cycles). Axis programs can be started in the same IPO cycle, e.g. by scanning digital inputs.

**Synchronized axes**

Synchronized axes require the same time to traverse the path as -> geometry axes.

### **Synchronous spindle**

Accurate synchronism of an angle between a master spindle and one or more slave spindles. This way, the flying transfer of a workpiece from Spindle 1 to Spindle 2 is possible in turning machines.

In addition to speed synchronism, relative angle positions of the spindles can also be programmed, e.g. "floating" or the position-oriented transfer of inclined workpieces.

Several pairs of synchronous spindles can be implemented.

### **System variables**

A variable that exists even though it was not programmed by -> part program programmers. It is defined by the data type and the variable name with the sign \$. See also -> User-defined variable.

### **Teach-in**

Teach-in is a means of creating and correcting part programs. The individual program blocks can be input via the keyboard and executed immediately. The positions approached with the direction keys or the handwheel can also be stored. Additional information such as G functions, feedrates or M functions can be entered in the same block.

### **Tool**

A tool that is used to shape a workpiece. Tools are, e.g. turning tools, milling cutters, twist drills, laser beams, grinding wheels, etc.

### **Tool offset**

A tool is selected by programming a T function (5 digits, integer) in the block. Every T number can be assigned up to 9 cutting edges (D addresses). The number of tools that are managed in the control system can be parameterized.

The tool length compensation is selected by programming D numbers.

### **Tool radius compensation**

A contour is programmed on the basis of the assumption that a tool is used with a tool tip. As this is not always the case in practice, the radius of curvature of the used tool is specified so that an allowance is considered by the tool. The center of curvature is guided to the contour which is equidistant with an offset that corresponds to the radius of curvature.

**Transformation**

If programming is done in one Cartesian coordinate system and execution is done in another non-Cartesian coordinate system (e.g. with machine axis as rotary axis); then inclined axes and 5-axis transformation are used in connection with transmit.

**Transmit**

This function can be used to mill outside contours on rotating parts, e.g. four-sided parts (linear axis with rotary axis).

3D interpolation with two linear axes and one rotary axis is also possible. The advantages of Transmit simplify programming and improve the efficacy of the machine through complete machining: Turning and milling can be executed on the same machine without rechucking.

**Traversing range**

The maximum possible traversing range for linear axes is  $\pm 9$  decades. The absolute value depends on the selected resolution for the data input and position control, and on the used unit of measurement (inch or metric).

**User interface**

The user interface (UI) is the human machine interface (HMI) of a CNC. It appears on the screen and has eight horizontal and eight vertical softkeys.

**User memory**

All programs and data, such as part programs, subprograms, comments, tool offsets, work offsets/frames, as well as channel and program user data, can be stored in the shared CNC user memory.

**User-defined variables**

Users have the option of defining variables for their own purposes in the -> part program or in a data block (global user data). The variable definition contains the specification of the data type and the variable name. See also -> System variable.

### Variable definition

A variable is defined by specifying a data type and a variable name. The variable name can be used to access the value of the variable.

### Velocity control

The control system can execute a look-ahead evaluation over several blocks (-> LookAhead) to achieve an acceptable traversing speed during movements that require only very small adjustments of position in a block.

### Vocabulary words

Words with a certain notation and a defined meaning in the programming language for -> part programs.

### Work memory

The work memory is a memory offering free access (RAM or Random Access Memory) in the -> CPU which the processor accesses while executing the application program.

### Work offset

Specification of a new reference point for a coordinate system by referring to an existing zero point and a -> Frame.

#### 1. Settable

SINUMERIK 840D sl: There is a parameterizable number of adjustable work offsets for each CNC axis. Each work offset can be selected via G functions; the selection is exclusive.

#### 2. External

All movements that define the position of the workpiece zero can be superimposed by an external work offset, which is either defined by a handwheel (DRF movement) or the PLC.

#### 3. Programmable

The TRANS instruction can be used to program work offsets for all path and positioning axes.

### Working area

Three-dimensional zone in which the tool tip can be moved due to the physical design of the machine. See also -> Protection zone.

**Workpiece**

The part that is produced/machined on the machine tool.

**Workpiece contour**

Programmed contour of the -> workpiece to be made/machined.

**Workpiece coordinate system**

The origin of the workpiece coordinate system is the -> workpiece zero. For working cycles that are programmed in the workpiece coordinate system, the dimensions and directions refer to this system.

**Workpiece zero**

The workpiece zero is the origin of the -> workpiece coordinate system. It is determined based on its distance from the machine zero point.



# Index

## A

Absolute/incremental dimensioning, 49  
Additional function, 64

## B

Basic coordinate system, 44, 45  
Block skip level, 12

## C

Checking the return to the reference point, 33  
Checkpoint, 58  
Circular interpolation, 21  
Comments, 11  
Compressor, 128  
Compressor function, 128  
Constant cutting rate, 62  
Coordinate system, 43  
Cutting of multiple-start thread, 39  
Cutting of thread with variable lead, 41  
Cylindrical interpolation, 27

## D

Decimal point, 10  
Deep hole drilling and grooving in the transverse axis, 92  
Diameter and radius commands for the X-axis, 52  
Drilling cycle front face, 106  
Drilling cycle side surface, 106  
DryRun mode, 129  
Dwell time, 54

## E

Error Messages, 179

## F

F function, 13  
Finishing cycle, 87  
Function program interruption, 130

## G

G code  
  Display, 8  
G code system A, 9  
G00, 13, 17, 19, 141  
  Linear interpolation, 19  
G01, 20, 141  
G02, 141  
G02, G03, 21  
G03, 141  
G04, 54, 143  
G05, 126, 143  
G05.1, 143  
G07.1, 27, 143  
G10, 143  
G10.6, 113, 143  
G12.1, 143  
G12.1, G13.1, 29  
G13.1, 143  
G17, 142  
G18, 142  
G19, 142  
G20, 141  
G20, G21, 53  
G21, 141  
G22, 142  
G23, 142  
G27, 33  
G28, 32, 143  
G290, 8, 143  
G291, 8, 143  
G30, 34, 143  
G30.1, 143  
G31, 113, 143  
G31, P1 - P4, 115  
G33, 35, 38, 39, 141  
G34, 41, 141  
G35, 141  
G36, 141  
G40, 141  
G40, G41/G42, 57  
G41, 141  
G42, 141  
G50.2, 143  
G51.2, 143  
G52, 143

G53, 44, 143  
G54, 142  
G54 P{1...48}, 142  
G54 P0, 142  
G55, 142  
G56, 142  
G57, 142  
G58, 142  
G59, 142  
G60, 143  
G65, 143  
G65, G66, G67, 119  
G66, 142  
G67, 142  
G68, 141  
G69, 141  
G70, 87, 143  
G71, 80, 143  
G72, 84, 143  
G73, 143  
G74, 91, 143  
G75, 92, 143  
G76, 93, 143  
G77, 141  
G78, 141  
G79, 141  
G80, 142  
G80 to G89, 96  
G83, 100, 142  
G83 or G87, 103  
G83, G87, 100, 101  
G84, 105, 142  
G85, 106, 142  
G87, 100, 142  
G88, 105, 142  
G89, 106, 142  
G90, 49, 141  
G91, 49, 141  
G92, 45, 143  
G92.1, 46, 143  
G93, 15  
G94, 15, 141  
G95, 15, 141  
G96, 141  
G96, G97, 62  
G97, 62, 141  
G98, 142  
G98/G99, 99  
G99, 142

## H

HMI, 135

## I

Input inch/metric, 53  
Interpolation commands, 17  
Interrupt program with M96/M97, 115  
Involute interpolation, 26  
ISO dialect mode, 7

## L

Linear feed per minute, 15  
Linear interpolation, 20

## M

M function, 64  
M functions for stopping operations, 65  
M functions that can be used in many ways, 68  
M00, 65  
M01, 65  
M02, 65  
M30, 65  
M96, 115  
M96, M97, 130  
M97, 115  
M98, M99, 109  
Macro program call, 119, 126  
Macro programs, 119  
Maximum programmable values for axis movements, 9  
Modal call, 122  
Multiple repetitive cycles, 79  
Multiple-thread cutting cycle, 93

## O

Operating modes  
    Switching over, 8

## P

Path feed, 13  
Pattern repeat cycle, 91  
Polar-coordinates interpolation, 29  
Positioning, 17

**R**

Rapid retraction, 113  
Rapid traverse, 13, 17  
Reference point selection, 34  
Revolutional feedrate, 15

**S**

S function, 62  
Second additional function, 68  
Setting data  
    List, 176  
Siemens mode, 7  
Simple call, 119  
Skip block, 12  
Skip level, 129  
Specification of several M functions in one block, 68  
Spindle function, 62  
Straight facing cycle, 77  
Subprograms, 119

**T**

Tapping cycle, 80, 86  
    Transverse axis, 84  
Tapping cycle front face, 105  
Tapping cycle side surface, 105  
Thread cutting, 35  
Thread cutting cycle, 72  
Thread cutting function, 35  
Time inverse feed, 15  
Tool length compensation, 56  
Tool nose radius compensation, 57  
Tool offset data memory, 55  
Tool offset functions, 55  
Transverse-taper turning cycle, 78

