SIEMENS

SINUMERIK 802D sl/840D/840D sl/ 840Di//840Di sl/810D

Programming Manual ISO Milling

Programming Guide

Valid for

Software	*Version*
SINUMERIK 840D/DE powerline	7.4
SINUMERIK 840Di/DiE powerline	3.3
SINUMERIK 810D/DE powerline	7.4
SINUMERIK 840D sl/DE sl	1.4
SINUMERIK 840Di sl/DiE sl	1.4
SINUMERIK 802D sl	1.4

04.2007 Edition

SINUMERIK® documentation

Printing history

Brief details of this edition and previous editions are listed below.

The status of each edition is shown by the code in the “Remarks” column.

Status code in the “Remarks” column:

A . . . . . New documentation.
B . . . . . Unrevised reprint with new Order No.
C . . . . . Revised edition with new status.

Edition	Order No.	Remarks
02.01	6FC5 298–6AC20–0BP0	**A**
12.01	6FC5 298–6AC20–0BP1	**C**
11.02	6FC5 298–6AC20–0BP2	**C**
04.05	6FC5 298–7AC20–0BP0	**C**
04.07	6FC5398–7BP10–0BA0	**C**

Trademarks
All product designations could be trademarks or product names of Siemens AG or other companies which, if used by third parties, could infringe the rights of their owners.

Exclusion of liability
We have checked the contents of the documentation for consistency with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee complete conformance. The information in this document is regularly checked and necessary corrections are included in reprints. Suggestions for improvement are also welcome.

Siemens-Aktiengesellschaft Siemens-Aktiengesellschaft

Preface

Structure of the documentation

The SINUMERIK documentation is structured in three levels:

- General documentation
- User documentation
- Manufacturer/service documentation.

An overview of publications that is updated monthly is provided in a number of languages in the Internet at:

http://www.siemens.com/motioncontrol

Follow menu items > ”Support” > ”Technical Documentation” > ”Overview of Documents”.

DOConWEB, the Internet edition of DOConCD, is available at:

http://www.automation.siemens.com/doconweb

Information on the training courses offered as well as FAQs (frequently asked questions) are provided on the Internet at:

http://www.siemens.com/motioncontrol, ”Support” menu item.

Target audience

This documentation is intended for:

- Project engineers
- Technologists (from machine manufacturers)
- System startup (Systems/Machines
- Programmers

Standard scope

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

It may be possible to runfunctions that are not described in this document in your controller. This does not, however, represent an obligation to supply such functions with a new control or when servicing.

Further, for the sake of simplicity, this documentation does not contain all detailed information about all types of the product and cannot cover every conceivable case of installation, operation or maintenance.

Technical Support

If you have any questions, please get in touch with our Hotline:

	Europe / Africa	**Asia / Australia**	**America**
Phone	+49 180 5050 222	+86 1064 719 990	+1 423 262 2522
Fax	+49 180 5050 223	+86 1064 747 474	+1 423 262 2289
Internet	http:// www.siemens.com/automation/support-request		
E-Mail	mailto:adsupport@siemens.com		

Note

Should you require technical support, please call one of the country-specific phone numbers provided on the Internet:

http://www.siemens.com/automation/services&support

Questions regarding the manual

If you have any queries (suggestions, corrections) in relation to this documentation, please send a fax or e-mail to the following address:

Fax	+49 9131 98 63315
E-Mail	mailto:motioncontrol.docu@siemens.com

Fax form: see reply form at the end of the manual.

SINUMERIK Internet address

http://www.siemens.com/sinumerik

Origin

In contrast to the Siemens mode programming of YASKAWA SIEMENS 840DI, ISO dialect programming is mainly based on SINUMERIK 6T-B and SINUMERIK 6M-B, a CNC control which had already been phased out. However, OEM and end-user requirements on SINUMERIK 6T-B programming compatibility lead to the development of the ISO dialect function.

Safety Instructions

This manual contains information which you should carefully observe to ensure your own personal safety and the prevention of material damage. These notices referring to your personal safety are highlighted by a safety alert symbol. The notices referringto property damage alone have no safety alert symbol. The warnings appear in decreasing order of risk as given below.

Danger

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

Warning

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

Caution

with a warning triangle indicates that minor personal injury can result if proper precautions are not taken.

Caution

without warning triangle indicates that material damage can result if proper precautions are not taken.

Notice

indicates that an undesirable event or state may arise if the relevant notes are not observed.

If several hazards of different degree occur, the hazard with the highest degree must always be given priority. If a warning note with a warning triangle warns of personal injury, the same warning note can also contain a warning of material damage.

Qualified personnel

The associated device/system may only be set up and operated using this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Qualified persons are defined as persons who are authorized to commission, to ground, and to tag circuits, equipment, and systems in accordance with established safety practices and standards.

Prescribed Usage

Please note the following:

Warning

The equipment may only be used for single purpose applications explicitly described in the catalog and in the technical description and it may only be used along with third−party devices and components recommended by Siemens. To ensure trouble−free and safe operation of the product, it must be transported, stored and installed as intended and maintained and operated with care.

Further notes

Note

This icon is displayed in the present documentation whenever additional facts are being specified.

Table of Contents

Notes

1 Programming Basics

1.1 Introductory explanations

1.1.1 Siemens mode

The following conditions apply when Siemens mode is active:

- Siemens G commands are interpreted on the control by default. This applies to all channels.
- It is not possible to extend the Siemens programming system with ISO Dialect functions because some of the G functions have different meanings.
- Downloadable MD files can be used to switch the control to ISO Dialect mode. In this case, the system boots the ISO Dialect mode by default.

1.1.2 ISO Dialect mode

The following conditions apply when ISO Dialect mode is active:

- Only ISO Dialect G codes can be programmed, not Siemens G codes.
- It is not possible to use a mixture of ISO Dialect code and Siemens code in the same NC block.
- It is not possible to switch between ISO Dialect-M and ISO Dialect-T via G command.
- Siemens subprogram calls can be programmed.
- If further Siemens functions are to be used, it is necessary to switch to Siemens mode first.

1.1.3 Switchover

The following two G commands are used to switch between Siemens mode and ISO Dialect mode:

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

The active tool, the tool offsets and the zero offsets are not changed by this action.

G290 and G291 must be programmed in a separate program block.

1.1.4 G code display

The G code display must always be implemented in the same language type (Siemens/ISO Dialect) as the current block display. If the block display is suppressed with DISPLOF, the current G codes continue to be displayed in the language type of the active block.

Example

The Siemens standard cycles are called up using the G functions of the ISO Dialect mode. DISPLOF is programmed at the start of the cycle, with the result that the ISO Dialect G commands remain active for the display.

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

Procedure

External main program calls Siemens shell cycle. Siemens mode is selected implicitly on the shell cycle call.

DISPLOF freezes the block display at the call block; the G code display remains in external mode. This display is refreshed while the Siemens cycle is running.

The SAVE attribute resets the G codes modified in the shell cycle to their original state when the shell cycle was called on the return jump to the main program.

1.1.5 Maximum number of axes/axis designation

In ISO Dialect–M the maximum number of axis is 9. Axis designation for the first three axes is fixed to X, Y and Z. Further axes can be designated A, B, C, U, V, W.

1.1.6 Decimal point programming

There are two notations for the interpretation of programming values without a decimal point in ISO Dialect mode:

- **pocket calculator type notation**
  Values without decimal points are interpreted as mm, inch or degrees.
- **standard notation**
  Values without decimal points are multiplied by a conversion factor.

The setting is defined by MD 10884.

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

Example of linear axis in mm:
X 100.5 corresponds to value with decimal point: 100.5mm
X 1000 pocket calculator type notation: 1000mm
standard notation: IS-B: 1000* 0.001= 1mm
IS-C: 1000* 0.0001 = 0.1mm

ISO-Dialekt Milling

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

Address	Unit	IS-B	IS-C
Linear axis	mm inch	0.001 0.0001	0.0001 0.00001
Rotary axis	deg	0.001	0.0001
F feed G94 (mm/inch per min.)	mm inch	1 0.01	1 0.01
F feed G95 (mm/inch per min.)	mm inch	0.01 0.0001	0.01 0.0001
F thread pitch	mm inch	0.01 0.0001	0.01 0.0001
C chamfer	mm inch	0.001 0.0001	0.0001 0.00001
R radius, G10 toolcorr	mm inch	0.001 0.0001	0.0001 0.00001
Q	mm inch	0.001 0.0001	0.0001 0.00001
I, J, K interpolation parameters	mm inch	0.001 0.0001	0.0001 0.00001
G04 X or U	s	0.001	0.001

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

Address	Unit	IS-B	IS-C
A contour angle	deg	0.001	0.0001
G74, G84 thread drilling cycles $MC_EXTERN_FUNCTION_MASK Bit8 = 0 F feedrate like G94, G95 Bit8 = 1 F thread pitch			

ISO dialekt Turning

Table 1-2 Different conversion factors for IS-B and IS-C

Address	Unit	IS-B	IS-C
Linear axis	mm inch	0.001 0.0001	0.0001 0.00001
Rotary axis	deg	0.001	0.0001
F feed G94 (mm/inch pro min.)	mm inch	1 0.01	1 0.01
F feed G95 (mm/inch pro Umdr.) $MC_EXTERN_FUNCTION_MASK			
Bit8 = 0	mm inch	0.01 0.0001	0.01 0.0001
Bit8 = 1	mm inch	0.0001 0.00000 1	0.0001 0.00000 1
F thread pitch	mm inch	0.0001 0.00000 1	0.0001 0.00000 1
C chamfer	mm inch	0.001 0.0001	0.0001 0.00001
R radius, G10 toolcorr	mm inch	0.001 0.0001	0.0001 0.00001
I, J, K interpolation parameters	mm inch	0.001 0.0001	0.0001 0.00001
G04 X or U		0.001	0.001
A contour angle		0.001	0.0001
G76, G78 thread drilling cycles $MC_EXTERN_FUNCTION_MASK Bit8 = 0 F feedrate like G94, G95 Bit8 = 1 F thread pitch			
G84, G88 thread drilling cycles $MC_EXTERN_FUNCTION_MASK Bit9 = 0 G95 F	mm inch	0.01 0.0001	0.01 0.0001

Table 1-2 Different conversion factors for IS-B and IS-C

Address	Unit	IS-B	IS-C
Bit8 = 1 G95 F	mm inch	0.0001 0.00000 1	0.0001 0.00000 1

1.1.7 Comments

In ISO dialect mode, round brackets are interpreted as comment characters.
In Siemens mode, ";" is interpreted as a comment. To simplify matters, ";" is also interpreted as a comment in ISO dialect model.
If the comment start character "(" is used again within a comment, the comment will not be terminated until all open brackets have been closed again.

Example:

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

In blocks N5 and N10 X100 Y100 is executed, in block N15 only Y100, as the first bracket is closed only after X100. Everything up to this position is interpreted as a comment.

1.1.8 Block skip

The skip character "/" can be anywhere within the block, even in the middle. If the programmed skip level is active at the moment of compiling, the block will not be compiled from this position to the end of the block. An active skip level therefore has the same effect as an end of block.

Example:

```
N5 G00 X100. /3 YY100      --> Alarm 12080,
N5 G00 X100. /3 YY100      --> No alarm when skip level 3 is active
```

Skip characters within a comment are not interpreted as skip characters.

Example:

```
N5 G00 X100. ( /3 part1 ) Y100          ;even when skip level 3 is
active, the
                           ;Y axis will be traversed
```

The skip level can be /1 to /9. Skip values <1 >9 give rise to alarm 14060
The function is mapped onto the existing Siemens skip levels. In contrast to ISO Dialect Original, / and /1 are separate skip levels and therefore have to be activated separately.

Note

- “0” can be omitted for “/0”.
- The optional block skip function is processed when a part program is read to the buffer register from either the tape or memory. If the switch is set ON after the block containing the optional block skip code is read, the block is not skipped.
- The optional block skip function is disregarded for program reading (input) and punch out (output) operation.

1.2 Basics of feed function

This section describes the feed function that specifies feedrate (distance per minute, distance per revolution) of a cutting tool.

1.2.1 Rapid traverse

Rapid traverse is used for positioning (G00) and manual rapid traverse (RAPID) operation. In the rapid traverse mode, each axis moves at the rapid traverse rate set for the individual axes; the rapid traverse rate is determined by the machine tool builder and set for the individual axes by using parameters. Since the axes move independently of each other, the axes reach the target point at different time. Therefore, the resultant tool paths are not a straight line generally.

Note

Setting units of rapid traverse rate
1 mm/min
0.1 inch/min
1 deg./min

Since the most appropriate value is set conforming to the machine capability, refer to the manuals published by the machine tool builder for the rapid traverse rate of your machine.

1.2.2 Cutting feed (F command)

Note

The unit ”mm/min” is normally used for feedrate for cutting tool in this manual, as long as there is especially no explanation.

The feedrate at which a cutting tool should be moved in the linear interpolation (G01) mode or circular interpolation (G02, G03) mode is designated using address character F.

With a 6-digit numeral specified following address character F, feedrate of a cutting tool can be designated in units of “mm/min”.

Refer to the manuals published by the machine tool builder for programmable range of the F code.

The upper limit of feedrates could be restricted by the servo system and the mechanical system. In this case, the allowable upper limit is set by MD and if a feedrate command exceeding this limit value is specified, the feedrate is clamped at the set allowable upper limit.

An F command specified in the simultaneous 2-axis linear interpolation mode or in the circular interpolation mode represents the feedrate in the tangential direction.

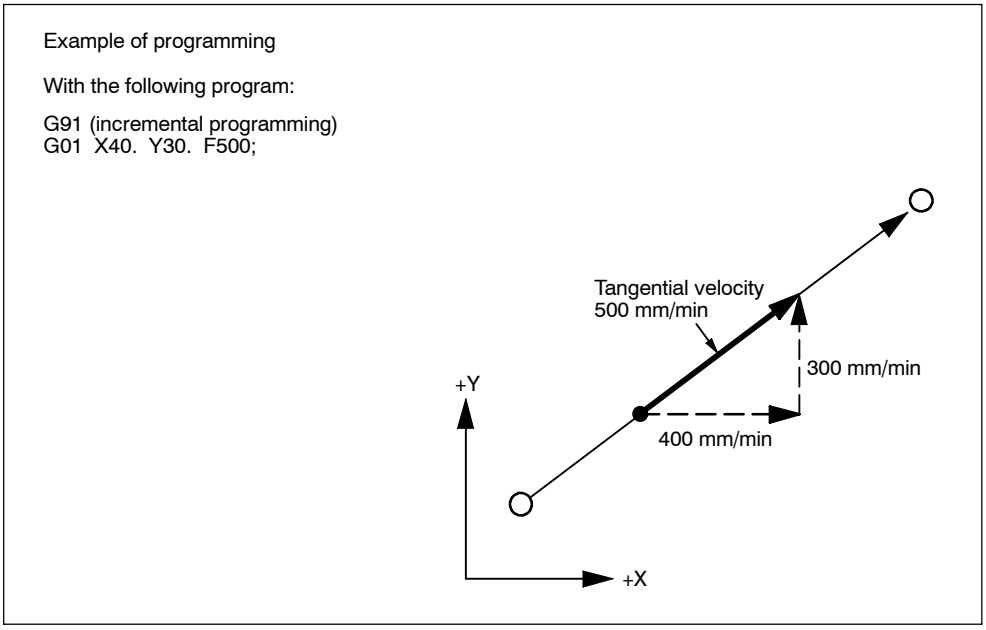


Fig. 1-1 F command in simultaneous 2-axis control linear interpolation

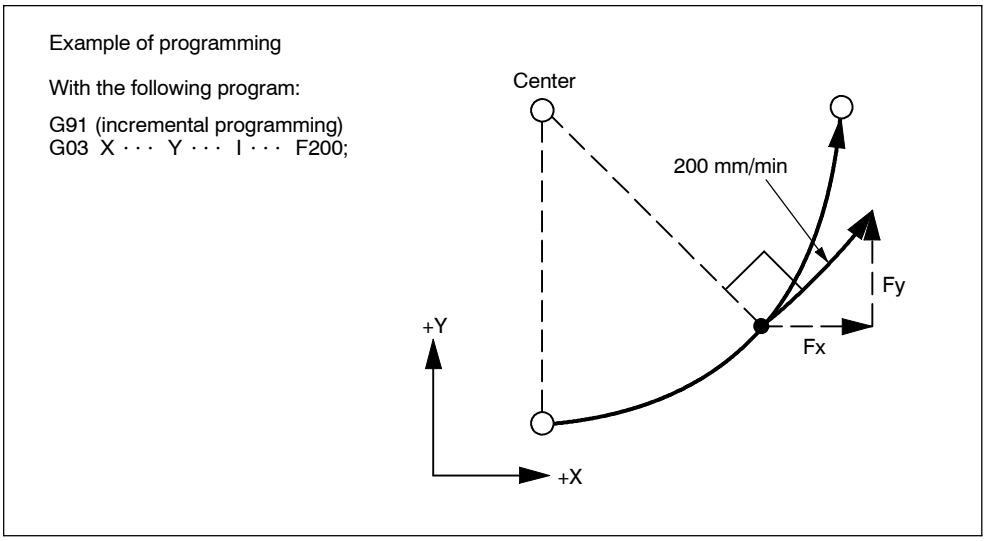


Fig. 1-2 F command in simultaneous 2-axis control circular interpolation

In the simultaneous 3-axis control linear interpolation, an F command indicates the tangential feedrate.

Example of programming

With the following program:
G01 X ··· Y ··· Z ··· F400;

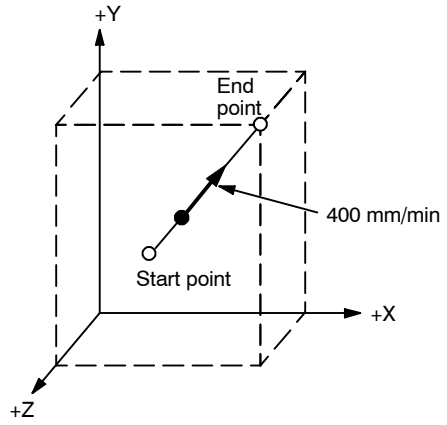


Fig. 1-3 F command in simulaneous 3-axis control linear interpolation

In the simultaneous 4-axis control linear interpolation, an F command indicates the tangential feedrate.

$$F\ (mm/min) = \sqrt{Fx^2 + Fy^2 + Fz^2 + F\alpha^2}$$

In the simultaneous 5-axis control linear interpolation, an F command indicates the tangential feedrate.

$$F\ (mm/min) = \sqrt{Fx^2 + Fy^2 + Fz^2 + F\alpha^2 + F\beta^2}$$

Note

1. If “F0” is specified and F 1-digit feed is not used, an alarm occurs.
2. For an F command, a minus value must not be specified. If a minus value is specified for an F command, correct operation cannot be guaranteed.

1.2.3 F1-digit feed

It is possible to select a feedrate by specifying a 1-digit numeral (1 to 9) following address F. With this manner of designation of an F command, the feedrate preset for the specified numeral is selected.
The F1-Digit Feed function needs to be enabled by MD setting as follows:

$MC_FIXED_FEEDRATE_F1_F9_ACTIV = TRUE: F1-Digit Feed enable
$MC_FIXED_FEEDRATE_F1_F9_ACTIV = FALSE: F1-Digit Feed disable

With the above mentioned MD set to FALSE, F1 to F9 in a machining program is interpreted as standard feed (F) programming, i.e. F2 = 2 mm/min. With the above mentioned MD set to TRUE, the feedrate to be selected in response to the designation of F1 to F9 should be set for the setting data indicated in Table 1-3. Feedrate 0 is activated if the corresponding value of the setting data is 0.

Table 1-3 Setting data used for preseting F1-digit feedrates

F command	Setting data
F1	$SC_EXTERN_FIXED_FEEDRATE_F1_F9[0]
F2	$SC_EXTERN_FIXED_FEEDRATE_F1_F9[1]
F3	$SC_EXTERN_FIXED_FEEDRATE_F1_F9[2]
F4	$SC_EXTERN_FIXED_FEEDRATE_F1_F9[3]
F5	$SC_EXTERN_FIXED_FEEDRATE_F1_F9[4]
F6	$SC_EXTERN_FIXED_FEEDRATE_F1_F9[5]
F7	$SC_EXTERN_FIXED_FEEDRATE_F1_F9[6]
F8	$SC_EXTERN_FIXED_FEEDRATE_F1_F9[7]
F9	$SC_EXTERN_FIXED_FEEDRATE_F1_F9[8]

Note: Input format=REAL

Note

1. If F1-digit command is activated by setting MD $MC_FIXED_FEEDRATE_F1_F9_ON = TRUE and F1 to F9 should not be used, be sure to program the feedrate F as a REAL value. For example, not F1 but F 1.0 for 1 mm/min.
2. If “F0” is specified, it is switched to rapid traverse mode (G00) automatically. Subsequently, G01 needs to be specified in order to use F1-digit command.
3. When the DRY RUN switch is ON, feed commands are all executed at the feedrate set for the dry run operation.
4. The feed override function is invalid for the feedrate selected by the F1-digit command.
5. The feedrate set for setting data is retained in memory if the power is turned OFF.
6. In a macro call using G65/G66, the value commanded with address F is always stored in system varible $C_F, meaning that numeral values 1 to 9 will stored.
7. If F1-digit command is used in a machining program containing a cycle call (G81 to G87), the feedrates are read from the corresponding setting data and stored into variable $C_F.

Example

$SC_EXTERN_FIXED_FEEDRATE_F1_9[0] = 1500.0
$SC_EXTERN_FIXED_FEEDRATE_F1_9[1] = 550.0

```
N10 X10 Y10 Z10 F0 G94 ; Positioning, rapid traverse
N20 G01 X150 Y30 F1 ; feedrate 1500 mm/min active
N30 Z0 F2 ; feedrate 550 mm/min active
N40 Z10 F0 ; Positioning, rapid traverse
```

1.2.4 Feed per minute function (G94)

When G94 is designated, a feedrate specified following address F is executed in units of “mm (inch)/min”.

1.2.5 Inverse time feed (G93)

When G93 is designated, a feedrate specified following address F is executed in units of “1/min”. G93 is a modal G code.

Example

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

i.e. the programmed distance will be moved within half a minute.

Notes

Commands Calling Axis Movements 2

2.1 Interpolation commands

This section describes the positioning commands and the interpolation commands that control the tool path along the specified functions such as straight line and arc.

2.1.1 Positioning (G00)

In the absolute programming mode (G90), the axes are moved to the specified point in a workpiece coordinate system, and in the incremental programming mode (G91), the axes move by the specified distance from the present position at a rapid traverse rate.

For calling the positioning, the following G codes can be used.

Table 2-1 G codes for positioning

G code	Function	Group
G00	Positioning	01

Positioning (G00)

Format

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

Explanation

When G00 is designated, positioning is executed. The program advances to the next block only when the number of lag pulses due to servo lag are checked after the completion of pulse distribution has reduced to the permissible value.

In the G00 mode, positioning is made at a rapid traverse rate in the simultaneous 3-axis (*5-axis) control mode. The axes not designated in the G00 block do not move. In positioning operation, the individual axes move independently of each other at a rapid traverse rate that is set for each axis. The rapid traverse rates set for the individual axes differ depending on the machine. For the rapid traverse rates of your machine, refer to the manuals published by the machine tool builder.

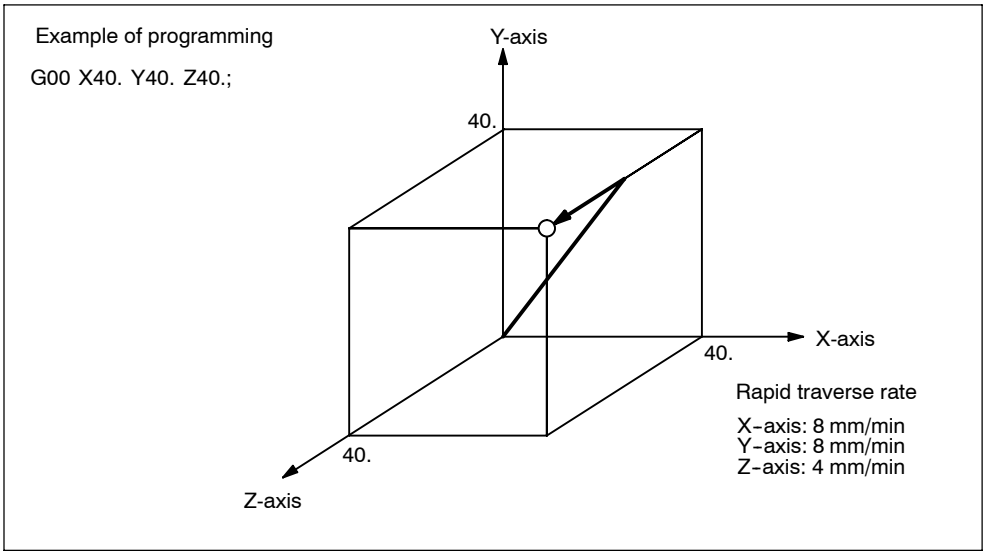


Fig. 2-1 Positioning in simultaneous 3-axis control mode

Note

In the G00 positioning mode, since the axes move at a rapid traverse rate set for the individual axes independently, the tool paths are not always a straight line. Therefore, positioning must be programmed carefully so that a cutting tool will not interfere with a workpiece or fixture during positioning.

G0 Linear Mode

The G0 linear mode is valid if MD $MC_EXTERN_G0_LINEAR_MODE is set. In this case, all programmed axes move in linear interpolation and reach their target position at the same point of time.

2.1.2 Linear interpolation (G01)

Format

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

With the commands of G01, linear interpolation is executed in the simultaneous 3-axis (*5-axis) control mode. The axes not designated in the G01 block do not move. For the execution of the linear interpolation, the above command must be specified.

Feedrate

Feedrate is designated by an F code. The axes are controlled so that vector sum (tangential velocity in reference to the tool moving direction) of feedrate of the designated axes will be the specified feedrate.

$$F\ (mm/min) = \sqrt{Fx^2 + Fy^2 + Fz^2 + (F\alpha^2 + F\beta^2)}$$

(Fx: feedrate in the X-axis direction)

Note

If no F code is designated in the block containing G01 or in the preceding blocks, execution of a G01 block causes an alarm.

If the optional 4th and 5th axis are rotary axes (A-, B-, or C-axis), feedrates of basic three axes (X-, Y-, and Z-axis) and the optional 4th and 5th axis are determined in the machine data (MD).

End point

The end point can be specified in either incremental or absolute values. In G code system B and C it is determind corresponding to the designation of G90 or G91 (for details, see 3.2.1, “Absolute/Incremental Programming”).

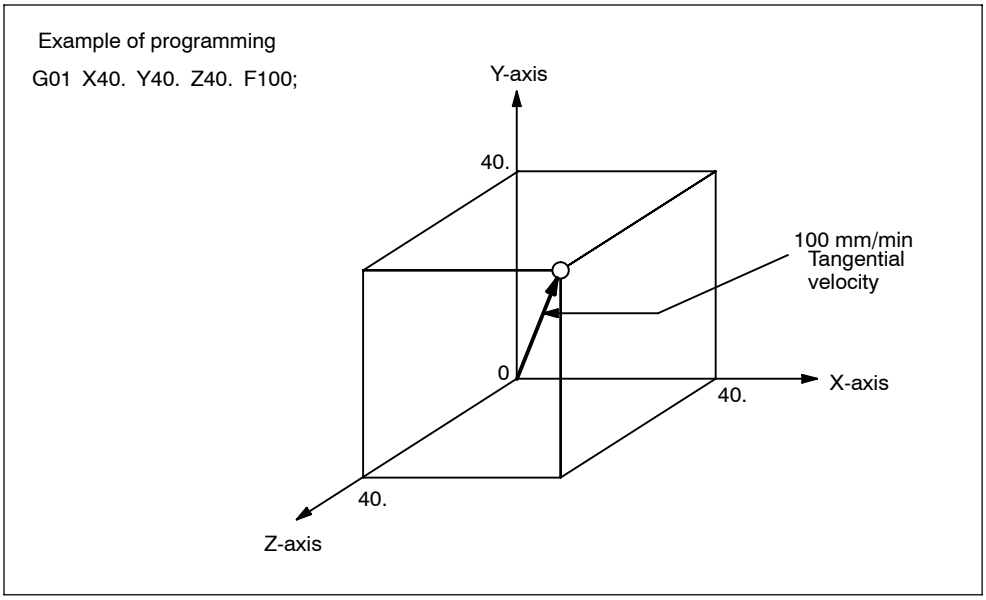


Fig. 2-2 Linear interpolation

2.1.3 Circular interpolation (G02, G03)

Command format

To execute the circular interpolation, the commands indicated in Table 2-2 must be specified.

Table 2-2 Commands necessary for circular interpolation

Item	Command	Description
Plane designation	G17	Circular arc in the XY plane
	G18	Circular arc in the ZX plane
	G19	Circular arc in the YZ plane
Direction of rotation	G02	Clockwise (CW)
	G03	Counterclockwise (CCW)
Position of end point	Two axes among X, Y, and Z	End point position in a workpiece coordinate system
	Two axes among X, Y, and Z	Signed distance from the start point to the end point
Distance from the start point to the center	Two axes among I, J, and K	Signed distance from the start point to the center
Radius of circular arc	R	Radius of circular arc
Feedrate	F	Velocity along the circular arc

Plane designation

With the commands indicated below, a cutting tool moves along the specified circular arc in the XY plane, ZX plane, or YZ plane so that the feedrate specified by the F command will be the tangential velocity of the arc.

- In the XY Plane
  G17 G02 (or G03) X···Y···R··· (or I···J···) F···;
- In the ZX Plane
  G18 G02 (or G03) Z···X···R··· (or K···I···) F···;
- In the YZ Plane
  G19 G02 (or G03) Y···Z···R··· (or J···K···) F···;

To designate the circular interpolation mode (G02, G03), the plane of interpolation should be selected first by specifying the G17, G18, or G19. For the 4th and 5th axis, circular interpolation is allowed only when they are linear axes.

The G code designated to select the plane in which circular interpolation is executed also selects the plane where tool radius offset (G41/G42) is executed. When the power is turned ON, the XY plane (G17) is automatically selected.

G17	XY plane, or Xα or Xβ plane
G18	ZX plane, or Zα or Zβ plane
G19	YZ plane, or Yα or Yβ plane

If an optional linear 4th-axis is selected, circular interpolation is possible in the Xα, Zα, or Yα plane which includes the 4th-axis in addition to the XY, YZ, and ZX planes. (α=U, V, or W)

- Circular interpolation in Xα plane
  G17 G02 (or G03) X ··· α ··· R ··· (or I ··· J ···) F ···;
- Circular interpolation in Zα plane
  G18 G02 (or G03) Z ··· α ··· R ··· (or K ··· I ···) F ···;
- Circular interpolation in Yα plane
  G19 G02 (or G03) Y ··· α ··· R ··· (or J ··· K ···) F ···;

If an optional linear 5th-axis is selected, circular interpolation is possible in the Xβ, Zβ, or Yβ plane which includes the 5th-axis in addition to the XY, YZ, and ZX planes. (β=U, V, or W)

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

Rotation direction

The direction of arc rotation should be specified in the manner indicated in Fig. 2-3.

G02	Clockwise direction (CW)
G03	Counterclockwise direction (CCW)

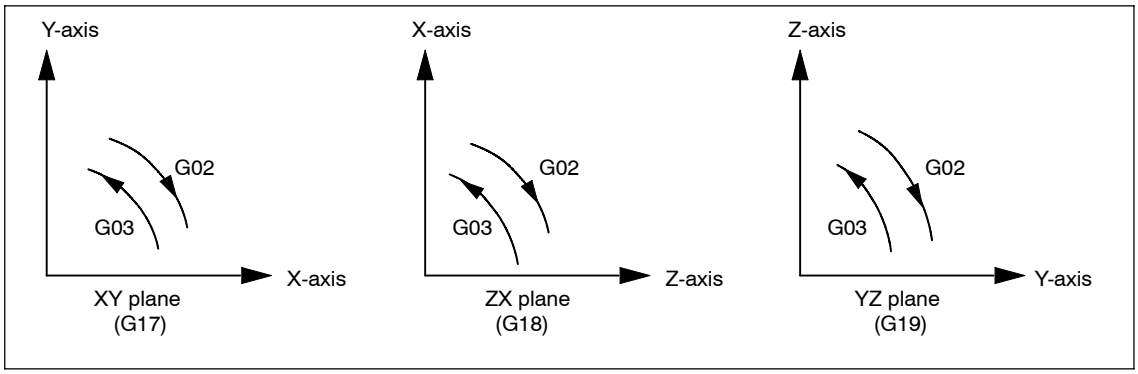
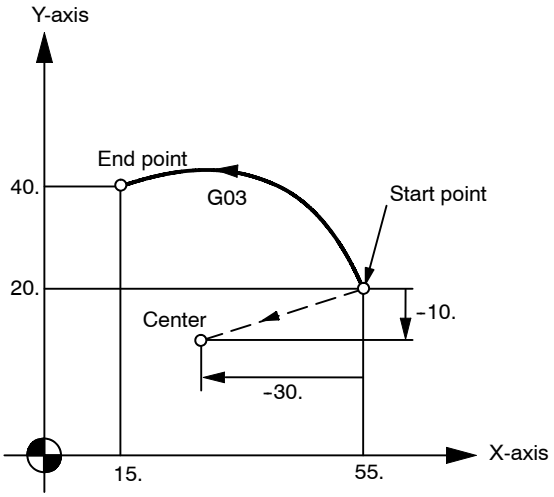


Fig. 2-3 Rotation direction of circular arc

End point

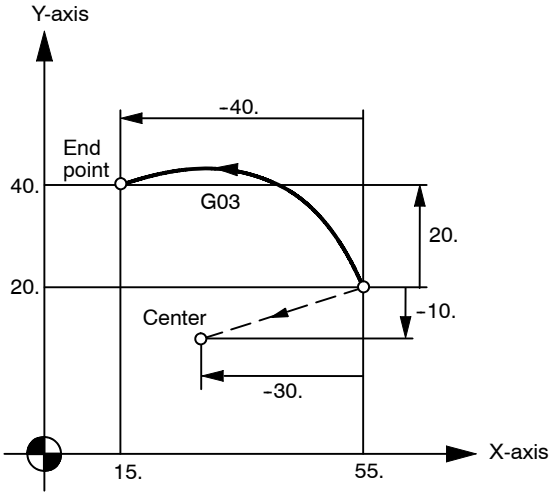
The end point can be specified in either absolute or incremental values corresponding to the designation of G90 or G91 (not in G code system A).

Example of programming
G17 G90 G03 X15. Y40. I-30. J-10. F150;



(a) Absolute programming (G90)

Example of programming
G17 G91 G03 X-40. Y20. I-30. J-10. F150;



(b) Incremental programming (G91)

Fig. 2-4 End point of circular arc

If the specified end point is not on the specified arc, the arc radius is gradually changed from the start point to the end point to generate a spiral so that the end point lies on the specified arc.

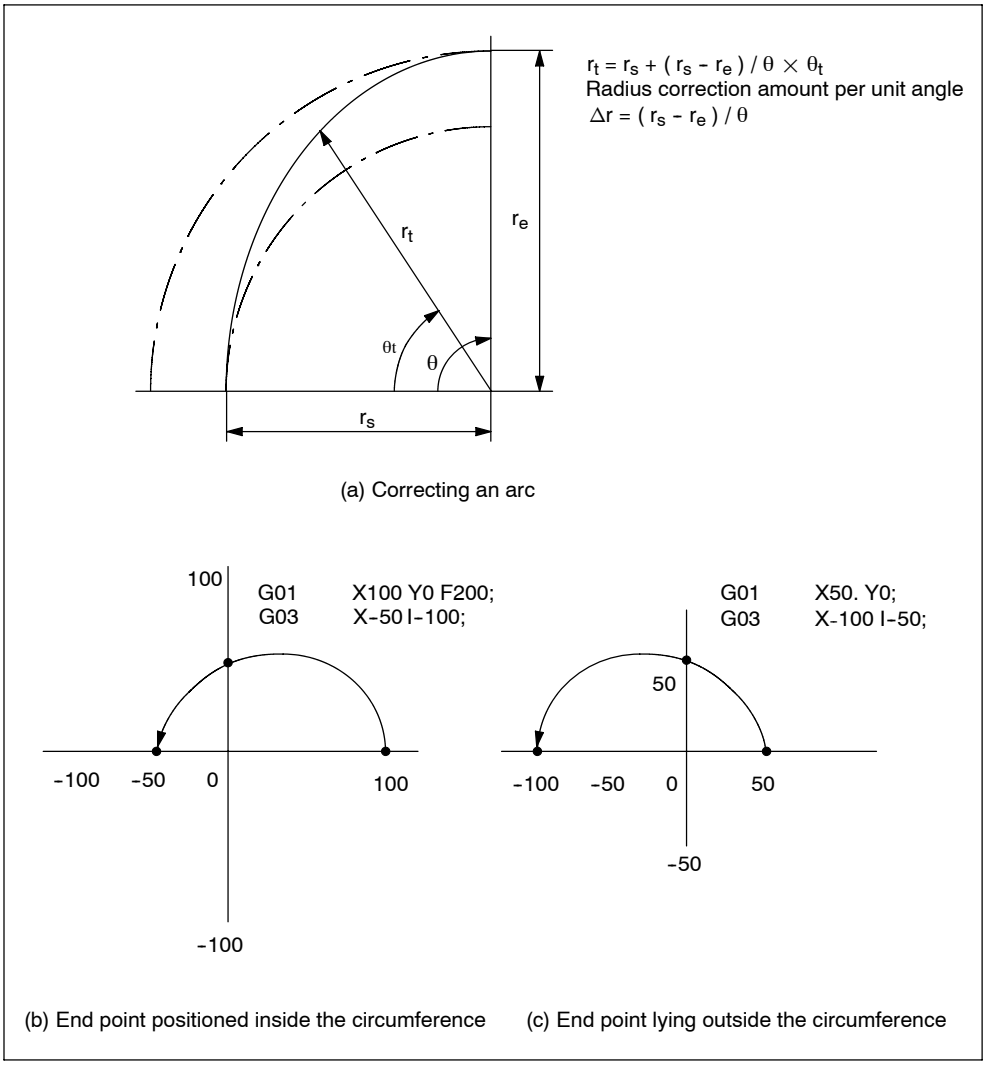


Fig. 2-5 Interpolation with end point of the specified arc

Center of arc

The center of arc can be specified in two methods – designation of the distance from the start point to the center of the arc and designation of the radius of the arc.

- Specifying the distance from the start point to the center
  Independent of the designated dimensioning mode (G90 or G91), the center of an arc must be specified in incremental values referenced from the start point.
- Specifying the radius
  When defining an arc, it is possible to specify the radius by using address R instead of specifying the center of the arc by addresses I, J, or K. This is called “circular interpolation with R designation” mode.
- For the circular arc with the central angle of 180 deg. or smaller, use an R value of “R > 0”.
- For the circular arc with the central angle of 180 deg. or larger, use an R value of “R < 0”.

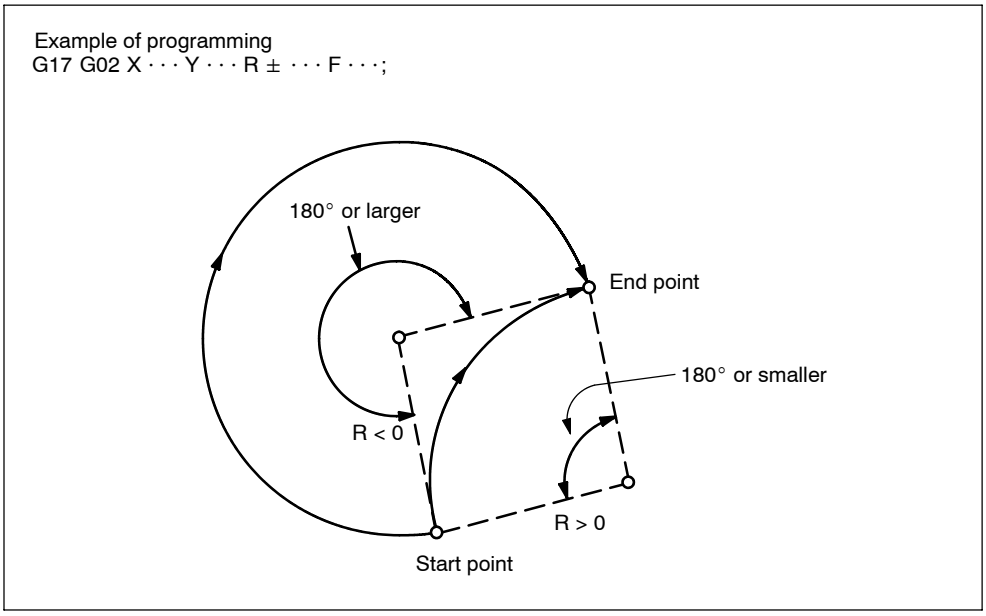


Fig. 2-6 Circular interpolation with radius R designation

Feedrate

In the circular interpolation mode, the feedrate can be specified in the same manner as in the linear interpolation mode. Refer to 2.1.2 “Linear interpolation (G01)”.

Supplements to circular interpolation

A circular arc extending to multiple quadrants can be defined by the commands in a single block. It is also possible to specify a full circle.

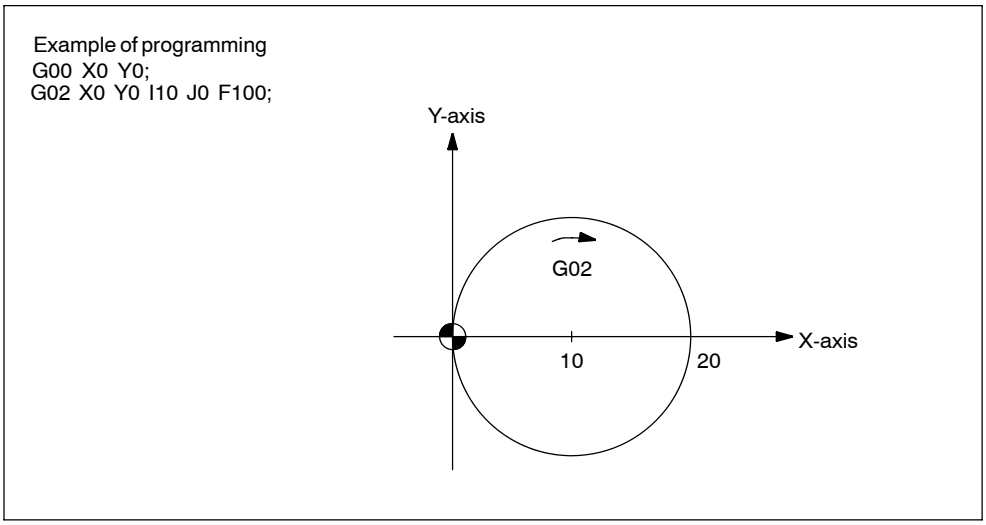


Fig. 2-7 Full circle

With the commands of “G17 G02 (or G03) I ··· J ··· F ··· Ln;”, full-circle interpolation is repeated by n times. If address L is omitted, interpolation is executed once. Execution of the commands with the single-block function ON causes full-circle interpolation to be interrupted after the execution of one full-circle interpolation.

2.1.4 Helical interpolation (G02, G03)

It is possible to execute linear interpolation in synchronization with circular interpolation with the axis which is not included in the circular interpolation plane. This is called helical interpolation. The command format is indicated below.

- In the XY plane
  G17 G02 (or G03) X ·· Y ·· R ·· (or I ·· J ··) Z (α, β) ·· F ··;
- In the ZX plane
  G18 G02 (or G03) Z ·· X ·· R ·· (or K ·· I ··) Y (α, β) ·· F ··;
- In the YZ plane
  G19 G02 (or G03) Y ·· Z ·· R ·· (or J ·· K ··) X (α, β) ·· F ··;
- In the Xα plane
  G17 G02 (or G03) X ·· α ·· R ·· (or I ·· J ··) Z (β) ·· F ··;
- In the Zα plane
  G18 G02 (or G03) Z ·· α ·· R ·· (or K ·· I ··) Y (β) ·· F ··;
- In the Yα plane
  G19 G02 (or G03) Y ·· α ·· R ·· (or J ·· K ··) X (β) ·· F ··;

- In the Xβ plane
  G17 G02 (or G03) X · · β · · R · · (or I · · J · ·) Z (α) · · F · ·;
- In the Zβ plane
  G18 G02 (or G03) Z · · β · · R · · (or K · · I · ·) Y (α) · · F · ·;
- In the Yβ plane
  G19 G02 (or G03) Y · · β · · R · · (or J · · K · ·) X (α) · · F · ·;

Where, α and β are the linear 4th and 5th axes respectively, each representing any of U-, V-, and W-axis. If no 4th or 5th axis is specified as the end point command of the arc, any of the command format is selected among the commands in the XY plane, ZX plane, and YZ plane.

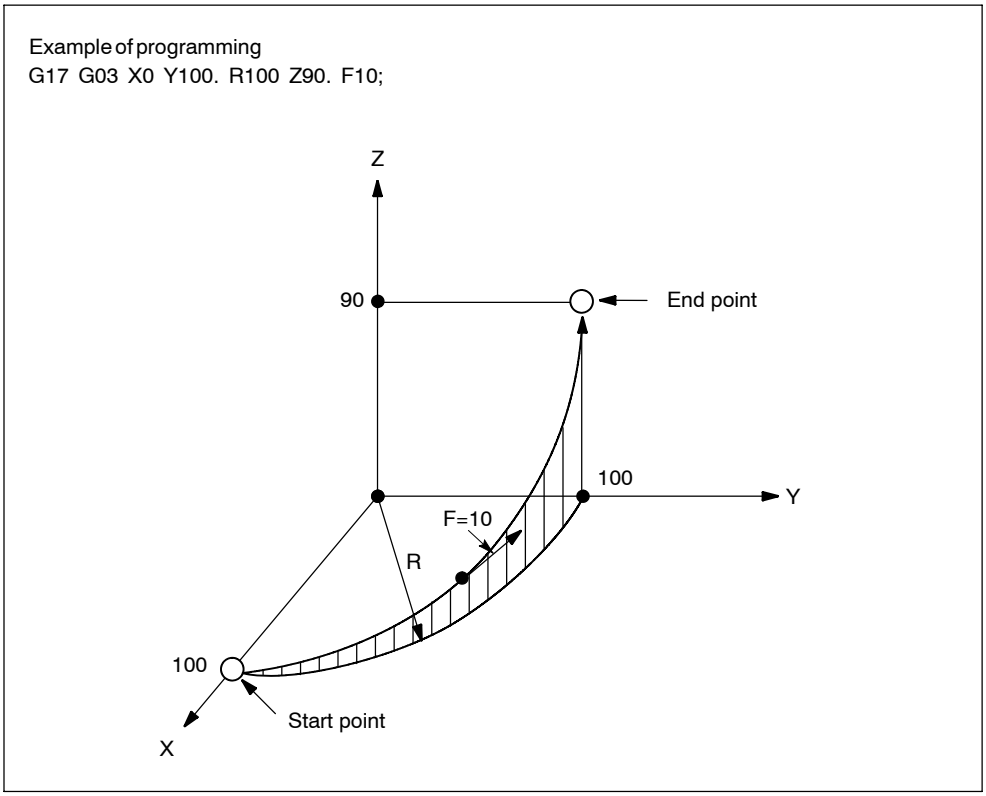


Fig. 2-8 Helical interpolation

Note

An arc must be programmed within 360° range.

The feedrate specified with an F command indicates the tangential velocity in the three dimensional space constituted by the circular interpolation plane and the linear axis perpendicular to the interpolation plane.

2.2 Reference point return

2.2.1 Automatic return to reference point (G28)

Format

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

With the commands of “G28 X · · · Y · · · Z · · · ;”, the numerically controlled axes are returned to the reference point. The axes are first moved to the specified position at a rapid traverse rate and then to the reference point automatically. This reference point return operation is possible in up to simultaneous 3-axis control. The axes not designated in the G28 block are not returned to the reference point.

Reference position

The reference position refers to a fixed position The position of the tool can easily be referenced by means of the reference position return function. This could, for instance, be used as the tool change position. A total of four reference positions can be determined by setting the coordinates using MD $_MA_REFP_SET_POS[0] to [3]).

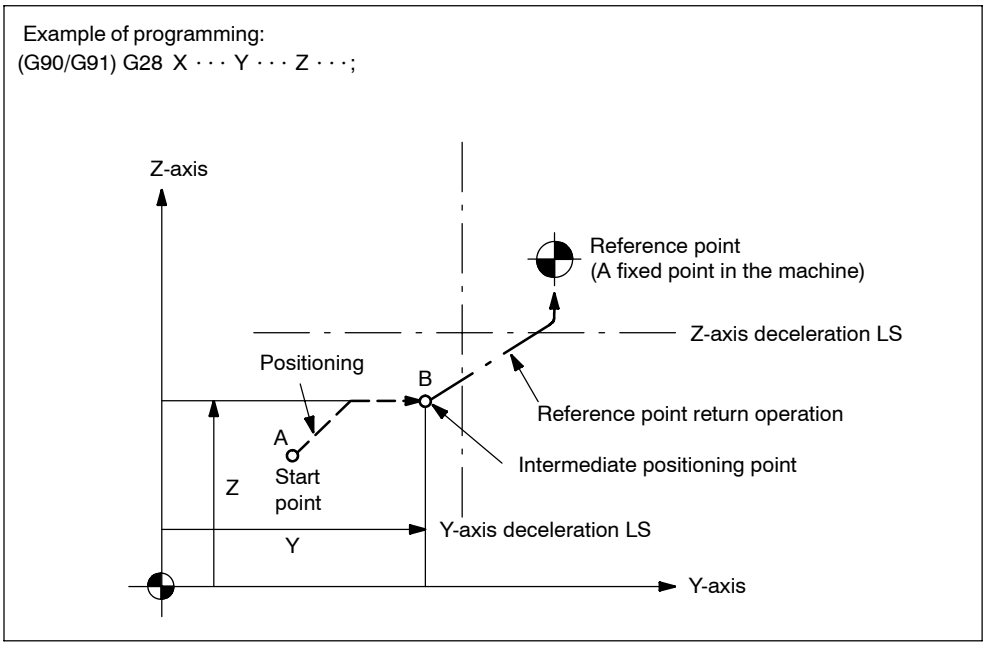


Fig. 2-9 Automatic reference point return

Reference point return operation

Reference point return operation is the series of operations in which the axes return to the reference point after the reference point return operation has been started manually.

Reference point return is executed in the following manner:

- After the positioning at the intermediate positioning point B, the axes return directly to the reference point at a rapid traverse rate. The axes can be returned to the reference point in a shorter time compared to the normal reference point return operation that uses a deceleration limit switch for the individual axes.
- Even if point B is located outside the area in which reference point return is allowed, the high-speed reference point return specification allows the axes to return to the reference point.
- High-speed automatic reference point return is valid only when reference point return is called by G28, and it does not influence manual reference point return operation.

Automatic reference point return for rotary axes

With a rotary axis, it is possible to execute the automatic reference point return the same as with a linear axis. With a rotary axis, if it has been moved by more than ±360.000° from the reference point established first, reference point return is executed to the closest reference point in the preset direction of reference point return. The illustration below shows how the reference point return is executed from points A and B. (The reference point return direction is determined by the setting of MD_$MA_REFP_CAM_IS_MINUS.

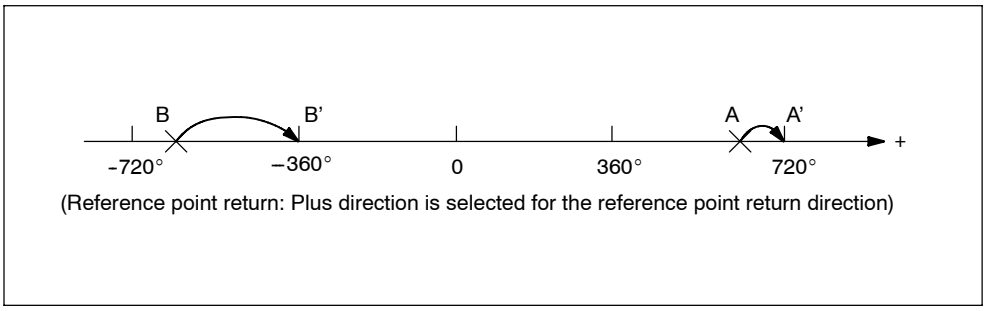


Fig. 2-10

Supplements to the automatic reference point return commands

Tool radius offset and canned cycle

G28 must not be specified in the tool radius offset mode (G41, G42) or in a canned cycle.

Warning

Issuing G28 will cancel tool radius offset (G40) followed by axes movement towards the reference point. For that reason, make sure to disable tool radius offset before issuing G28.

Tool position offset

If G28 is specified in the tool position offset mode, positioning at the intermediate positioning point is made with the offset data valid. However, for the positioning at the reference point, the offset data are invalid and positioning is made at the absolute reference point.

Tool length offset

It is possible to cancel the tool length offset mode by G28 by changing the setting for a parameter. Although cancellation of the tool length offset mode is possible by G28, the tool length offset mode should be canceled before the designation of G28.

Machine lock intervention

The lamp for indicating the completion of return does not go on when the machine lock is turned on, even when the tool has automatically returned to the reference position. In this case, it is not checked whether the tool has returned to the reference position even when a G27 command is specified.

2.2.2 Reference point return check (G27)

Format

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

This function checks whether the axes are correctly returned to the reference point at the completion of the part program which is created so that the program starts and ends at the reference point in the machine by specifying the commands of "G27 X···Y···Z···;".

In the G27 mode, the function checks whether or not the axes positioned by the execution of these commands in the simultaneous 3-axis (* 5-axis) control mode are located at the reference point. For the axes not specified in this block, and not moved although the axis command specified, positioning and check are not executed.

Operation after the check

When the position reached after the execution of the commands in the G27 block agrees with the reference point, the reference point return complete lamp lights. The automatic operation is continuously executed when all of the specified axes are positioned at the reference point. If there is an axis that has not been returned to the reference point, reference point return check error occurs and the automatic operation is interrupted.

Supplements to the reference point return check command and other operations

- If G27 is specified in the tool offset mode, positioning is made at the position displaced by the offset amount and the positioning point does not agree with the reference point. It is necessary to cancel the tool offset mode before specifying G27. Note that the tool position offset and tool length offset functions are not canceled by the G27 command.
- Check is not made if G27 is executed while the machine lock state is valid even for one axis. For example, if an X-axis movement command is specified in the G27 block while in the Z-axis neglect state, X-axis position is not checked.
- The mirror image function is valid to the direction of axis movement in the reference point return operation called by G27. To avoid a position unmatch error, the mirror image function should be canceled before executing G27.

2.2.3 Second to fourth reference point return (G30)

Format

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

With the commands of “G30 Pn X · · · Y · · · Z;”, the axes are moved to P2 (second reference point), P3 (third reference point*), or P4 (fourth reference point*) in the simultaneous 3-axis (* 5-axis) control mode after the positioning at the specified intermediate positioning point. If “G30 P3 X30. Y50.;” is specified, the X- and Y-axis return to the third reference point. If “Pn” is omitted, the second reference point is selected. The axes not specified in the G30 block do not move.

Reference point positions

The position of each reference point is determined in reference to the first reference point. The distance from the first reference point to each of the reference points is set for the following machine data:

Table 2-3 Reference points

Item	MD
3rd reference point	$_MA_REFP_SET_POS[2]
4th reference point	$_MA_REFP_SET_POS[3]

Supplements to the 2nd to 4th reference point return commands

- For the points to be considered to for the execution of G30, refer to the supplements in 2.2.1, “Automatic Return to Reference Point (G28)”.
- For the execution of G30, reference point return must have been completed after power-ON either manually or by the execution of G28. If an axis for which reference point return has not been completed is included in the axes specified in the G30 block, an alarm occurs.

2.2.4 Rapid lift with G10.6

G10.6 <AxisPosition> is used to activate a retraction position for the rapid lifting of a tool (e.g., in the event of a tool break). The retraction motion itself is started with a digital signal. The second NC fast input is used as the start signal.
Machine data $MN_EXTERN_INTERRUPT_NUM_RETRAC is used to select a different fast input (1 – 8).

In Siemens mode, the activation of the retraction motion comprises a number of part program commands.

```
N10 G10.6 X19.5 Y33.3
```

generates internally in the NCK

```
N10 SETINT (2) PRIO=1 CYCLE3106 LIFTFAST  ; Activate interrupt input
N30 LFPOS                                 ; Select lift mode
N40 POLF[X]=19.5  POLF[Y]=33.3            ; Program lift positions
                                          ; for x19.5 and y33.3
N70 POLFMASK(X, Y)                        ; Activate retraction
                                          ; of x and y axis
```

G10.6 is used to group these part program commands internally in a command set.

In order to activate an interrupt input (SETINT(2)), an interrupt program (ASUP) must also be defined. If one has not been programmed, the part program will not be able to continue as it will be interrupted with a reset alarm once the retraction motion is complete. The interrupt program (ASUP) CYCLE3106.spf is always used for fast retraction with G10.6. If the part program memory does not contain program CYCLE3106.spf, alarm 14011 “Program CYCLE3106 not available or not enabled for processing” is output in a part program set with G10.6.

The behavior of the control following fast retraction is specified in ASUP CYCLE3106.spf. If the axes and spindle are to be stopped following fast retraction, M0 and M5 must be programmed accordingly in CYCLE3106.spf.
If CYCLE3106.spf is a dummy program, which only contains M17, the part program will continue uninterrupted following fast retraction.

If G10.6 <AxisPosition> is programmed to activate fast retraction, when the input signal of the second NC fast input changes from 0 to 1, the motion currently in progress is interrupted and the position programmed in set G10.6 is approached at rapid traverse. The positions are approached absolutely or incrementally according to the program settings in set G10.6.

The function is deactivated with G10.6 (without positional data). Fast retraction by means of the input signal of the second NC fast input is disabled.

Siemens

To some extent, the fast retraction function with G10.6 can be achieved using function POLF[<AxisName>] = <RetractionPosition>. This function will also retract the tool to the programmed position. However, it does not support the remainder of the ISO dialect original functionality. If the interrupt point cannot be approached directly, obstructions must be bypassed manually.

References: /PGA/, Programming Guide Advanced,
Chapter “Extended Stop and Retract”

Restrictions

Only one axis can be programmed for fast retraction.

Notes

Movement Control Commands

3

3.1 The coordinate system

A tool position is clearly determined by coordinates within a coordinate system. These coordinates are defined by program axes. For example, if there are 3 program axes involved designated as X, Y, and Z, the coordinates are specified as:

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

The above command is called a dimension word.

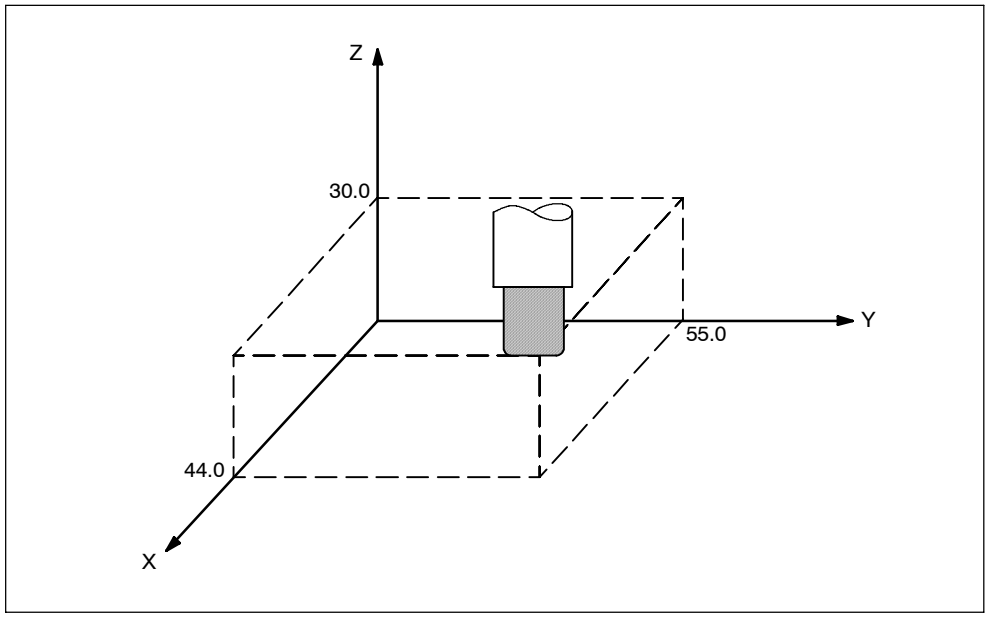


Fig. 3-1 Tool position specified by X... Y... Z...

The following three coordinate systems are used to determine the coordinates:

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

3.1.1 Machine coordinate system (G53)

The machine zero point represents the point that is specific to a machine and serves as the reference of the machine. A machine zero point is set by the MTB for each machine tool. A machine coordinate system consists of a coordinate system with a machine zero point at its origin.

A coordinate system with a machine zero point set at its origin is referred to as a machine coordinate system. By using manual reference position return after power-on the machine coordinate system is set. Once set, the machine coordinate system remains unchanged until power–off.

Format

(G90) G53 X... Y... Z... ;
X, Y, Z, Absolute dimension word

How to select a machine coordinate system (G53)

Once a position has been determined in terms of machine coordinates, the tool moves to that position in rapid traverse. G53 is a one–shot G code. Thus, any command based on the selected machine coordinate system is effective only in the block where G53 is issued. The G53 command has to be determined by using absolute values. Program the movement in a machine coordinate system based on G53 whenever the tool should be moved to a machine–specific position.

Cancel of the compensation function

If $MN_G53_TOOLCORR = 0, G53/G153/SUPA is non–modal suppression of zero offsets, tool length compensation and tool radius compensation, however, remain active.
If $MN_G53_TOOLCORR = 1, G53/G153/SUPA is non–modal suppression of zero offsets, and active tool length and tool radius compensation.

G53 specification right after power–on

At least one manual reference position return must be applied after power–on, since the machine coordinate system must be set before the G53 command is determined.
If an absolute position detector is attached, this is not required.

Reference

A machine coordinate system is set so that the reference position is at the coordinate values set using MD $MC_CHBFRAME_POWON_MASK Bit 0 whenever manual reference position return is applied after power-on.

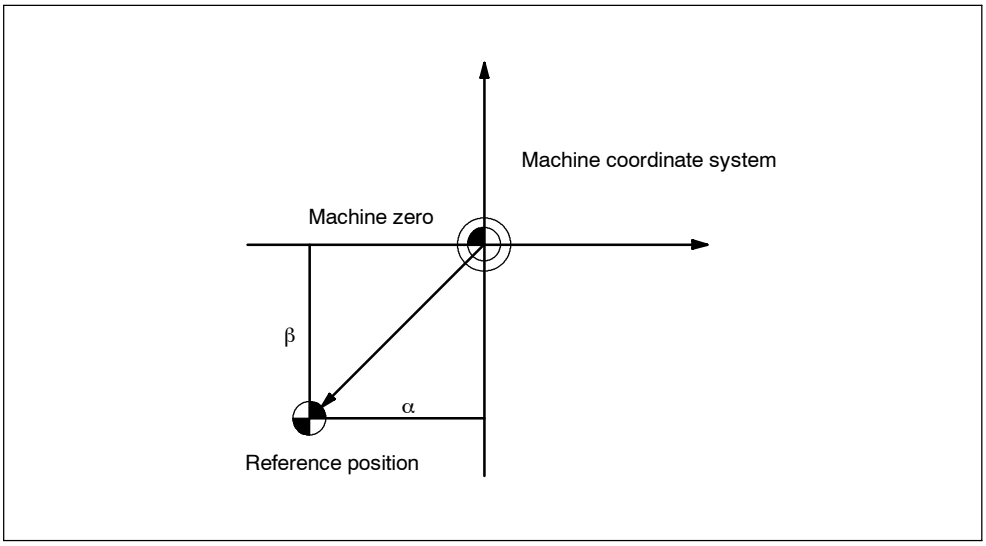


Fig. 3-2 Reference

3.1.2 Workpiece coordinate system (G92)

Prior to machining, a coordinate system for the workpiece, the so-called workpiece coordinate system, needs to be established. This section describes the various methods of how to set, select, and change a workpiece coordinate system.

How to set a workpiece coordinate system

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

1. Using G92

   A workpiece coordinate system is set by determining a value subsequent to G92 within the program.

2. Manually, using the HMI panel

Format

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

Examples

Example 1:
G92X30.5Z27.0;
(The tip of tool is the start point.)

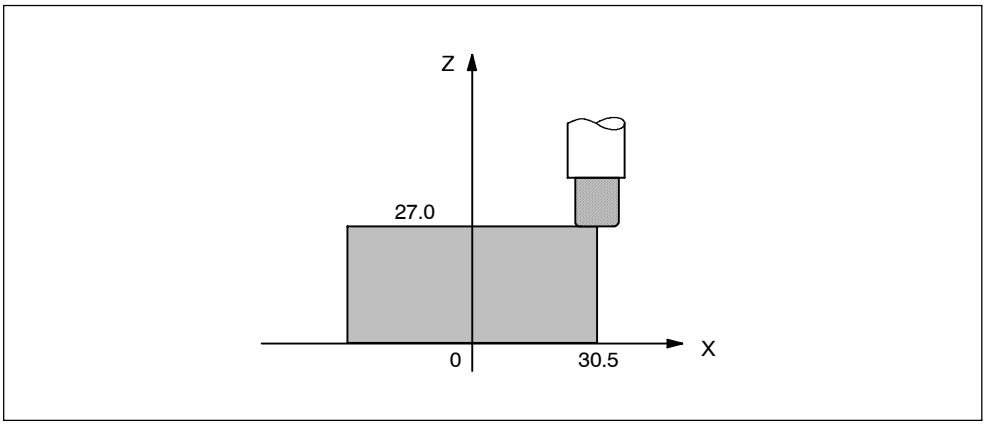


Fig. 3-3 Example 1

Example 2:
G92X500.0Z1100.0;
(The base point on the tool holder is the start point.)

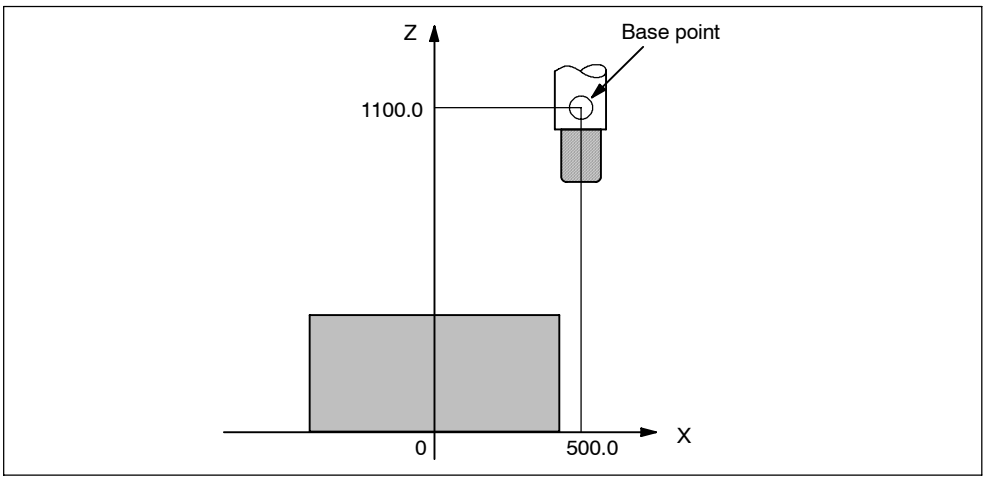


Fig. 3-4 Example 2

Whenever an absolute command is issued, the base point moves to the targeted position. The difference in position between the tool tip and the base point is compensated by the tool length offset in order to move the tool tip to the targeted position.

3.1.3 Resetting the work (G92.1)

With G92.1 X.., you can reset an offset coordinate system before shifting it. This resets the work to the coordinate system which is defined by the actively settable work offsets (G54–G59). If not settable work offset is active, the work is set to the reference position. G92.1 resets offsets which have been performed by G92 or G52. Only axes which 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 How to select a workpiece coordinate system

As described below, the user may choose from set workpiece coordinate systems.

1. G92

   Absolute commands work with the workpiece coordinate system once a workpiece coordinate system has been selected.

2. Selecting from workpiece coordinate systems previously set up by using the HMI.

   A workpiece coordinate systems can be selected by determining a G code from G54 to G59, and G54 P{1...100}.
   Workpiece coordinate systems are set up subsequent to reference position return after power–on. The default coordinate system after power–on is G54.

Examples

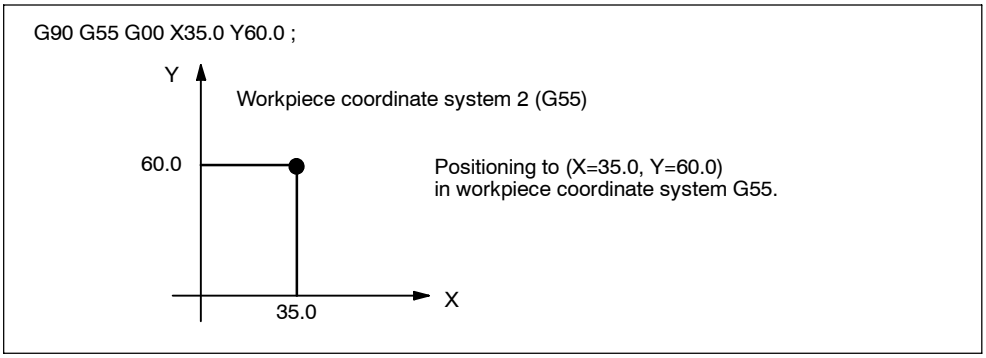


Fig. 3-5 Workpiece coordinate system G55

3.1.5 Instantaneous mapping of the ISO functions onto Siemens frames

(until powerline 7.04.2, solution line 1.4)

By changing an external workpiece zero point offset value or workpiece zero point offset value, the workpiece coordinate systems determined through G54 to G59 as well as G54 P{1 ... 93} are changed.

In order to change an external workpiece zero point offset value or workpiece zero point offset value, two methods are available.

1. Entering data using the HMI panel
2. By program command G10 or G92

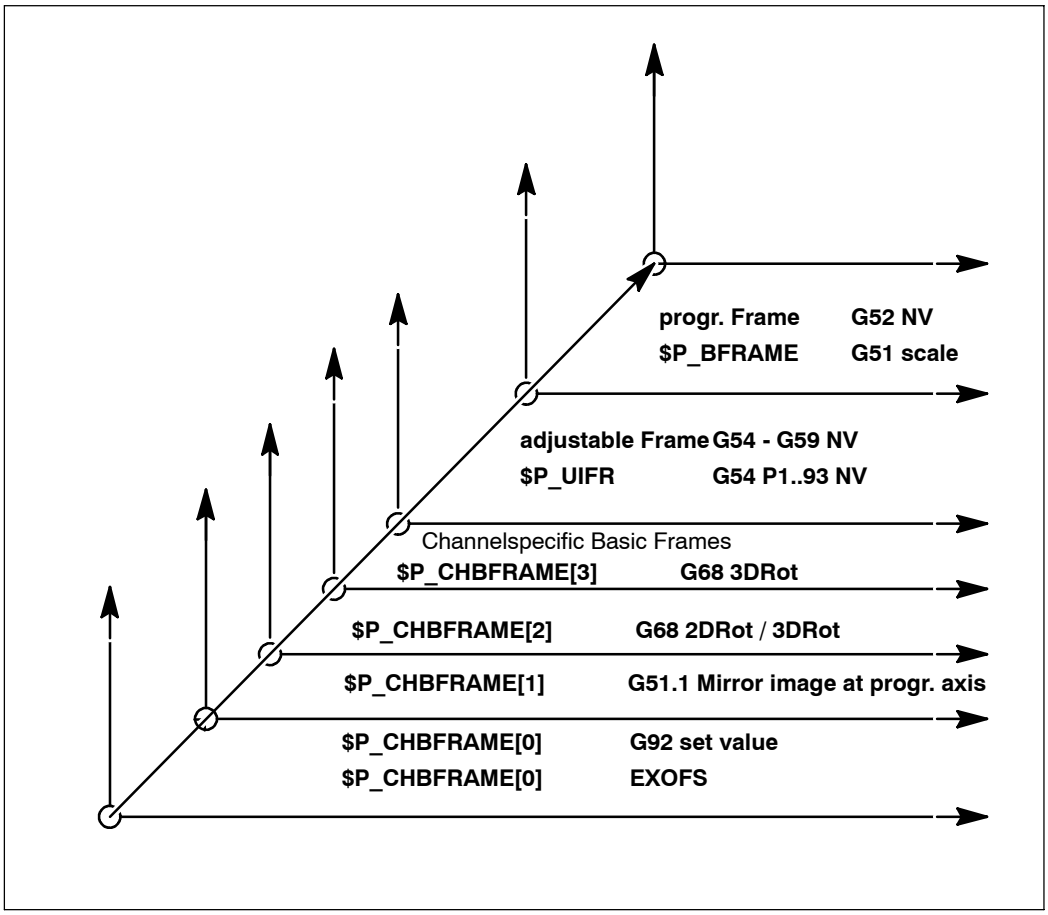


Fig. 3-6 ISO-dialect coordinate systems

G54P1...P93 (changes at Siemens Mode G505–G597)
G58 (changes at Siemens Mode G505)
G59 (changes at Siemens Mode G506)

Format

Changing by G10:

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

p=0:	External workpiece zero point offset value (EXOFS)
p=1 to 6:	Workpiece zero point offset value correspond to workpiece coordinate system G54 to G59
X, Y, Z:	Workpiece zero point offset for each axis in case of absolute command (G90). Value to be added to the set workpiece zero point offset for each axis in case of an incremental command (G91).

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

p=1 to 93: Workpiece zero point offset value correspond to workpiece coordinate system G54 P1 ... P93

X, Y, Z: For an absolute command (G90), workpiece zero point offset for each axis.
Value to be added to the set workpiece zero point offset for each axis in case of an incremental command (G91).

Changing by G92:

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

Explanations

Changing by using G10

Each workpiece coordinate system can be changed separately by using the G10 command.
If G10 is executed in the main run, G10 must execute an internal STOPRE command before writing the value.
In MD $MC_EXTERN_FUNCTION_MASK Bit 13, you can configure whether the G10 command shall execute an internal STOPRE. The machine data bit affects all G10 commands in ISO-Dialect-T and ISO-Dialect-M.

Changing by using G92

A workpiece coordinate system (selected with a code from G54 to G59 and G54 P{1 ...93}) is shifted to set a new workpiece coordinate system by specifying G92 X... Y... Z.... This way, the current tool position is made to match the specified coordinates. If X, Y, Z, is an incremental command value, the work coordinate system is defined so that the current tool position coincides with the result of adding the specified incremental value to the coordinates of the previous tool position (coordinate system shift). Subsequently, the value of the coordinate system shift is added to each individual workpiece zero point offset value. In other words, all of the workpiece coordinate systems are systematically shifted by the same value amount.

Example

When the tool is positioned at (190, 150) in G54 mode, workpiece coordinate system 1 (X' – Y') shifted by vector A is created whenever G92X90Y90; is commanded.

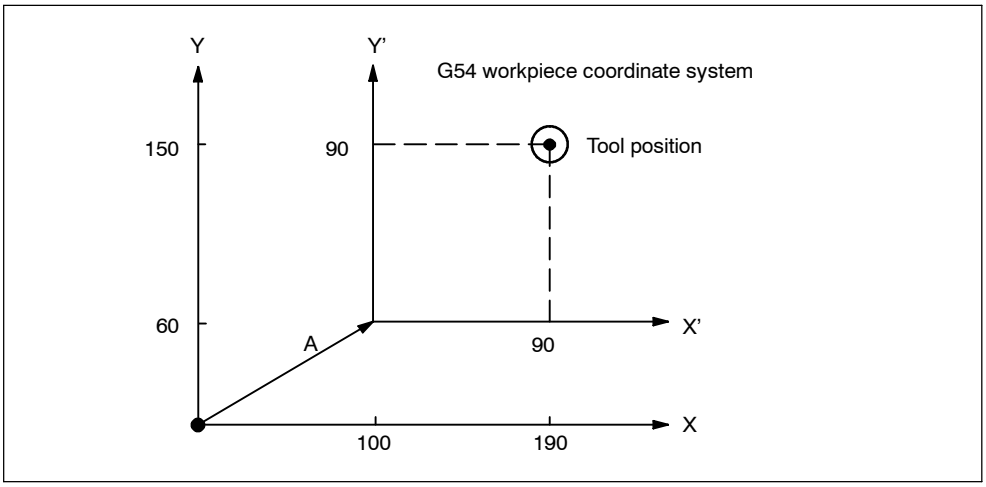


Fig. 3-7 Example of the setting of coordinates

3.1.6 Uncoupling the frames between the Siemens and the ISO modes

(with powerline 7.04.02 or solution line 1.4 and higher)

In the ISO mode, various G codes occupied the programmable frame $P_FRAME, the settable frame $P_UIFR and three base frame $P_CHBFRAME[ ]. If you switch from the ISO mode to the Siemens mode, these frames will not be available to the user of the Siemens language. This pertains to:

G52 Programmable zero offset -> progr. frame $P_PFRAME

G51 Scaling -> progr. frame $P_BFRAME SCALE

G54–G59 Zero offset -> settable frame $P_UIFR

G54 P1..100 Zero offset -> settable frame $P_UIFR

G68 3D rotation -> base frame $P_CHBFRAME[3]

G68 2D rotation -> base frame $P_CHBFRAME[2]

G51.1 Mirroring -> base frame $P_CHBFRAME[1]

G92 Set actual value-> base frame $P_CHBFRAME[0]S

G10 L2 P0 Ext. zero offset -> base frame $P_CHBFRAME[0]S

To uncouple the concerned frames between the Siemens and the ISO modes, four new system frames are provided: $P_ISO1FRAME to $P_ISO4FRAME. The frames are created with the machine data 28082: $MC_MM_SY-

STEM_FRAME_MASK, bits 7 to 10. The reset behavior is set using the machine data 24006: $MC_CHSFRAME_RESET_MASK, bits 7 to 10.

Fig. 3-8 shows the G codes in the ISO mode and the assignment of the frames if the system frames $P_ISO1FRAME to $P_ISO4FRAME, $P_SETFRAME and $P_EXTFRAME are created.

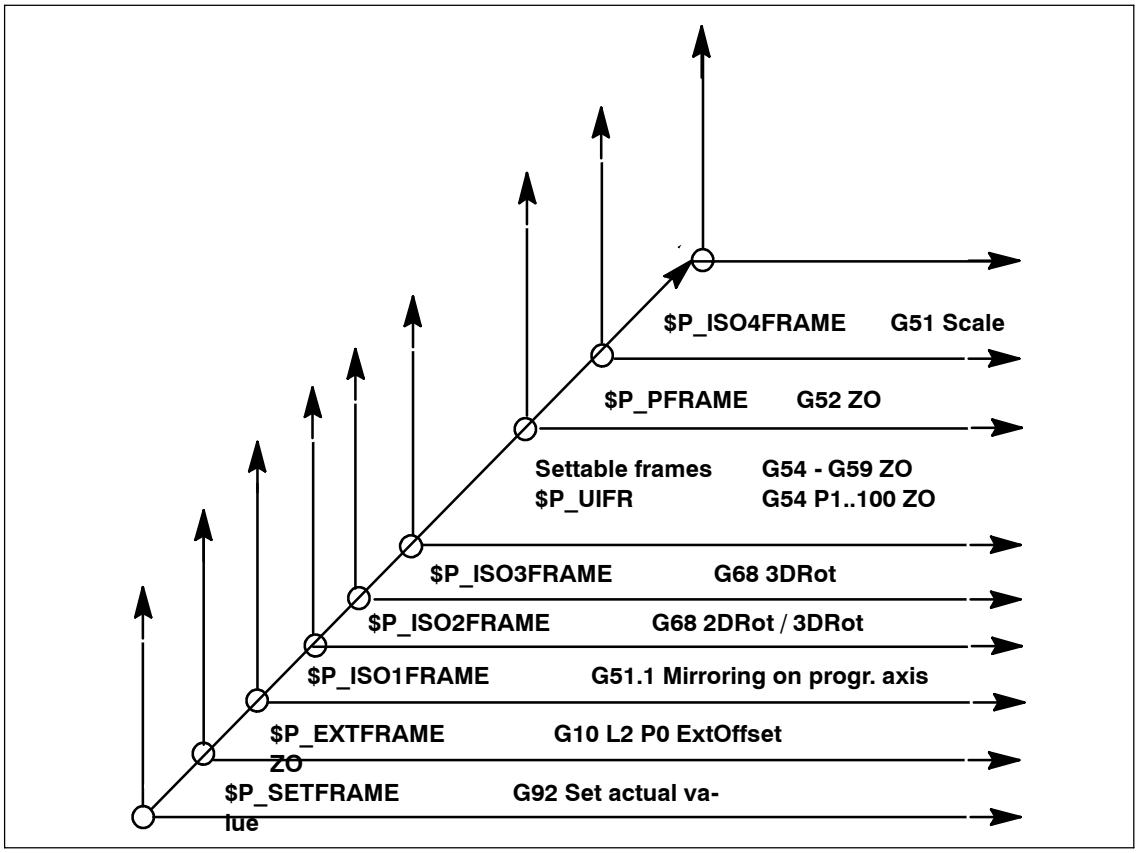


Fig. 3-8 Mapping of the ISO functions to the ISO frames and Siemens frames

Note

If the new frames are created, the ISO G codes will write to these frames; if they are not created, the frames are written as described in Section 3.1.5.

The tables on the following pages illustrate which G codes write to which frames, how they are created and how the reset behavior of the frames must be set to achieve a compatible behavior to the ISO mode original. The reset behavior can be set deviating from the ISO mode original using the MDs mentioned above. This can be necessary when switching from the ISO mode to the Siemens mode.

G51: Scaling

G51 X10 writes to	$P_ISO4FRAME
Component	TRANS, SCALE
Creates	$MC_MM_SYSTEM_FRAME_MASK Bit10 = 1
Reset behavior	Delete frame $MC_CHSFRAME_RESET_MASKBit 10 = 0

G52:Programmable zero offset

G52 X10 writes to	$P_PFRAME
Component	TRANS
Creates	Always present
Reset behavior	Is deleted in case of RESET

G54 – G59 P1...100: Settable zero offset

G52 – G59	$P_UIFER
Component	TRANS
Creates	Always present
Reset behavior	G54 is active after RESET $MC_EXTERN_GCODE_RESET_VALUES[13] = 1

G68 3DRot

G68 X Y I J K R	$P_ISO3FRAME
Component	TRANS, SCALE
Creates	$MC_MM_SYSTEM_FRAME_MASK Bit 9 = 1
Reset behavior	Delete frame $MC_CHSFRAME_RESET_MASKBit 9 = 0

G68 2DRot

G68 X Y R	$P_ISO2FRAME
Component	TRANS, SCALE

Creates	$MC_MM_SYSTEM_FRAME_MASK Bit 8 = 1
Reset behavior	Delete frame $MC_CHSFRAME_RESET_MASKBit 8 = 0

G51.1: Mirroring

G51.1 X Y	$P_ISO1FRAME
Component	TRANS, MIRROR
Creates	$MC_MM_SYSTEM_FRAME_MASK Bit 7 = 1
Reset behavior	Delete frame $MC_CHSFRAME_RESET_MASKBit 7 = 0

G92: Set actual value

G92 X Y R	$P_SETFRAME
Component	TRANS
Creates	$MC_MM_SYSTEM_FRAME_MASK Bit 0 = 1
Reset behavior	Frame is maintained after RESET $MC_CHSFRAME_RESET_MASKBit 0 = 1

G10 L2 P0

G10 L2 P0	$P_EXTFRAME
Component	TRANS
Creates	$MC_MM_SYSTEM_FRAME_MASK Bit 1 = 1
Reset behavior	Delete frame $MC_CHSFRAME_RESET_MASKBit 1 = 0

If all frames are created, it is no longer necessary for the ISO mode that the frames are configured using the FINE component. The machine data 18600: $MN_MM_FRAME_FINE_TRANS need not be set to ”1”. If you switch from the ISO mode to the Siemens mode and if the Siemens mode uses a function which requires a fine offset (e.g. G58, G59), $MN_MM_FRAME_FINE_TRANS must remain ”1”.

3.1.7 Local coordinate system (G52)

For easier programming, a kind of sub–workpiece coordinate system can be set whenever a program is created in a workpiece coordinate system. Such a sub–coordinate system is called a local coordinate system.

Format

G52 X... Y... Z... ; Local coordinate system set

G52 X0 Y0 Z0 ; Local coordinate system cancel

X, Y, Z: Local coordinate system origin

Explanations

A local coordinate system can be set in all the workpiece coordinate systems (G54 to G59) by specifying G52 X... Y... Z...;. Within the workpiece coordinate system, the origin of each local coordinate system is set to the position determined by X, Y, and Z.
Whenever a local coordinate system is set, the motion commands subsequently commanded in the absolute mode (G90) correspond to the coordinate values within the local coordinate system. By determining the G52 command through the zero point of a new local coordinate system in the workpiece coordinate system, the local coordinate system can be changed.
Match the zero point of the local coordinate system with that of the workpiece coordinate system in order to cancel the local coordinate system and to determine the coordinate value within the workpiece coordinate system.
The position value displayed as the coordinate value of workpiece coordinate sytem refers to the zero point of workpiece coordinate system even if the local coordinate system is set by specifying G52.

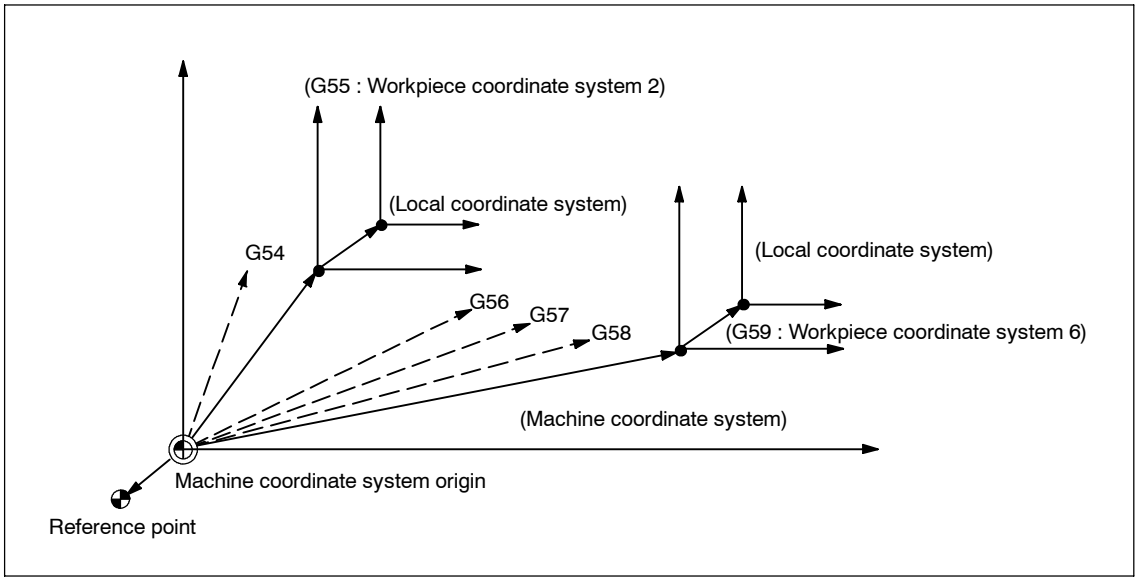


Fig. 3-9 Setting the local coordinate system

3.1.8 Plane selection (G17, G18, G19)

The plane where circular interpolation, tool radius offset, and coordinate system rotation are executed is selected by specifying the following G code.

Table 3-1 Plane selection G codes

G code	Function	Group
G17	XY plane	02
G18	ZX plane	02
G19	YZ plane	02

A plane is defined in the following manner (in the case of XY plane):

The horizontal axis in the first quadrant is “+X-axis” and the vertical axis in the same quadrant “+Y-axis”.

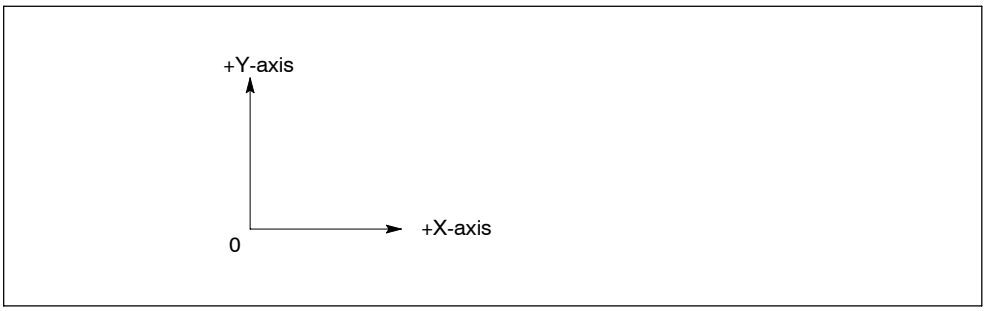


Fig. 3-10 Plane selection

- When the power is turned ON, the XY plane (G17) is selected.
- Axis move command of a single axis can be specified independent of the selection of plane by G17, G18, and G19. For example, the Z-axis can be moved by specifying “G17 Z ....;”.
- Execution of a canned cycle is possible only in the G17 plane (hole machining axis: Z-axis).
- The plane on which the tool radius offset is executed by the G41 or G42 command is determined by the designation of G17, G18 or G19; the plane that includes the rotary 4th- or 5th-axis cannot be selected as the offset plane.

3.1.9 Parallel axes (G17, G18, G19)

Using the function G17 (G18, G19) <axis name>, an axis parallel to one of the three basic axes of the coordinate system can be activated.
The three basic axes are, for example, X, Y, and Z.

Example

G17 U0 Y0

Parallel axis U is activated, replacing the X axis within the G17 plane.

Explanations

- The parallel axes command is emulated using the Siemens function GEOAX(..,..). With the help of this function, a geometrical axis can be exchanged by any available channel axis.
- For each of the geometrical axes, a related parallel axis can be determined using machine data
  $MC_EX-TERN_PARALLEL_GEOAX[].
- Only axes related to the programmed plane (G17, G18, G19) can be exchanged.
- Usually, when exchanging axes, all offsets (frames) except for handwheel and external offsets, work area limitation and protection zones are cleared. Be sure to set the following machine data to prevent from clearing such values:

  Offsets (frames)
  $MN_FRAME_GEOAX_CHANGE_MODE

  Protection zones
  $MC_PROTA-REA_GEOAX_CHANGE_MODE

  Work area limitation
  $MN_WALIM_GEOAX_CHANGE_MODE
- Refer to machine data description for detail.
- Alarm 12726 is issued, if a basic axis is programmed together with its parallel axis in a plane selection command.

3.1.10 Rotation of coordinate system (G68, G69)

Does not work with SINUMERIK 802D sl.

Using the G68 and G69 commands

Features of G68 and G69

For the rotation of a coordinate system, the following G codes are used.

Table 3-2 Coordinate system rotation G codes

G code	Function	Group
G68	Coordinate system rotation	16
G69	Cancel of coordinate system rotation	16

G68 and G69 are modal G codes belonging to 16-group. When the power is turned ON and when the NC is reset, G69 is automatically selected.

The G68 and G69 blocks must not include other G codes.
The coordinate system rotation which is called by G68 must be canceled by G69.

Command format

G68 X_ Y_ R_ ;

X_, Y_ :
Absolute coordinate values of the center of rotation. If omitted, the actual position is regarded as center of rotation.
R_ :
Rotation angle, absolute or incremental depending on G90/G91. If omitted, the value of the channel specific setting $SC_DEFAULT_ROT_FACTOR_R is used as rotation angle.

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

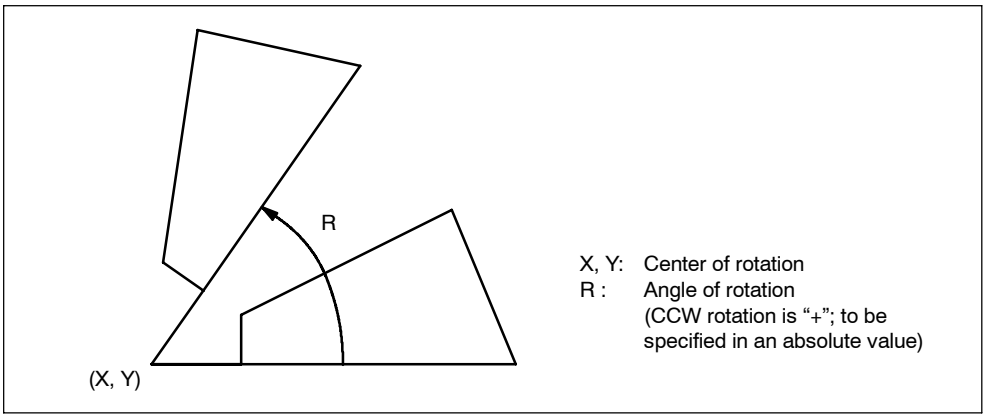


Fig. 3-11 Rotation of coordinate system

- By specifying "G69;", the coordinate system rotation mode is canceled.
- The G68 command is executed in the plane that has been selected when the G68 command is specified. The 4th and 5th axis must be linear axes.

G17	:	XY plane or Xα, Xβ plane
G18	:	ZX plane or Zα, Zβ plane
G19		YZ plane or Yα, Yβ plane

Supplements to the coordinate system rotation commands

- MD $MC_MM_NUM_BASE_FRAMES must be set to a value >= 3 if coordinate system rotation is used.
- If "X" and "Y" are omitted, the present position when the G68 block is executed is taken as the center of rotation.
- When the coordinate system is rotated, position data are given in the rotated coordinate system.
- Usually, the coordinate system rotation is turned ON before the start of approach motion and turned OFF after the completion of machining. The workpiece cannot be machined correctly if it is turned ON during machining.

Note

For incoupling the frames between the Siemens and the ISO modes (solution line) see section 3.1.6.

3.1.11 3D rotation G68 / G69

G code G68 has been expanded for 3D rotation.

Format

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

X.. Y.. Z..:	Coordinates of the pivot point related to the current workpiece zero. If a coordinate is not programmed the pivot point is at the workpiece zero. The coordinates of the pivot point act like a zero offset.
I.. J.. K..:	Vector in the pivot point. The coordinate system is rotated about this vector by the angle R.
R..:	Angle of rotation, always interpreted as an absolute value. If an angle is not programmed, the angle from setting data 42150 $SA_DEFAULT_ROT_FACTOR_R is active.

G68 must be in a block of its own. A G90/91 in the block has no effect on the G68 command.

Explanations

The distinction between 2D and 3D rotation is determined solely by programming the vector I, J, K. If no vector exists in the block, G68 2DRot is selected. If a vector exists in the block, G68 3DRot is selected.

If a vector of length 0 (I0, Y0, K0) is programmed, the alarm 12560 ”programmed value %3 exceeds allowed limits” is output.

With G68, two rotations cn be connected in series. If a G68 is not already active in a block containing G68, the rotation is written into channel–specific base frame 2. If G68 is already active, the rotation is written in channel–specific base frame 3. This means that both rotations are activated in sequence. For this purpose, machine data $MC_MM_NUM_BASE_FRAMES = 4 must be set.

Note

For incoupling the frames between the Siemens and the ISO modes (solution line) see section 3.1.6.

With G69, 3D rotation is terminated. If two rotations are active, they are both deactivated with G69. G69 does not have to be in a block of its own.

3.2 Determining the coordinate value input modes

This section describes the commands used to input coordinate values.

3.2.1 Absolute/incremental designation (G90, G91)

These G codes specify whether dimension values specified following an axis address are given in an absolute value or incremental value.

Using the G90/G91 command

Features of G90 and G91

Table 3-3 Absolute/incremental designation G codes

G code	Function	Group
G90	Absolute designation	03
G91	Incremental designation	03

- G90 and G91 are modal G code belonging to 03-group. If G90 and G91 are specified in the same block, the one specified later is valid.
- The power-ON state, that is the G90 or G91 mode that is valid when the power is turned ON, can be set to MD 20154: EXTERN_GCODE_RESET_VALUES[2].

Command format

- For the commands specified in and after the G90 block, the dimension values specified following an address of X, Y, Z, 4th are treated as absolute values.
- For the commands specified in and after the G91 block, the dimension values are treated as incremental values.

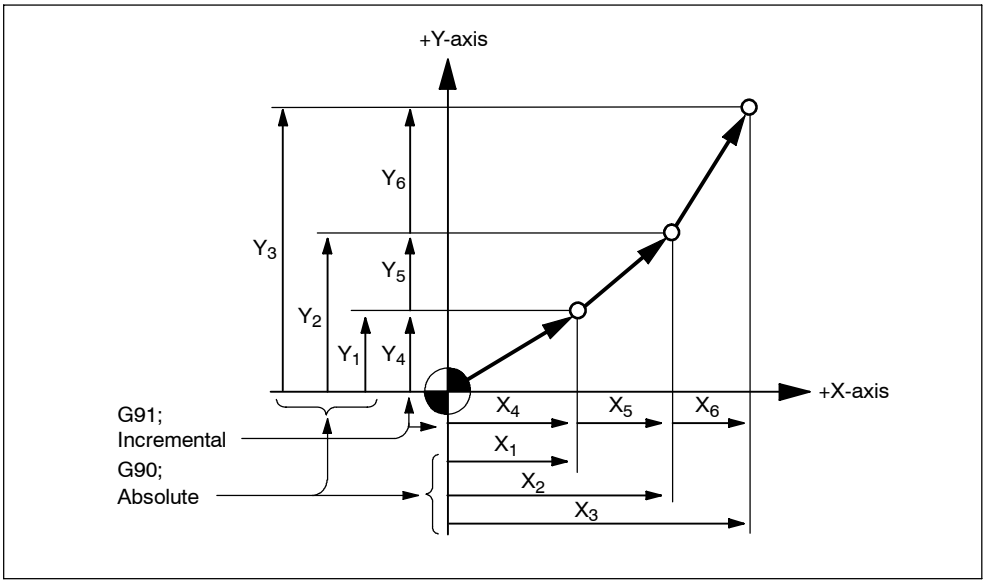


Fig. 3-12 Absolute/incremental commands (G90, G91)

3.2.2 Inch/Metric input designation (G20, G21)

It is possible to select the dimension unit for the input data between “mm” and “inch”. For this selection, the following G codes are used.

Table 3-4 Dimension unit selection G codes

G code	Function	Group
G20	Input in “inch” system	06
G21	Input in “mm” system	06

Command format

G20 and G21 should be specified at the beginning of a program in a block without other commands. When the G code which selects the input dimension unit is executed, the following values are processed in the selected dimension unit: subsequent programs, offset amount, a part of parameters, a part of manual operation and display.

```
G291;
G20;
        .
        .  <-------  Designating the input in “inch” system
        .
```

Fig. 3-13 Example pf programming

Supplements to the dimension unit designation commands

- The state when the power is turned ON is determined by MD $MC_EXTERN_GCODE_RESET_VALUES[5].
- On switchover, the zero offset values are converted completely.
- If the dimension unit system should be switched over during the execution of a program, the following processing must be accomplished in advance.

  If a workpiece coordinate system (G54 to G59) is being used, return it to the base coordinate system.

  Cancel all tool offsets (G41 to G48).
- After switching over the dimension unit system between G20 and G21, the following processing must be accomplished.

  Execute G92 (coordinate system setting) for all axes before specifying axis move commands.
- The handwheel and increment weighting are not switched over with G20 and G21. This switchover is initiated by PLC program in this case. The relevant MD is $MA_JOG_INCR_WEIGHT.

3.2.3 Scaling (G50, G51)

The shape defined by a part program can be enlarged or reduced according to a required scale. For the scaling processing, the following G codes are used.

Table 3-5 Scaling G codes

G code	Function	Group
G50	Scaling OFF	11
G51	Scaling ON	11

The G50 and G51 blocks must be specified in the manner as indicated above without other commands entered in these blocks. The scaling function which is called by G51 must be canceled by G50. If G51 is specified in the scaling mode, it is disregarded.

Format

Two different kinds of scaling can be applied.

Scaling along all axes at same magnification rate

G51 X... Y... Z... P... ; Scaling start
G50; Scaling cancel

X, Y, Z: Center coordinate value of scaling (absolute command)
P: Scaling magnification

Scaling along each individual axis at a different magnification rate (mirror image)

G51 X... Y... Z... I... J... K... ; Scaling start
G50; Scaling cancel

X, Y, Z: Center coordinate value of scaling (absolute command)
I, J, K: X–, Y–, and Z axis scaling magnification

The type of scaling magnification is dependent on MD 22914 $MC_AXES_SCALE_ENABLE.

$MC_AXES_SCALE_ENABLE = 0:
”P” is available for magnification rate. If ”I,J,K” is programmed in this setting, SD 42140 $SC_DEFAULT_SCALE_FACTOR_P is used for magnification rate.

$MC_AXES_SCALE_ENABLE = 1:
”I,J,K” are available for magnification rate. If ”P” is programmed in this setting, SD 43120 $SC_DEFAULT_SCALE_FACTOR_AXIS is used for magnification rate.

Explanations

Scaling along all axes at same magnification rate

Least input increment of scaling magnification is: 0.001 or 0.00001 depending on the setting of MD $MN_EXTERNINCREMENT_SYSTEM. If P is not specified in the block of scaling (G51X... Y... Z... P... ;), the scaling magnification set to MD $MC_WEIGHTING_FACTOR_FOR_SCALE is applied.
The reference point during scaling is always the workpiece zero. It is not possible to program a reference point.

Programmable mirror image (negative magnification)

Applying a negative magnification value will generate a mirror image.
Each axis scaling (mirror image) needs to be enabled by setting MD $MC_AXES_SCALE_ENABLE = 1.
Omitting I, J, K within the G51 block activates the default values from the setting data.

Example

```
_N_0512_MPF;              (part program)
N01 G291;
N10 G17 G90 G00 X0 Y0;    Approach start position
N30 G90 G01 G94 F6000;
N32 M98 P0513;            1) Contour as programmed in subprogram
N34 G51 X0. Y0. I-1000 J1000;    2) Mirror contour around X
N36 M98 P0513;
N38 G51 X0. Y0. I-1000 J-1000;  3) Mirror contour around X and Y
N40 M98 P0513;
N42 G51 X0. Y0. I1000 J-1000;    4) Mirror contour around Y
N44 M98 P0513;
N46 G50;                  Deselect scaling and mirroring
N50 G00 X0 Y0
N60 M30

_N_0513_MPF;              (subprogram for 00512)
N01 G291
N10 G90 X10. Y10.;
N20 X50;
N30 Y50;
N40 X10. Y10.;
N50 M99;
```

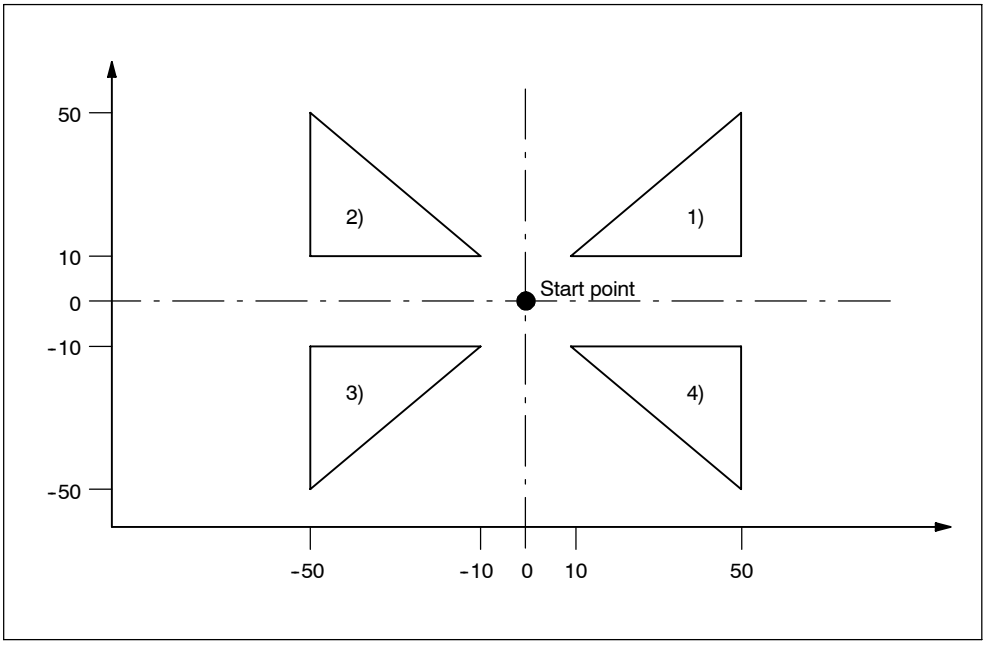


Fig. 3-14 Scaling of each axis, programmable mirror image

Tool compensation

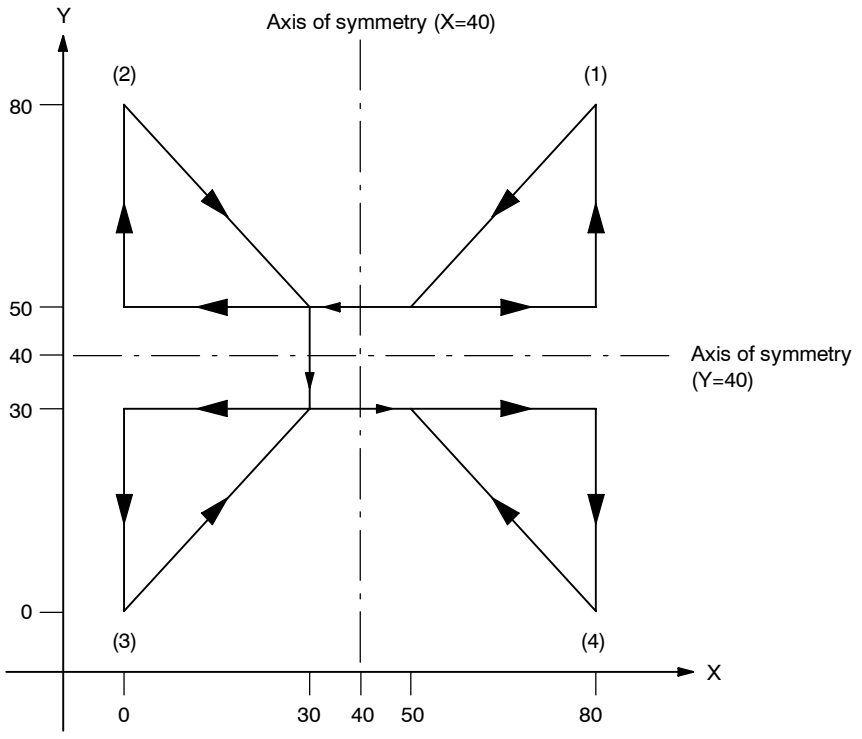
This scaling is not applicable to cutter compensation values, tool length offset values, and tool offset values.

Commands with respect to reference position return and coordinate system

Do not use G27,G28,G30, or commands related to the coordinate system (G52 to G59,G92) in scaling mode.

3.2.4 Programmable mirror image (G50.1, G51.1)

With respect to a programmed axis of symmetry, a mirror image of a programmed contour can be created.



(1) Originally programmed contour
(2) Symmetrical image with respect to a line parallel to Y and crossing X at 40
(3) Symmetrical image with respect to a point (40, 40)
(4) Symmetrical image with respect to a line parallel to X and crossing Y at 40

Fig. 3-15 Programmable mirror image

Format

```
G51.1 X... Y... Z... ; Creating a programmable image
...          ;
...          ; These blocks describe the contour through which a mirror image
...          ; is created with respect to the axis of symmetry
...          ; specified by G51.1 X... Y... Z... ;
...          ;
G50.1 X... Y... Z... ; Programmable mirror image cancel
```

X, Y, Z :
Position and axis of symmetry for creating a mirror image when specified through G51.1.

Explanations

Related machine data

G51.1 uses the channel specific basic frame[1]. Therefore, set MD $MC_MM_NUM_BASE_FRAMES > = 2.

Mirror image with respect to single axis in a specified plane

The following commands are subject to be changed when applying mirror image to one of the axes on a preset plane as described below:

Table 3-6

Command	Explanation
Circular interpolation	G02 and G03 are interchanged
Cutter compensation	G41 and G42 are interchanged
Coordinate rotation	CW and CCW (directions of rotation) are interchanged

Limitations

Scaling/coordinate system rotation

Processing proceeds from program mirror image to scaling and coordinate rotation in the stated order. The commands should be specified in this order, and, for cancellation, in the reverse order.
Do not specify G50.1 or G51.1 during scaling or coordinate rotation mode.

Commands related to reference position return and coordinate system

Do not use G codes related to reference position return (G27,G28,G30), or commands related to the coordinate system (G52 to G59,G92, etc.) in programmable mirror image mode.

3.2.5 G60: Oriented positioning

Does not work with SINUMERIK 802D sl.

G60 is used in the ISO dialect original for backlash compensation. With Sinumerik, it is achieved using the internal backlash compensation; therefore, there is no G function in the Siemens mode, which corresponds to G60 in the ISO dialect original.

It is not possible to replace G60 by a G macro call, since it is not possible to execute two subroutine calls in one NC block. Since the oriented positioning (backlash) must be performed before executing the NC block, the call of a G macro at the end of the block would be too late.

Since G60 is used for backlash compensation and this function can be activated via the axial machine data $MA_BACKLASH[ ], G60 is skipped in the ISO mode without triggering a reaction.

If the programmed G60 is to be taken into account when running envelope cycles, this information is provided to the cycle variable $C_G60_PROG. If G60 is programmed, $C_G60_PROG = 1 is set; $C_G60_PROG is canceled with return to the subroutine. If you require, in addition, the information in a block whether the cycle call is also programmed, you can take this information from the cycle variable $C_G_PROG. The information from these two system variables can be used to add a G60 functionality to the envelope cycles. The information whether a modal cycle is active can also be obtained from the system variable $P_MC ($P_MC = 1 -> a modal subroutine is active).

$C_G60_PROG is only set to ”1” if G60 is programmed in an NC block such as if G60 were a modal G function.

Example:

```
N32 G00 X0. Y0. Z0. R0.
N33 G60 X11.8407 Y2.4418   ;$C_G60_PROG = 1, $C_G_PROG = 0, $P_MC = 0
N34 G60 G83 X11.8407 Y2.4418 Z-6.9051 R-5.9 Q0.25F8
                           ;$C_G60_PROG = 1,
                           ;$C_G_PROG = 1, $P_MC = 1
N35 G60 X9.3969 Y2.6099    ;$C_G60_PROG = 1, $C_G_PROG = 0, $P_MC = 1
N36 X6.4128 Y2.4511        ;$C_G60_PROG = 0, $C_G_PROG = 0, $P_MC = 1
N37 G60 X4.0368 Y2.3131    ;$C_G60_PROG = 1, $C_G_PROG = 0, $P_MC = 1
N38 G60 X1.3995 Y2.5461    :$C_G60_PROG = 1, $C_G_PROG = 0, $P_MC = 1
N39 G80                    ;$C_G60_PROG = 0, $C_G_PROG = 0, $P_MC = 0
```

cycle383m.spf

```
PROG CYCLE383M
....
IF $C_G60_PROG == 1
                ;G60 functionality
ENDIF
                ;Continue with the envelope cycle functionality
```

3.3 Time-controlling commands

3.3.1 Dwell (G04)

It is possible to suspend the execution of axis move commands specified in the next block for the specified length of time (dwell period) or a number of spindle revolutions.
In the feed per minute mode (G94) the dwell time unit is seconds [s], while in the feed per revolution mode (G95) the dwell time unit is spindle revolution [rev].

Format

G04 X_; or G04 P_;

X_:Specify a time (decimal point permitted)
P_:Specify a time (decimal point not permitted)

By specifying G04 X_; or G04 P_; execution of programmed commands is suspended for the length of time or number of spindle revolutions specified by address X or P.

- The block used to specify dwell must not include commands other than G04 commands.
- The maximum programmable value with address X or P is indicated in the table below.

Table 3-7 Command value range of dwell time (command by X)

Increment system	Command value range	Dwell time unit
IS-B	0.001 to 99999.999	s or rev
IS-C	0.0001 to 9999.9999	s or rev

Table 3-8 Command value range of dwell time (command by P)

Increment system	Command value range	Dwell time unit
IS-B	1 to 99999999	0.001 s or rev
IS-C	1 to 99999999	0.001 s or rev

3.4 Cutting feedrate control

3.4.1 Automatic corner override G62

It is often better to reduce the feedrate at inside corners with active tool radius compensation.

G62 is operative only at inside corners with active tool radius compensation and active continuous-path mode. Any corner whose inside angle is higher than the setting in $SC_CORNER_SLOWDOWN_CRIT is ignored. The inside angle is determined by the bend in the contour.

The feedrate is reduced by the factor $SC_CORNER_SLOWDOWN_OVR:
Applied feedrate = F * $SC_CORNER_SLOWDOWN_OVR * feedrate override.
The feedrate override is the product of the feedrate override from the machine control panel multiplied by the override from synchronized actions.

The feedrate reduction starts at a distance of $SC_CORNER_SLOWDOWN_START before the corner. It ends at a distance of $SC_CORNER_SLOWDOWN_END after the corner (see Fig. 3-16). An appropriate path is traveled on curved contours.

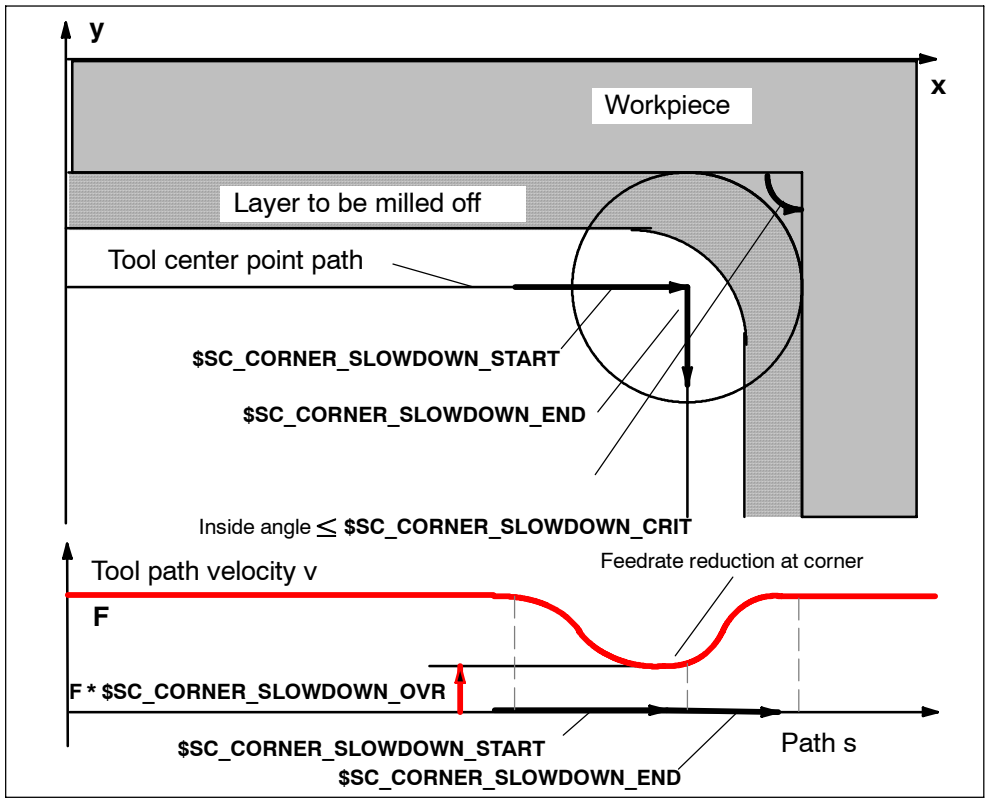


Fig. 3-16 Parameterization of feedrate reduction G62 illustrated by example of a 90° corner

Parameterization

The override value is programmed via the following setting data:

42520: $SC_CORNER_SLOWDOWN_START
42522: $SC_CORNER_SLOWDOWN_END
42524: $SC_CORNER_SLOWDOWN_OVR
42526: $SC_CORNER_SLOWDOWN_CRIT

The setting data are preset to a value of 0.

- If $SC_CORNER_SLOWDOWN_CRIT == 0, corner deceleration is operative only at reversing points.
- If $SC_CORNER_SLOWDOWN_START and $SC_CORNER_SLOWDOWN_END equal 0, the feedrate reduction is applied within the permissible dynamic response limits.
- If $SC_CORNER_SLOWDOWN_OVR == 0, a brief stop is inserted.
- $SC_CORNER_SLOWDOWN_CRIT refers to the geometry axes with G62. It defines the maximum inside angle in the current machining plane up to which corner deceleration is applied. – G62 is not operative with rapid traverse.

Activ action

The function is activated via G62 or G621. The G code is activated either by the relevant part program command or via $MC_GCODE_RESET_VALUES[56].

Example

```
$TC_DP1[1,1]=120
$TC_DP3[1,1]=0.                 ;Length compensation vector
$TC_DP4[1,1]=0.
$TC_DP5[1,1]=0.
$TC_DP6[1,1]=10                 ;Tool radius

N1000 G0 X0 Y0 Z0 F5000 G64 SOFT

N1010 STOPRE
N1020 $SC_CORNER_SLOWDOWN_START = 5.
N1030 $SC_CORNER_SLOWDOWN_END = 8.
N1040 $SC_CORNER_SLOWDOWN_OVR = 20.
N1050 $SC_CORNER_SLOWDOWN_CRIT = 100.

N2010 G1 X00 Y30 G90 T1 D1 G64
N2020 G1 X40 Y0  G62 G41;       Inside corner for N2030,
                                but TRC still being selected
N2030 G1 X80 Y30;               Inside corner for N2040 127de-
                                grees
N2040 G1 Y70;                   Inside corner for N2050 53 degrees
N2050 G1 X40 Y40;               Outside corner for N2060
N2060 G1 X20 Y70;               Inside corner for N2070 97 degrees
N2070 G1 X00 Y60;               Inside corner for N2080 90 degrees
N2080 G1 X20 Y20;               Outside corner for N2090,
                                irrelevant because TRC deselected
N2090 G1 X00 Y00 G40 FENDNORM

M30
```

3.4.2 Compressor in ISO dialect mode

The commands COMPON, COMPCURV, COMPCAD are Siemens language commands. They activate a compressor function which links a number of linear blocks to form a machining section.
If the compressor function is activated in Siemens mode, it can now be used to compress linear blocks in ISO dialect mode.
The blocks may not contain any commands other than those listed below:

- Block number
- G01, modal or nonmodal
- Axis assignments
- Feedrate
- Comments

If a block contains any other commands (e.g. auxiliary functions, other G codes, etc.), it will not be compressed.
Values can be assigned with $x for G, axes and feedrate and the Skip function can also be utilized.

Example: The function will compress these blocks

```
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)
```

The function will **not** compress these blocks

```
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
```

3.4.3 Exact stop (G09, G61), cutting mode (G64), tapping mode (G63)

Cutting feedrate can be controlled, as indicated in the table below.

Table 3-9

Designation	G code	Validity of G code	Description
Exact stop	G09	Valid only in the block where specified	Deceleration and stop at the end point of the block, position check prior to proceeding to the next block
Exact stop mode	G61	Modal G code, remains effective until G62, G63, or G64 is specified.	Deceleration and stop at the end point of the block, position check prior to proceeding to the next block
Cutting mode	G64	Modal G code, remains effective until G61, G62, or G63 is specified.	No deceleration to the end point of the block prior to proceeding to the next block
Tapping mode	G63	Modal G code, remains effective until G61, G62, or G64 is specified.	No deceleration to the end point of the block prior to proceeding to the next block, feedrate override is not effective

Format

G09 X... Y... Z... ; Exact stop
G61 ; Exact stop mode
G64 ; Cutting mode
G63 ; Tapping mode

3.5 Tool offset functions

3.5.1 Tool offset data memory

Since Siemens and ISO Dialect programs are to run alternately on the control, the implementation must use the Siemens tool data memory. The length, geometry and wear are therefore available in each offset memory. In Siemens mode, the offset memory is addressed by T (tool number) and D (tool edge number), abbreviated to T/D number.
In ISO Dialect programs, the offset number is addressed by D (radius) or H (length), referred to below as D/H number.
In order to establish a unique assignment between the D or H number and a T/D number, an element $TC_DPH[t,d] has been added to the offset data set. The D/H number of the ISO Dialect is entered in this element.

Table 3-10 Example: tool offset data set

T	D/cutting edge	ISO_H $TC_DPH	Radius	Length
1	1	10		
1	2	11		
1	3	12		
2	1	13		
2	2	14		
2	3	15		

Setting data $SC_TOOL_LENGTH_CONST must contain the value 17 for the assignment of tool length offsets to geometry axes to be independent of the plane selection. Length 1 is then always assigned to the Z axis.

3.5.2 Tool length offset (G43, G44, G49)

The tool length offset function adds or subtracts the amount stored in the tool offset data memory to or from the Z coordinate values specified in a program to offset the programmed paths according to the length of a cutting tool.

Commands

In the execution of the tool length offset function, addition or subtraction of the offset data is determined by the specified G code and the direction of offset by the H code.

G Codes used for tool length offset

The tool length offset function is called by the following G codes.

Table 3-11 G codes used for tool length offset

G code	Function	Group
G43	add	08
G44	Substract	08
G49	Cancel	08

- G43 and G44 are modal and, once executed, they remain valid until canceled by G49. G49 cancels the tool length offset mode. H00 also cancels the tool length offset mode.
- By specifying “G43 (or G44) Z ··· H ··· ; ”, tool offset amount specified by the H code is added to or subtracted from the specified Z-axis position, and the Z-axis moves to this offset target position. That is, the target position of Z-axis movement specified in the program is offset by the tool offset amount.
- By specifying “(G01) Z ··· ; G43 (or G44) H ··· ; ”, the Z-axis moves by the distance corresponding to the tool offset amount which is specified by the H code.
- By specifying “G43 (or G44) Z ··· H ··· ; H ··· ; ”, the Z-axis moves by the distance which is equivalent to the difference between the previous tool offset amount and the new tool offset amount.

Note

G43, G44, and G49 can be specified only in the mode called by the G code (G00, G01) in 01 group. If they are specified in other modes such as G02 or G03 mode, an error occurs.

H code for designating the direction of offset

The direction of offset is determined by the sign of tool offset amount, specified by an H code, and a G code.

Table 3-12 Sign of tool offset amount and direction of offset

	Sign of tool offset amount (H code)	
	Positive	**Negative**
G43	Offset in the positive direction	Offset in the negative direction
G44	Offset in the negative direction	Offset in the positive direction

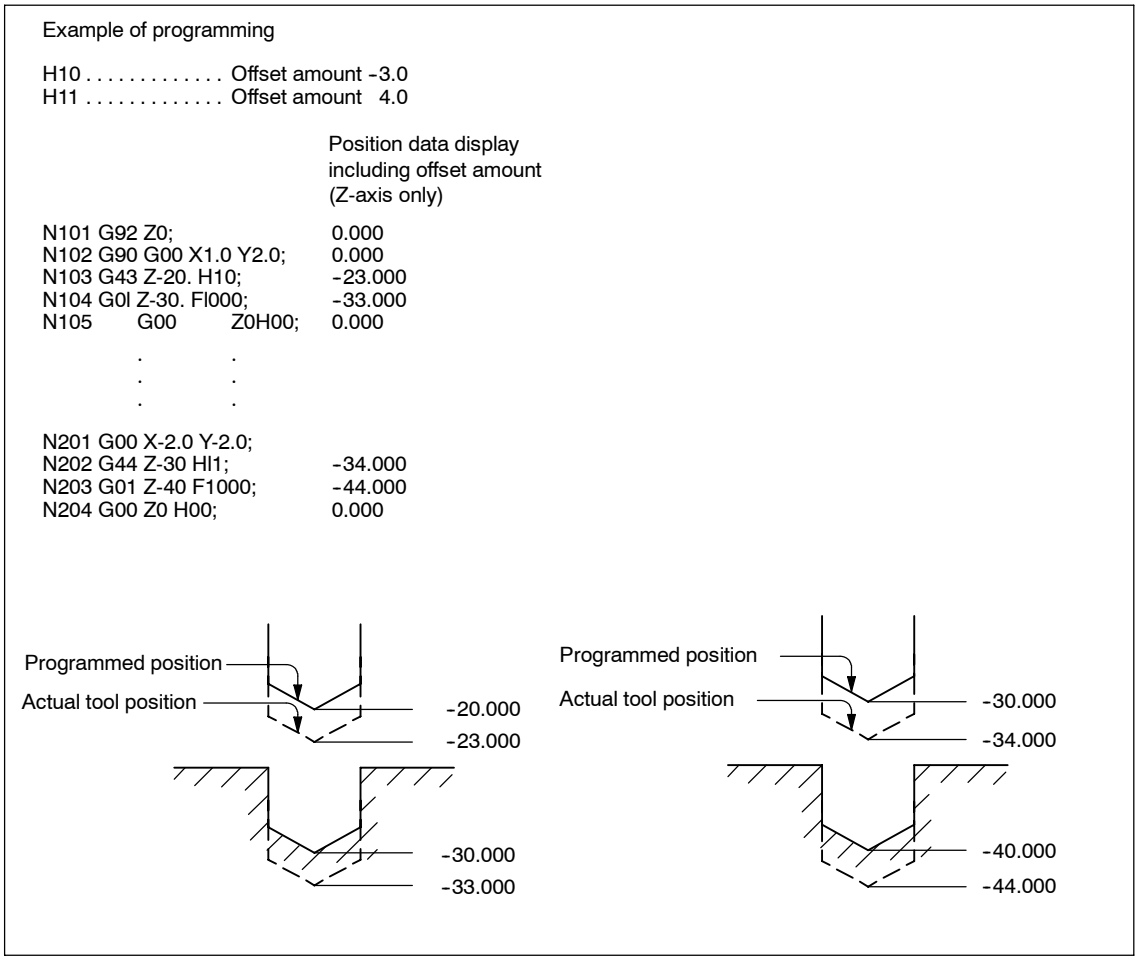


Fig. 3-17 Tool position offset function

- Related Machine data:
  $MC_TOOL_CORR_MOVE_MODE defines whether the compensation is applied in the block containing the selection or the next time the axis is programmed.

  $MC_CUTTING_EDGE_DEFAULT = 0 defines that no tool length compensa-

tion is active initially on a tool change.

$MC_AUXFU_T_SYNC_TYPE is used to define whether the output to PLC takes place during or after the movement.

$MC_RESET_MODE_MASK, bit 6 can be used to activate tool length compensation beyond a reset.

- It is possible to call up the cutter compensation function in the tool length offset mode.
- It is not allowed to specify G43, G44, or G49 in a canned cycle mode.
- G43, G44, and G49 can be specified only in the G00 or G01 mode. Designation of these G codes in the G02 or G03 mode is not allowed.

Tool length compensation in multiple axes

Tool length offsets can be activated on multiple axes. However, it is not possible to display the resulting tool length compensation.

3.5.3 Cutter radius compensation (G40, G41, G42)

The cutter radius compensation function automatically offsets the programmed tool paths by specifying the radius of the cutting tool to be used. The distance to be offset (radius of cutting tool) can be stored to the tool offset data memory by using the NC operation panel. Existing tool offsets can be overwritten using a G10 command, however, new tool offsets cannot be created by G10.
In a program, the offset data are called up by specifying the number of the tool offset data memory using a D code.

Commands

To call up the cutter radius compensation function, the following G codes are used.

Table 3-13 G codes used to call up the cutter radius compensation function

G code	Function	Group
G40	Tool radius offset C mode cancel	07
G41	Tool radius offset C (offset to the left)	07
G42	Tool radius offset C (offset to the right)	07

The cutter radius compensation function is called up by the execution of G41 or G42 and canceled by G40. Direction of offset is determined by the designated G code (G41, G42) and the offset amount is selected by the D code which is specified with the G code designated to call up the tool radius offset mode. When the power is turned ON, the G40 mode is set.

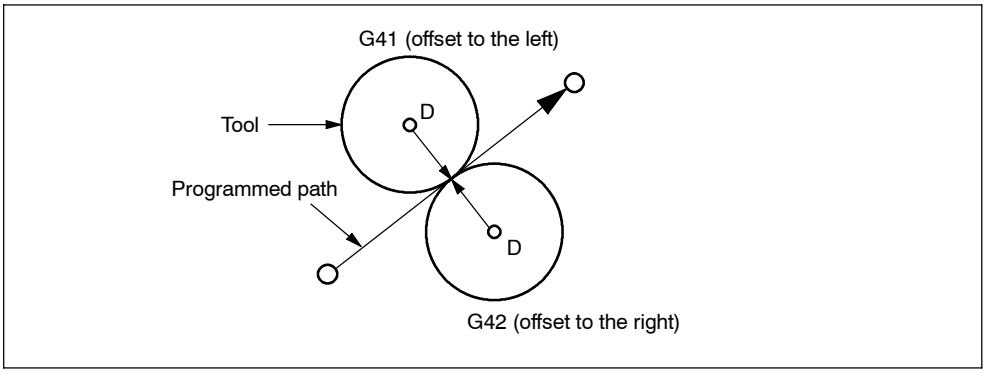


Fig. 3-18 Cutter radius compensation

- If a negative value is set in the tool offset data memory specified by the D code, the offset direction is reversed. The D code must be specified with G41 or G42 in the same block or in a preceding block. If D00 is specified, it specifies the tool radius of “0”.
- The tool radius offset plane is selected by the designation of G17, G18, or G19. The G code used to select the plane must be specified with G41 or G42 in the same block or in a block preceding the G41 or G42 block.

Table 3-14 Plane selection G codes

G code	Function	Group
G17	XY plane selection	02
G18	ZX plane selection	02
G19	YZ plane selection	02

- It is not allowed to change the selected plane in the offset mode. If a plane selection G code is specified in the offset mode, an alarm occurs.

Start-up of cutter compensation

Since the offset start-up is executed with the offset taken into account, the G code in 01-group must be either G00 or G01. If a G code other than G00 or G01 is specified, an alarm occurs. If the offset starts in the G00 mode, the axes move to the offset point at their individual rapid traverse rates. Therefore, be aware of possible interference of a cutting tool with the workpiece.

There are two types of start-up such as start-up at inside corner and start-up at outside corner.

Blocks not including axis move commands in the offset mode

In the tool radius offset mode, the NC generates the tool paths by buffering the data of two blocks. If a block not including axis move commands is read, the NC reads one more block to generate the offset tool paths. Designation of such a block which does not include axis move commands is allowed in the tool radius offset mode for up to two consecutive blocks.

After the designation of G41, there must not be three or more consecutive blocks that do not include the movement commands of the axes in the offset plane.

Consecutive three or more blocks not including axis move commands

If three or more blocks not containing axis move commands in the offset plane are given consecutively, the cutting tool is moved to the position offset normally by the specified offset amount at the end point of the block immediately preceding such blocks.

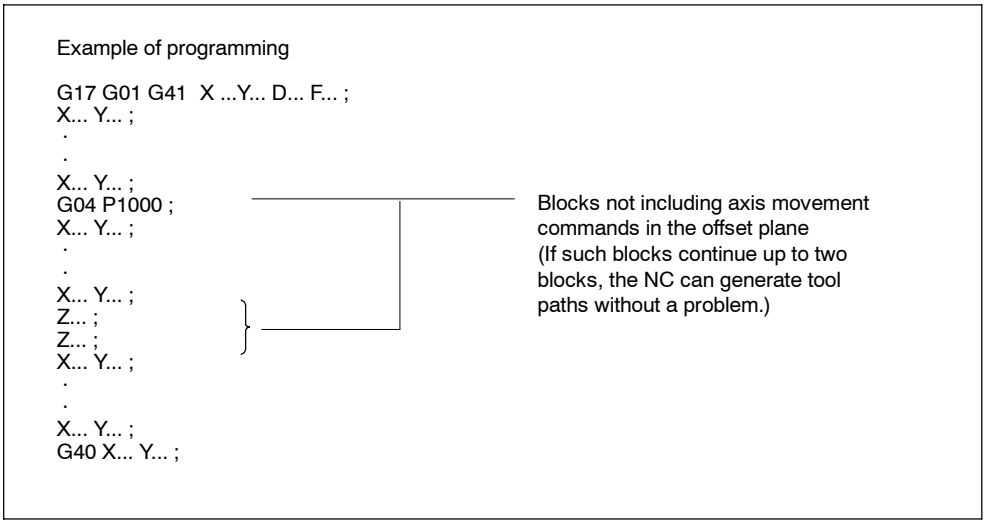


Fig. 3-19 Example of programming

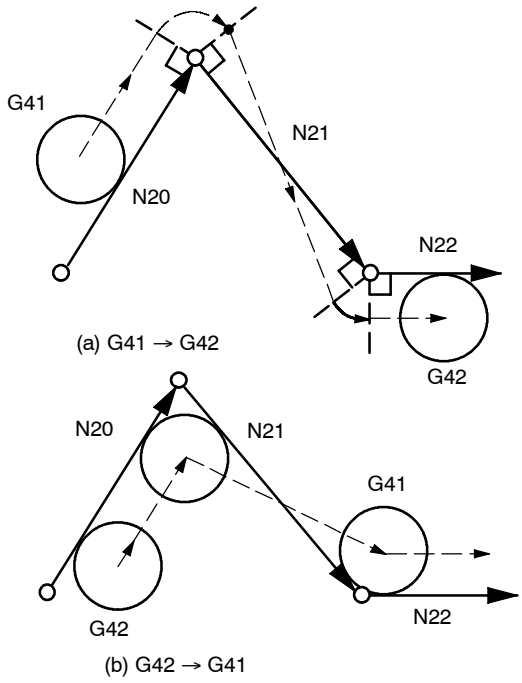
Switching the G41 and G42 in the cutter compensation mode

The direction of offset (left side and right side) can be directly switched without canceling the offset mode.

The offset direction is switched at the start and end of the block in which the switching of the offset direction is specified.

Example of programming

```
N10 G17 G01 F... ;
N11 G41 (G42) D... ;
              .
              .
              .
N20 G01 X... Y... F... ;
N21 G42 (G41) X... Y... ;        ← Offset direction switching block
N22 X...   ;
```



(a) G41 → G42

(b) G42 → G41

Note: If the contents of N21 block are expressed in two blocks as indicated below
G42 (or G41) ;
X Y ;
the offset direction is switched in the same manner.

Fig. 3-20 Switching the offset direction at the start and end of the block

Canceling the offset mode

There are two types of offset mode cancellation methods which can be selected by the MD setting.

1. Type A:

Offset mode cancellation movement is not executed in the G40 block if no axis move commands are given. The offset mode is canceled by the first axis move command given in a block specified following the G40 block. Axis move commands should be specified with G40 in the same block.

2. Type B:

Offset mode cancellation movement is executed in the G40 block even if there are no axis move commands given. The cutting tool moves normally to the offset position at the end point of the block immediately before the G40 block. Since G40 calls up offset mode cancellation axis movements, it must be specified in the G00 or G01 mode. If it is specified in a mode other than G00 or G01, an alarm occurs.

Canceling the offset mode at inside corner (smaller than 180°)

Straight-line to straight-line

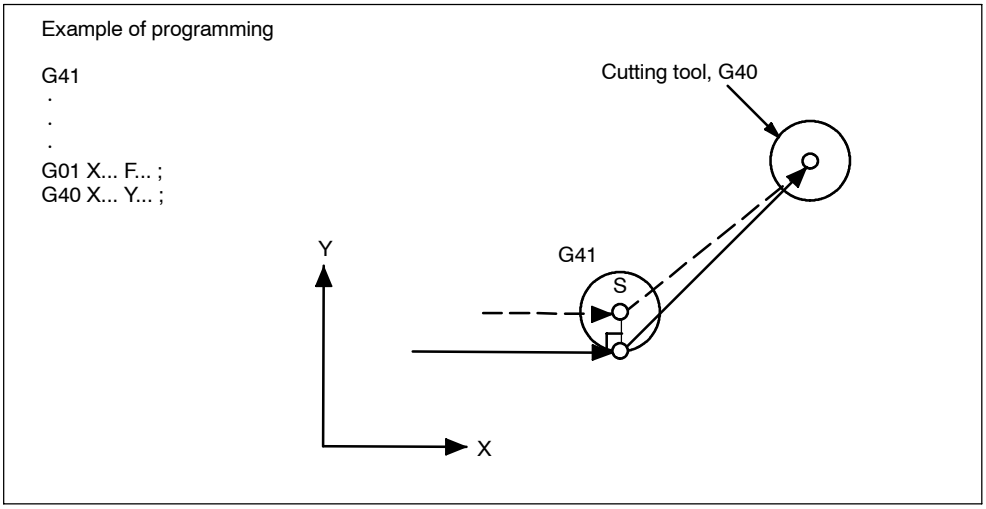


Fig. 3-21 Canceling the offset mode at inside corner (straight-line to straight-line)

Arc to straight-line

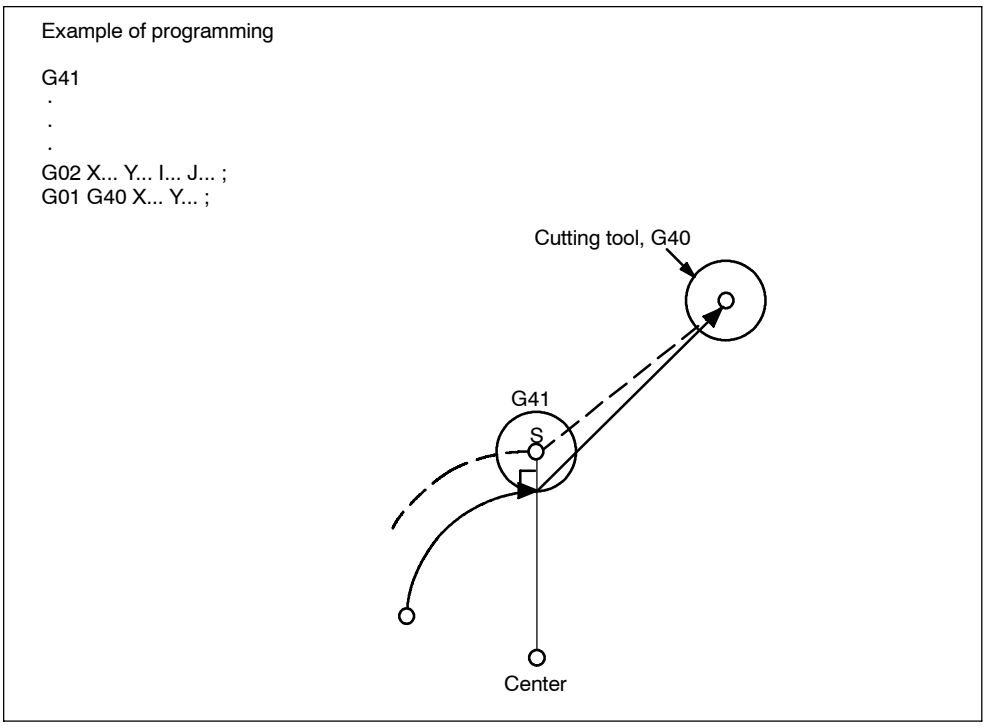


Fig. 3-22 Canceling the offset mode at inside corner (arc to straight-line)

3.5.4 Collision monitoring

Activation by NC program

Although the collision monitoring function is available only in Siemens mode, it can also be applied within the ISO dialect mode. However, activation and deactivation needs to be carried out in Siemens mode.

G290;	Activate Siemens mode
CDON;	Activate bottleneck detection
G291;	Activate ISO dialect mode
...	
...	
G290;	Activate Siemens mode
CDOF;	Deactivate bottleneck detection
G291;	Activate ISO dialect mode

Activation by MD setting

MD 20150 $MC_GCODE_RESET_VALUES[22] = 2: CDON (modal active)
MD 20150 $MC_GCODE_RESET_VALUES[22] = 1: CDOF (modal inactive)

Function

When CDON (Collision Detection ON) and tool radius compensation are active, the control monitors the tool paths with Look Ahead contour calculation. This Look Ahead function allows possible collisions to be detected in advance and permits the control to actively avoid them.
When collision detection is off (CDOF), a search is made at inside corners in the previous traversing block (and if necessary in blocks further back) for a common intersection point for the current block. If no intersection is found with this method, an error is generated.

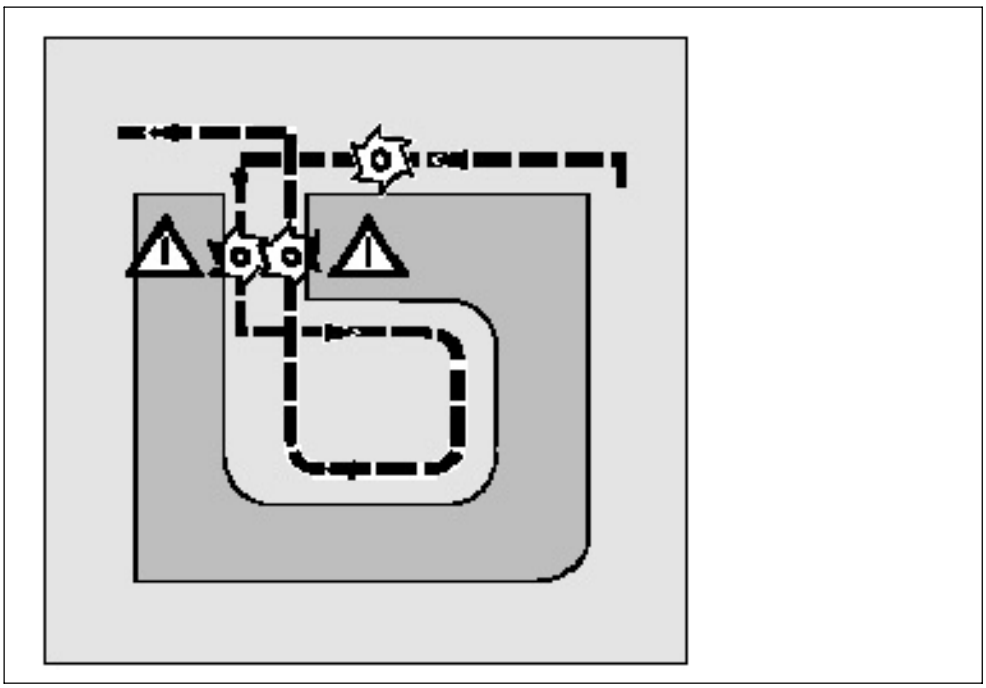


Fig. 3-23 Collision Detection

CDOF helps prevent the incorrect detection of bottlenecks, e.g. due to missing information which is not available in the NC program.
The number of NC blocks monitored can be defined in the machine data (see machine manufacturer).

Examples

The following are some examples of critical machining situations which can be detected by the control and compensated for by modifying the tool paths.
In order to prevent program stops, you should always select the tool with the widest radius from all of the tools used when testing the program.

In each of the following examples a tool with too wide a radius was selected for machining the contour.

Bottleneck detection

Since the tool radius selected is too wide to machine this inside contour, the ”bottleneck” is bypassed. An alarm is output.

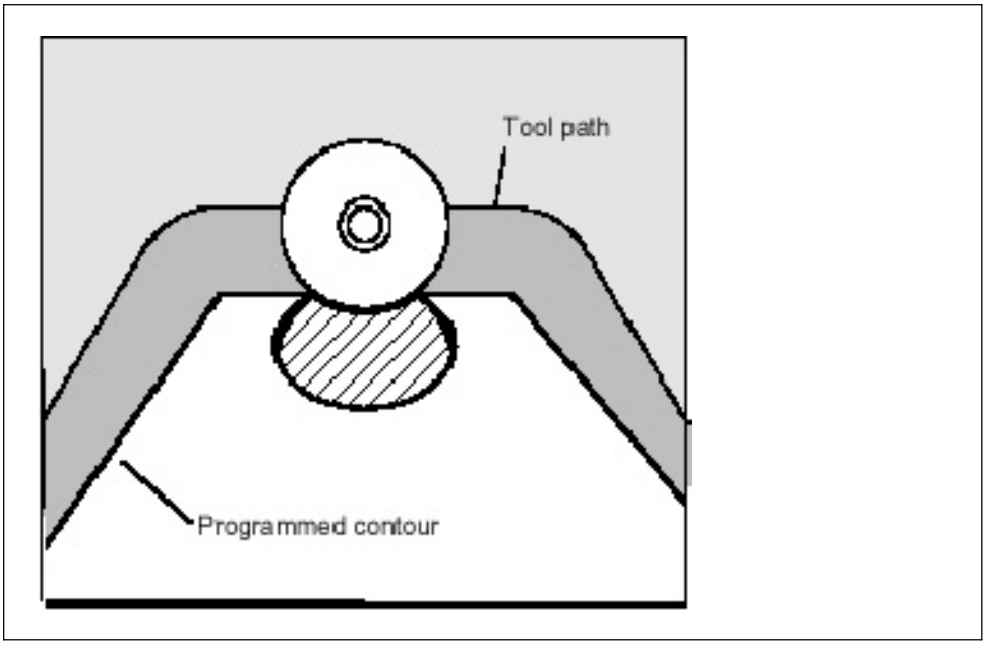


Fig. 3-24 Bottleneck detection

Contour path shorter than tool radius

The tool travels round the workpiece corner on a transition circle and then continues to follow the programmed contour exactly.

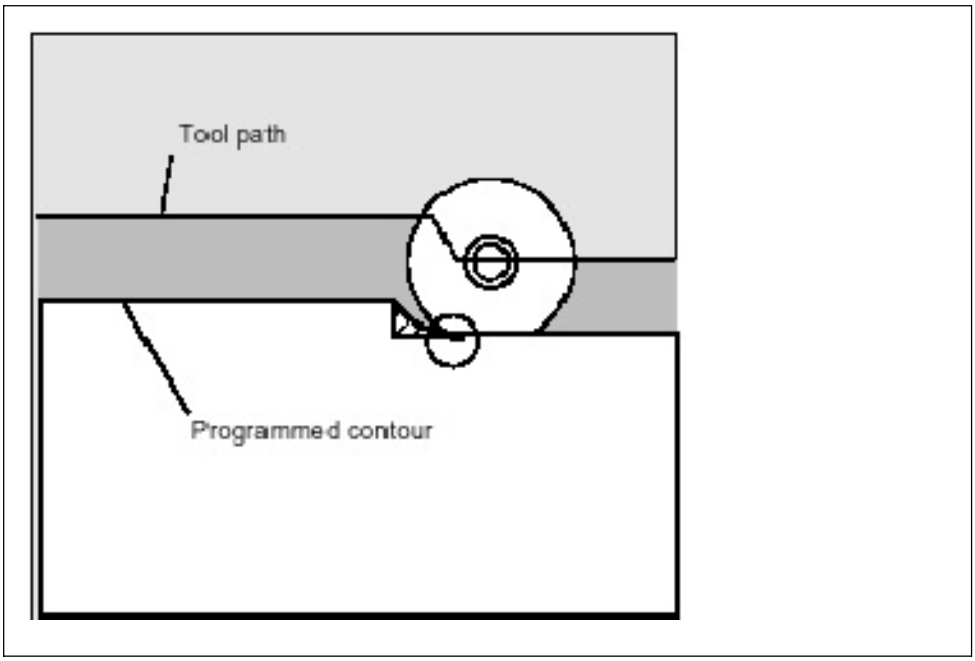


Fig. 3-25 Contour path shorter than tool radius

Tool radius too wide for inside machining

In such cases, machining of the contours is performed only as far as is possible without causing damage to the contour.

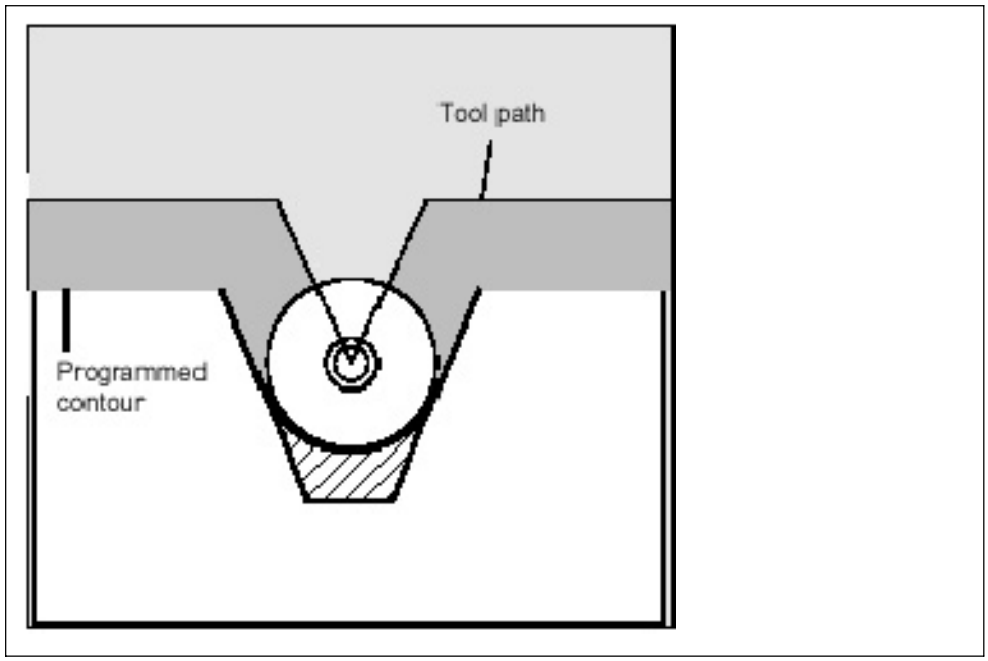


Fig. 3-26 Tool radius too wide for insiede machining

3.6 S, T, M, and B functions

3.6.1 Spindle function (S function)

A spindle speed can be directly specified by entering a 5-digit number following address S (S□□□□□). The unit of spindle speed is “r/min”. If an S command is specified with M03 (spindle forward rotation) or M04 (spindle reverse rotation), the program usually advances to the next block only after the spindle has reached the speed specified by the S command. For details, refer to the instruction manuals published by the machine tool builder.

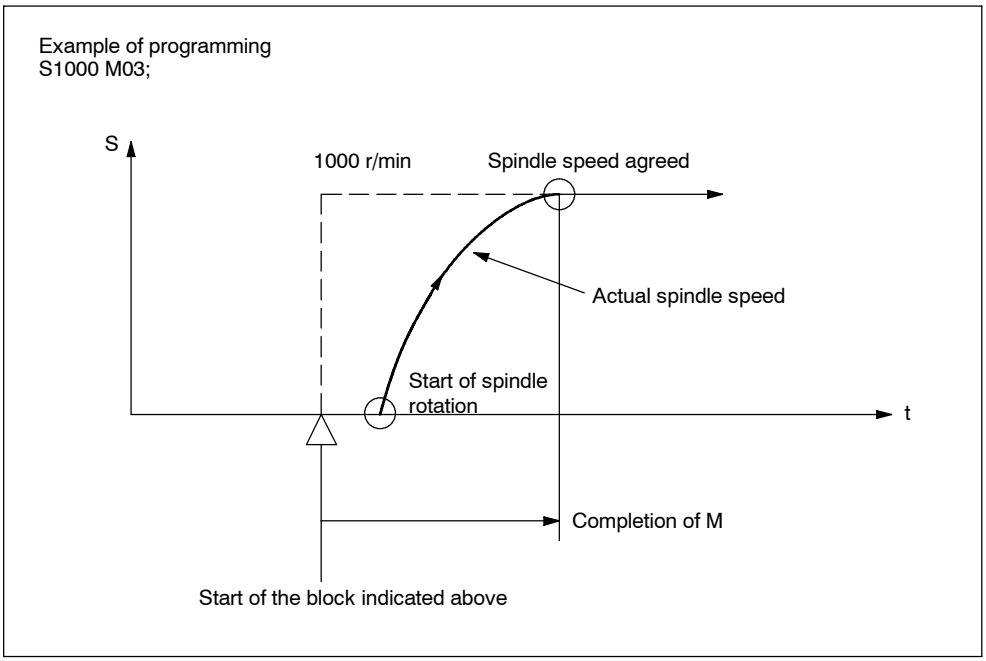


Fig. 3-27 Spindle speed command

- An S command is modal and, once specified, it remains valid until another S command is given next. If the spindle is stopped by the execution of M05, the S command value is retained. Therefore, if M03 or M04 is specified without an S command in the same block, the spindle can start by using the S command value specified before.
- If a spindle speed is changed while the spindle is rotating by the execution of M03 or M04, pay attention to the selected spindle speed gear range. For details, refer to the instruction manuals published by the machine tool builder.
- The lower limit of an S command (S0 or an S command close to S0) is determined by the spindle drive motor and spindle drive system, and it varies with each machine. Do not use a negative value for an S command. For details, refer to the instruction manuals published by the machine tool builder.

3.6.2 Tool function (T function)

The tool function has various command designation types. For details, refer to the instruction manuals published by the machine tool builder.

3.6.3 Miscellaneous function (M function)

The miscellaneous function is specified by a maximum of a three-digit number (M□□□) following address M. With the exception of specific M codes, the functions of M00 to M89 codes are defined by the machine tool builder. Therefore, for details of the M code functions, refer to the instruction manuals published by the machine tool builder.

The M codes specific to the NC are described below.

M codes relating to stop operation (M00, M01, M02, M30)

When an M code relating to stop is executed, the NC stops buffering. Whether spindle rotation, coolant discharge or another operation stops in response to the execution of such an M code is determined by the machine tool builder. For details, refer to the instruction manuals published by the machine tool builder. For these M codes, a code signal is output independently in addition to M2-digit BIN code.

M00 (program stop)

If M00 is specified during automatic operation, automatic operation is interrupted after the completion of the commands specified with M00 in the same block and the M00R signal is output. The interrupted automatic operation can be restarted by pressing the cycle start switch.

M01 (optional stop)

If M01 is executed with the optional stop switch ON, the same operation as with M00 is executed. If the optional stop switch is OFF, M01 is disregarded.

M02 (end of program)

M02 should be specified at the end of a program. When M02 is executed during automatic operation, automatic operation ends after the commands specified with M02 in the same block have been completed. The NC is reset. The state after the end of a program varies with each machine. For details, refer to the instruction manuals published by the machine tool builder.

M30 (end of tape)

Normally, M30 is specified at the end of tape. When M30 is executed during automatic operation, automatic operation ends after the commands specified with M30 in the same block have been completed. The NC is reset and the tape is rewound. The state after the execution of M30 varies with each machine. For details, refer to the instruction manuals published by the machine tool builder.

Note

When M00, M01, M02, or M30 is specified, the NC stops buffering. For these M codes, the NC output the independent decode signal in addition to the M2-digit BIN code.

Note

Refer to the manuals published by the machine tool builder concerning whether or not the spindle and/or coolant supply is stopped by the M00, M01, M02, and M30.

3.6.4 Internally processed M codes

M codes in the range of M90 to M99 are processed by the NC.

Table 3-15 Internally processed M codes

M code	Function
M98	Subprogram call
M99	End of subprogram

3.6.5 Macro call via M function

Similar to G65, a macro can be called via M numbers.

The 10 M-function replacement is configured via machine data
10814: $MN_EXTERN_M_NO_MAC_CYCLE and
10815: $MN_EXTERN_M_NO_MAC_CYCLE_NAME.

The parameters are transferred as with G65. Repeat procedures can be programmed with address L.

Limitations

Only one M function replacement (or one subprogram call) can be performed on each part program line. Conflicts with other subprogram calls are reported with alarm 12722. No more M functions will be replaced in the subprogram replaced.

Generally, the same limitations apply as with G65.

Configuring example

Call the subprogram M101_MAKRO by the M function M101

$MN_EXTERN_M_NO_MAC_CYCLE[0] = 101
$MN_EXTERN_M_NO_MAC_CYCLE_NAME[0] = ”M101_MAKRO”

Call of the subprogram M6_MAKRO by the M function M6.$MN_EXTERN_M_NO_MAC_CYCLE[1] = 6
$MN_EXTERN_M_NO_MAC_CYCLE_NAME[1] = ”M6_MAKRO”

Program examples for tool change with M function:

```
PROC MAIN
...
N10          M6 X10 V20
...
N90          M30PROC M6_MAKRO
...
N0010        R10 = R10 + 11.11
N0020        IF $C_X_PROG == 1 GOTOF N40 display($C_X_PROG)
N0030        SETAL(61000)     ;programmed variable transferred
                              ;incorrectly
N0040        IF $C_V == 20 GTOF N60 display($C_V)
N0050        SETAL(61001)
N0060        M17
```

3.6.6 General purpose M codes

Other general M codes

The functions of the M codes other than the specific M codes are determined by the machine tool builder. The representative use of several general M codes is given below. For details, refer to the instruction manuals published by the machine tool builder. If an M code is specified with axis move commands in the same block, whether the M code is executed with the axis move commands simultaneously or it is executed after the completion of the axis move commands is determined by the machine tool builder. For details, refer to the instruction manuals published by the machine tool builder.

Table 3-16 Other general M codes

M code	Function	Remarks
M03	Spindle start, forward direction	Generally, M state between M03 and M04 cannot be switched directly. To change the M code state, execute M05 once.
M04	Spindle start, reverse direction	
M05	Spindle stop	
M08	Coolant ON	
M09	Coolant OFF	

Designation of multiple M codes in a single block

It is possible to specify up to five M codes in a single block. The specified M codes and sampling output are output at the same time. Concerning the combinations of the M codes that can be specified in the same block, refer to the manuals published by the machine tool builder for restrictions on them.

Second miscellaneous function (B function)

B functions are output to the PLC as H auxiliary functions with address extension H1=.
Example: B1234 is output as H1=1234.

Enhanced Level Commands

4

4.1 Program support functions (1)

4.1.1 Canned cycles (G73 to G89)

By using canned cycles, it is made easier for the programmer to create programs. By means of canned cycles, machining operations frequently used can be determined in a single block through a G function. Normally more than one block is required when operating without canned cycles. Using canned cycles can also shorten the program in order to save memory.

The functionality of the ISO Dialect cycles is implemented in the standard Siemens cycles. A shell cycle is called from the ISO Dialect program. All addresses programmed in the block are passed to this shell cycle in the form of system variables. The shell cycle matches the data to the standard Siemens cycle and calls it by name.

Procedure for cycle call via G command

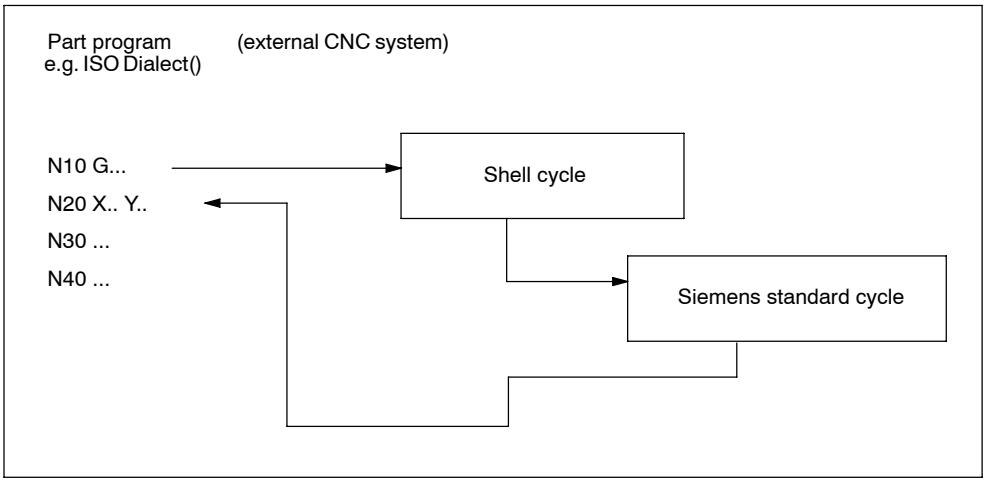


Fig. 4-1 General cycle call in ISO Dialect mode

Cycle parameters

Various cycle parameters in channel-specific GUD (Global User Data) must be initialized for the machining cycles. The names and meanings of the GUD are listed in the tables below.

Table 4-1 GUD7 for programmed cycle values (ISO dialect program data)

GUD	Description/use	Cycle
Real values		
_ZFPR[0]	Initial plane (current position on 1st call with G..), retraction position active on G98	381M, 383M, 384M, 387M
_ZFPR[1]	Reference plane, retraction position active on G99 (retraction is only possible to initial position with G87).	381M, 383M, 384M, 387M
_ZFPR[2]	Final drilling depth	381M, 383M, 384M, 387M
_ZFPR[3]	Retraction position, depending on G98/G99 (initial plane/R plane)	381M, 383M, 384M, 387M
_ZFPR[4]	Feed rate for drilling	381M, 383M, 384M, 387M
_ZFPR[5]	Dwell time at final depth (G82/G89/G76/G87)	381M, 384M, 387M
_ZFPR[6]	1st drilling depth (single drilling depth), incremental (G73/G83)	383M
_ZFPR[7]	1st drilling depth, absolute (G73/G83)	383M
_ZFPR[8]	Lift-off/in-feed distance (G76)	387M
_ZFPR[9]	Speed for tapping (G74/G84)	384M
Integer values		
_ZFPI[0]	Current G code of ISO Dialect0 drilling cycle	381M, 383M, 384M
_ZFPI[1]	M function for spindle start (M3, M4) after spindle stop	381M, 384M

Table 4-2 GUD7 for cycle setting data (ISO dialect setting data)

GUD	Description/use	Cycle
Integer values		
_ZSFR[0]	Safety clearance to reference plane	381M, 383M
_ZSFR[1]	Retraction amount for chip-breaking (G73). The value ”0” means 1 mm or 1 inch. If it is need to specify 0 mm or 0 inch, set a smaller value than the movable resolution.	383M
_ZSFR[2]	Angle offset for oriented spindle stop, tool must be oriented in the reverse directionof retraction (G76) The retraction direction is set by _ZSFI[5].	387M
_ZSFR[10]	(improve to the specification which can set up ”d” value for G83 > 0 = value is used for anticipation distance (distance minimal 0.001) = 0 = distance d is calculated internally	

Table 4-2 GUD7 for cycle setting data (ISO dialect setting data), continued

Integer values		
Integer values		
_ZSFI[0]	0=Drilling axis is the axis perpendicular to the selected plane (default) 1=Drilling axis always "Z"	381M, 383M, 384M, 387M
_ZSFI[1]	0= Rigid tapping 1= Tapping with compensating chuck 2= Deep hole tapping with chip–breaking 3= Deep hole tapping with swarf removal	384M, 387M
_ZSFI[2]	Retraction speed factor (1–200%) for tapping (G74/G84)	384M
_ZSFI[3]	Polar coordinates 0 = OFF 1 = ON	381M, 383M, 384M, 387M
_ZSFI[5]	Retraction direction (G76/G87) G17 G18 G19 _ZSFI[5] = 1 +X +Z +Y _ZSFI[5] = 0 or 2 –X –Z –Y _ZSFI[5] = 3 +Y +X +Z _ZSFI[5] = 4 –Y –X –Z	

For calling normal hole-machining canned cycles, the following G codes are used.

Table 4-3 GUD7 for cycle setting data (ISO dialect setting data)

G code	Drilling (–Z direction)	Processing at hole bottom	Retraction (+Z direction)	Applications
G73	Intermittent feed (dwell at each peck feed possible)	—	Rapid traverse	High-speed deep hole drilling
G74	Cutting feed	Spindle stop→ Spindle rotation in the reverse direction after dwell	Cutting feed → Dwell → Spindle reverse rotation	Reverse (left–hand) tapping
G76	Cutting feed	Spindle indexing after dwell → Shift	Rapid traverse → Shift, Spindle start	Boring
G80	—	—	—	Cancel
G81	Cutting feed	—	Rapid traverse	Drilling, spot drilling
G82	Cutting feed	Dwell	Rapid traverse	Drilling, counter boring
G83	Intermittent feed	—	Rapid traverse	Deep hole drilling (peck drilling)
G84	Cutting feed	Spindle stop→ Spindle start in the reverse direction after dwell	Cutting feed → Dwell → Spindle reverse rotation	Tapping
G85	Cutting feed	—	Cutting feed	Boring

Table 4-3 GUD7 for cycle setting data (ISO dialect setting data), continued

G code	Drilling (-Z direction)	Processing at hole bottom	Retraction (+Z direction)	Applications
G86	Cutting feed	Spindle stop	Rapid traverse → Spindle start	Boring
G87	Spindle indexing → Shift → Rapid traverse → Shift → Spindle CW → Cutting feed	Spindle indexing after dwell → Shift	Rapid traverse → Shift → Spindle start	Back boring
G89	Cutting feed	Dwell	Cutting feed	Boring

Explanations

When using canned cycles the sequence of operations is generally carried out as described below:

- Operation 1
  Positioning in the XY plane using cutting feed or rapid traverse rate
- Operation 2
  Rapid traverse movement to level R
- Operation 3
  Machining to drilling depth Z
- Operation 4
  Operation at hole bottom
- Operation 5
  Retraction to R level at cutting feed or rapid traverse rate
- Operation 6
  Rapid retraction to positioning plane XY at rapid traverse rate

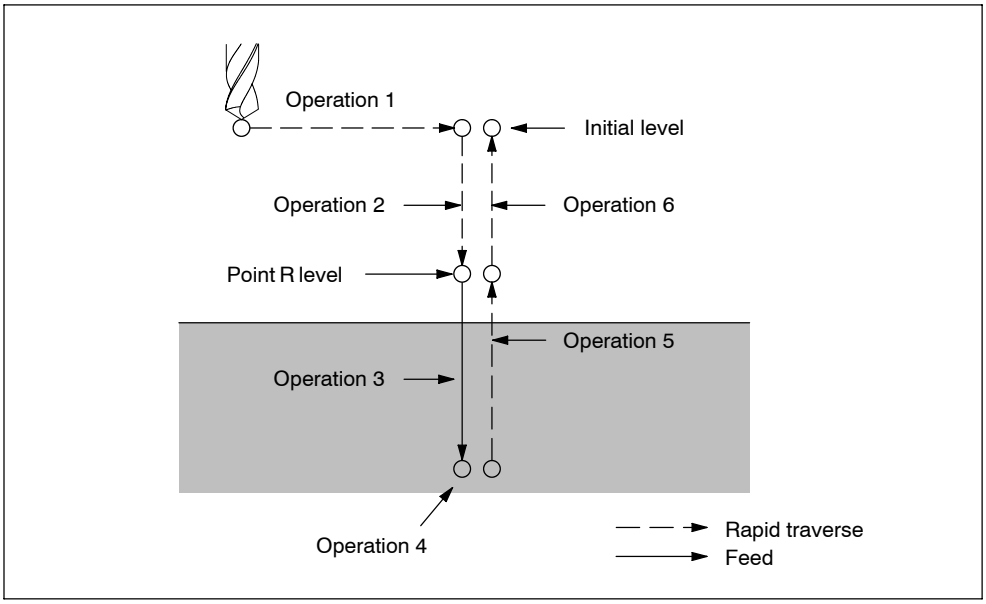


Fig. 4-2 Canned cycle operation sequence

In this chapter the term drilling will only be used to refer to operations implemented with canned cycles, although canned cycles encompass tapping and boring cycles as well as drilling cycles.

Plane definition

In the drilling cycles, it is generally assumed that the current workpiece coordinate system in which the machining operation is to be performed is defined by selecting plane G17, G18, or G19 and activating a programmable workpiece offset. The drilling axis is always the applicate of this coordinate system.
A tool length compensation must be selected before the cycle is called. Its effect is always perpendicular to the selected plane and remains active even after the end of the cycle.

Table 4-4 Positioning plane and drilling axis

G code	Positioning plane	Drilling axis
G17	Xp–Yp plane	Zp
G18	Zp–Xp plane	Yp
G19	Yp–Zp plane	Xp

Xp: X axis or an axis parallel to the X axis
Yp: Y axis or an axis parallel to the Y axis
Zp: Z axis or an axis parallel to the Z axis

Note

By applying GUD7 setting data _ZSFI[0], it can be decided whether the Z axis should always be used as the drilling axis. The Z axis always represents the drilling axis whenever _ZSFI[0] equals 1.

Canned cycle execution

The execution of canned cycles is determined as follows:

1. Cycle call
   G73, 74, 76, 81 through to 89
   depending on the desired machining
2. Data format G90/91

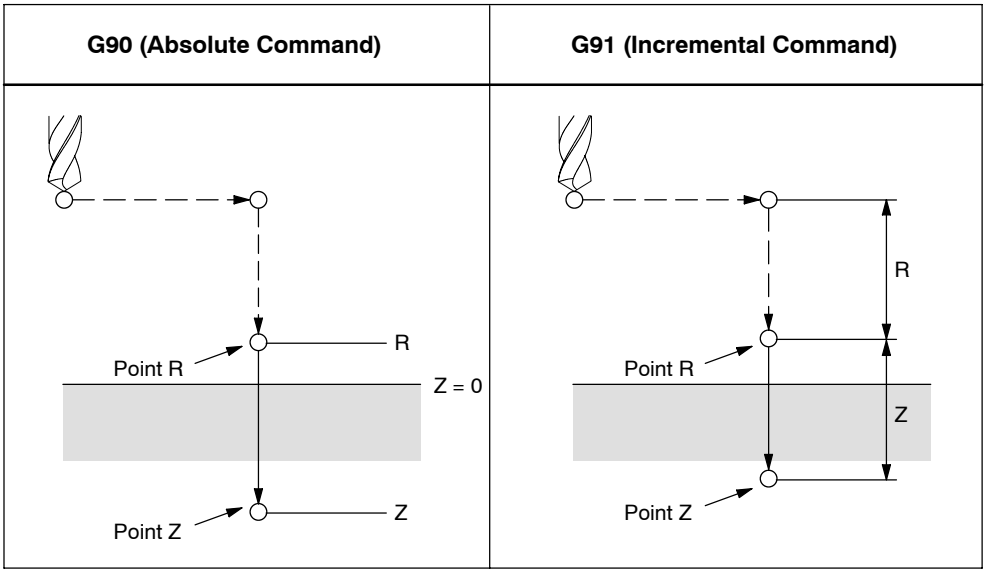


Fig. 4-3 Absolute / incremental command G90/G91

3. Drilling mode
   G73, G74, G76, and G81 through to G89 are modal G codes and remain effective until canceled. When they are applied, the drilling mode is the current state. The data is retained until modified or canceled, once the drilling data is determined within the drilling mode,
   At the beginning of canned cycles, determine all required drilling data. Only determine the data modifications whenever the canned cycles are being carried out.
4. Positioning / reference level (G98/G99)
   When using canned cycles, the retraction level for the Z axis is determined through G98/99. G98/G99 are modal G codes. G98 is usually set as power−on default.

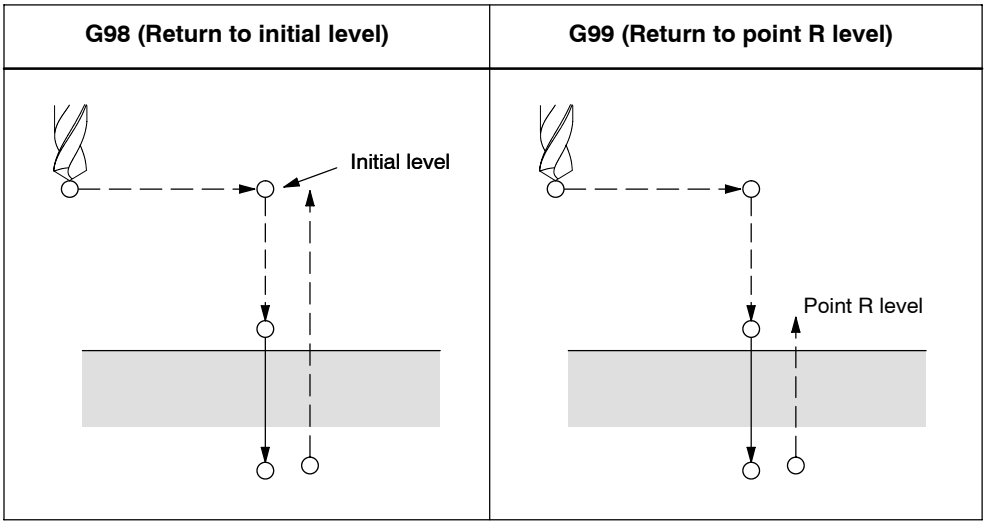


Fig. 4-4 Return point level (G98/G99)

Repetition

Specify the number of repeats in K in order to repeat the drilling for equally spaced holes. K only becomes effective in the block where it is specified. Specifying the first hole in absolute mode (G90) results in drilling at the same position. Therefore, specify K in incremental mode (G91).

Comments

A cycle call remains selected until it is cancelled through the G codes G80, G00, G01, G02, or G03, or through another cycle call.
Within the machining cycles the data specified at address Z, R, P, and Q function as self-retaining even after RESET operation. These data can only be changed by reprogramming or are cancelled using the G codes G80, G00, G01, G02, or G03.

Symbols in figures

Subsequent sections explain the individual canned cycles. Figures in these explanations use the following symbols:

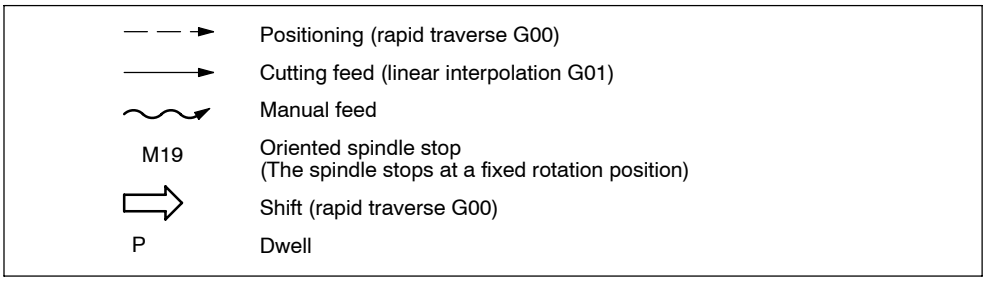


Fig. 4-5 Symbols in figures

4.1.2 High-speed peck drilling cycle (G73)

This cycle carries out high-speed peck drilling. It induces intermittent cutting feed to the bottom of a hole. Retract movements enables chip removal.

Format

G73 X.. Y... R... Q... F... K... ;

X,Y: Hole position
Z: Distance from point R to the bottom of the hole
R: Distance from the initial level to R level
Q: Depth of cut for each cutting
F: Cutting feedrate
K: Number of repeats

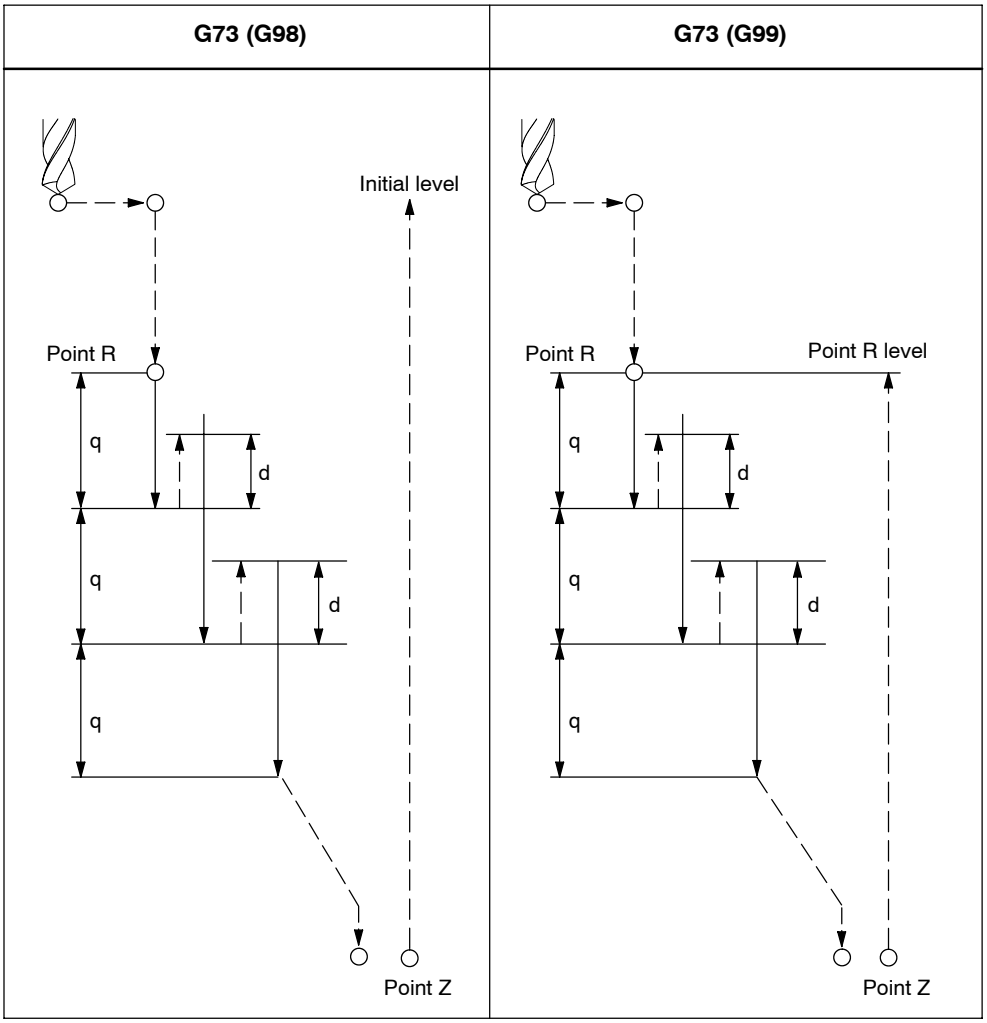


Fig. 4-6 High-speed peck drilling cycle (G73)

Explanations

When using cycle G73 retraction movement is performed in rapid traverse after each drilling operation. GUD _ZSFR[0] can be used to enter a safety clearance. The retraction amount for chipbreaking (d) is determined through GUD _ZSFR[1] as described below:

_ZSFR[1] > 0 Retraction amount as entered
_ZSFR[1] = 0 Retraction amount is always 1 mm or 1 inch with chipbreaking

Infeed is performed by using depth of cut for each cutting Q, which is incremented by the retraction amount d as of the 2nd infeed.

By means of this drilling cycle a rapid drilling infeed is accomplished. Removal of the drilling chips is facilitated through the retraction movement.

Note

If is need to specify 0 mm or 0 inch, set a smaller value than the movable resolution.

Example

```
M3 S1500;          Rotate spindle
G90 G0 Z100
G90 G99 G73 X200. Y-150. Z-100. R50. Q10. F150.;
                   Position, drill hole 1, and return to point R.
Y-500.;            Position, drill hole 2, and return to point R.
Y-700.;            Position, drill hole 3, and return to point R.
X950.;             Position, drill hole 4, and return to point R.
Y-500.;            Position, drill hole 5, and return to point R.
G98 Y-700.;        Position, drill hole 6, and return to the initial level.
G80;               Cancel canned cycle
G28 G91 X0 Y0 Z0; Return to the reference position return
M5;                Spindle stop
```

4.1.3 Fine boring cycle (G76)

Precise boring of a hole is accomplished through the fine boring cycle.

Format

G76 X... Y... R... Q... P... F... K... ;

X,Y: Hole position
Z_: Distance from point R to the bottom of the hole

R_: Distance from the initial level to point R level
Q_: Shift amount at the bottom of a hole
P_: Dwell time at the bottom of a hole
F_: Cutting feedrate
K_: Number of repeats

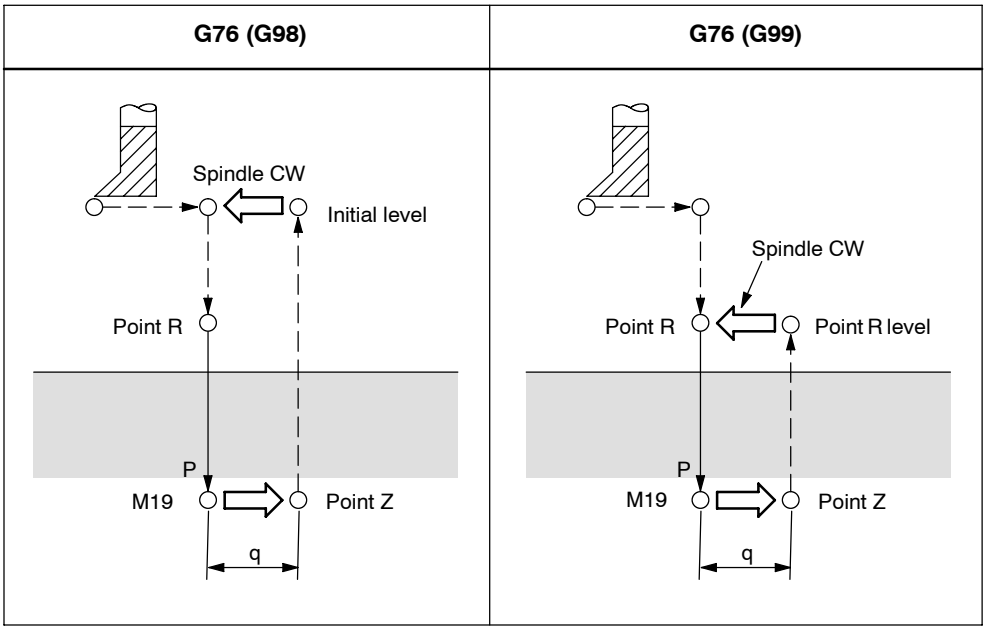


Fig. 4-7 Fine boring cycle (G76)

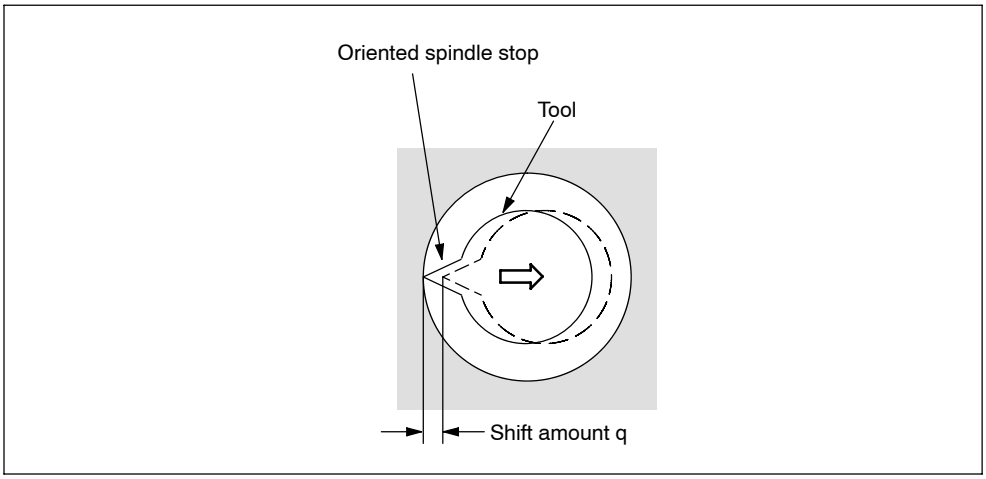


Fig. 4-8

Warning

Address Q is a modal value wich is retained within canned cycles. Special care has to be taken because it is also used as the depth of cut in cycles G73 and G83.

Explanations

The spindle is stopped at the fixed rotation position when the bottom of the hole has been reached.The tool is then moved in the direction opposite to the tool tip and retracted.

GUD _ZSFR[0] can be used to enter a safety clearance.
The lift-off path can be specified using _ZSFR[5].

	G17	G18	G19
_ZSFR[5] = 1	+X	+Z	+Y
_ZSFR[5] = 0 or 2	-X	-Z	-Y
_ZSFR[5] = 3	+Y	+X	+Z
_ZSFR[5] = 4	-Y	-X	-Z

The angle must be therefore be entered to GUD7 _ZSFR[2] such that the tool tip points in the reverse direction for lift-off path after the spindle stop.

Limitations

Axis switching

The canned cycle must be canceled before the drilling axis can be changed.

Boring

Boring is not carried out in a block that does not contain X, Y, Z, R, or any additional axes.

Q/R

By all means, specify a positive value at address Q. The sign is ignored if address Q is specified with a negative value. Q equal 0 is set whenever no lift-off amount is programmed. This leads to cycle execution without lift-off.

Cancel

G codes of group 01 (G00 to G03) and G76 must not be specified within a single block. Otherwise, G76 is canceled.

Tool offset

The tool offsets are ignored in the canned cycle mode.

Example

```
M3 S300;          Rotate spindle
G90 G0 Z100
G90 G99 G76 X200. Y-150. Z-100. R50. Q10. P1000 F120.;
                  Position, bore hole 1, then return to point R,
                  Stop at the bottom of the hole for 1 s.
Y-500.;           Position, drill hole 2, and return to point R.
Y-700.;           Position, drill hole 3, and return to point R.
X950.;            Position, drill hole 4, and return to point R.
Y-500.;           Position, drill hole 5, and return to point R.
G98 Y-700.;       Position, drill hole 6, and return to the initial level.
G80;              Cancel canned cycle
G28 G91 X0 Y0 Z0; Return to the reference position return
M5;               Spindle stop
```

4.1.4 Drilling cycle, spot drilling (G81)

Center drilling and spot drilling can be carried out by means of this cycle. After reaching drilling depth Z retraction movement is immediatly performed in rapid traverse rate.

Format

G81 X... Y... R... F... K... ;

X,Y: Hole position
Z: Distance from point R to the bottom of the hole
R: Distance from the initial level to R level
F: Cutting feedrate
K: Number of repeats

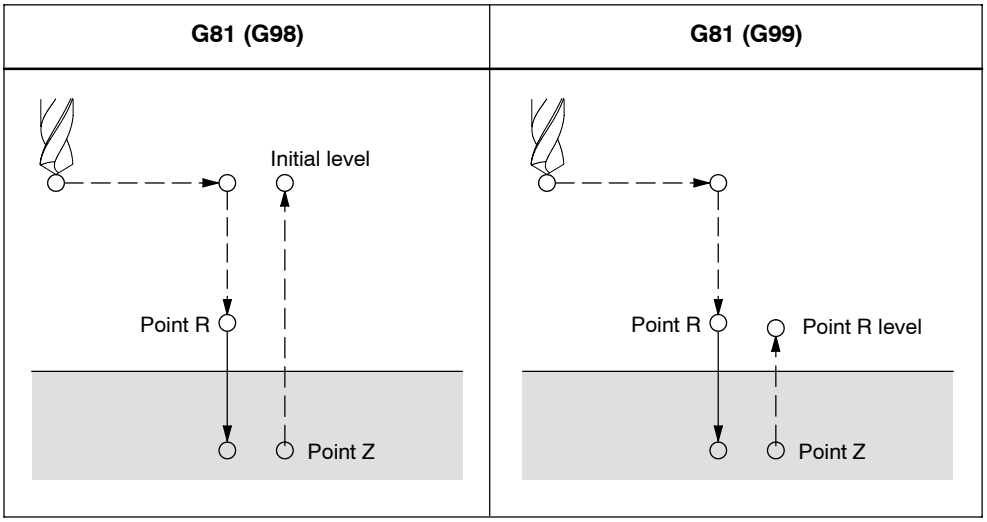


Fig. 4-9 Drilling cycle, spot drilling (G81)

Axis switching

Before the drilling axis can be changed, the canned cycle must be canceled.

Drilling

Drilling is not carried out in a block that does not contain X, Y, Z, R, or any additional axes.

Cancel

G codes of group 01 (G00 to G03) and G76 must not be specified within a single block. Otherwise, G76 is canceled.

Tool offset

Tool offsets are ignored in the canned cycle mode.

Example

```
M3 S1500;       Rotate spindle
G90 G0 Z100
G90 G99 G81 X200. Y-150. Z-100. R50. F120.;
                Position, drill hole 1, and return to point R.
Y-500.;         Position, drill hole 2, and return to point R.
Y-700.;         Position, drill hole 3, and return to point R.
X950.;          Position, drill hole 4, and return to point R.
Y-500.;         Position, drill hole 5, and return to point R.
G98 Y-700.;     Position, drill hole 6, and return to the initial level.
G80;            Cancel canned cycle
G28 G91 X0 Y0 Z0; Return to the reference position return
M5;             Spindle stop
```

4.1.5 Drilling cycle, counter boring cycle (G82)

Normal drilling can be carried out by means of this cycle. Upon reaching the drilling depth Z, a programmed dwell time is carried out after which the retraction movement is performed in rapid traverse.

Format

G82 X... Y... R... P... F... K... ;

X,Y: Hole position
Z: Distance from point R to the bottom of the hole
R: The distance from the initial level to R level
P: Dwell time at the bottom of a hole
F: Cutting feed rate
K: Number of repeats

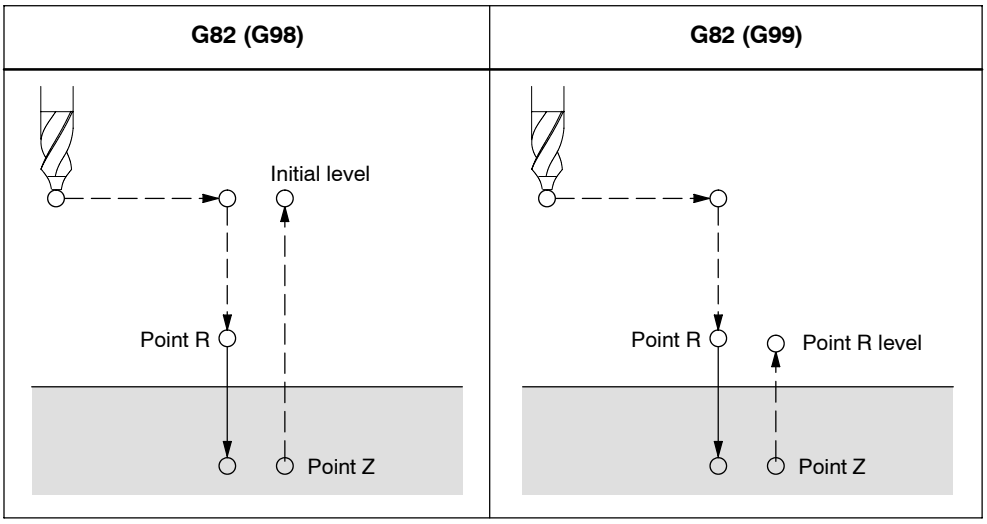


Fig. 4-10 Drilling cycle, counter boring cycle (G82)

Axis switching

The canned cycle must be canceled before the drilling axis can be changed.

Drilling

Drilling is not carried out in a block that does not contain X, Y, Z, R, or any other axes.

Cancel

G codes of group 01 (G00 to G03) and G82 must not be specified in a single block. Otherwise, G82 is canceled.

Tool offset

Tool offsets are ignored in the canned cycle mode.

Example

```
M3 S2000;          Rotate spindle
G90 G0 Z100
G90 G99 G82 X200. Y-150. Z-100. R50. P1000 F150.;
                   Position, drill hole 1, dwell for 1 s at the bottom of the hole,
                   and return to point R.
Y-500.;            Position, drill hole 2, and return to point R.
Y-700.;            Position, drill hole 3, and return to point R.
X950.;             Position, drill hole 4, and return to point R.
Y-500.;            Position, drill hole 5, and return to point R.
G98 Y-700.;        Position, drill hole 6, and return to the initial level.
G80;               Cancel canned cycle
G28 G91 X0 Y0 Z0; Return to the reference position return
M5;                Spindle stop
```

4.1.6 Peck drilling cycle (G83)

By means of this cycle peck drilling is performed. It is used for deep hole drilling with shaving extraction.

Format

G83 X... Y... R... Q... F... K... ;

X,Y: Hole position
Z: Distance from point R to the bottom of the hole
R: Distance from the initial level to R level
Q: Depth of cut for each cutting feed
F: Cutting feedrate
K: Number of repeats

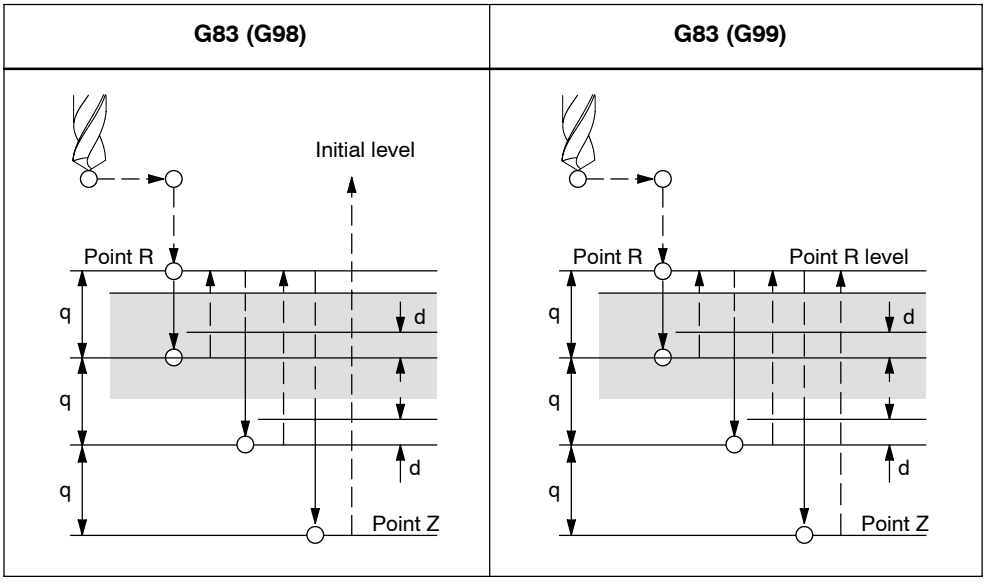


Fig. 4-11 Peck drilling cycle (G83)

Explanations

After reaching the programmed depth of cut for each cutting feed Q, retraction to reference level R is performed in rapid traverse. Approach movement for a renewed cut is again carried out in rapid traverse up to a distance (d) which is set to GUD7 _ZSFR[1]. Distance d and the depth of cut for each cutting feed Q are traversed with cutting feed. Specify Q incrementally implemented without sign.

Axis switching

The canned cycle must be canceled before the drilling axis can be changed.

Drilling

Drilling is not performed in a block that does not contain X, Y, Z, R, or any other axes.

Cancel

G codes of group 01 (G00 to G03) and G83 must not be specified in a single block. Otherwise, G83 is canceled.

Tool offset

Tool offsets are ignored in the canned cycle mode.

Example

```
M3 S2000;          Rotate spindle.
G90 G0 Z100
G90 G99 G83 X200. Y-150. Z-100. R50. Q10. F150.;
                   Position, drill hole 1, and return to point R.
Y-500.;            Position, drill hole 2, and return to point R.
Y-700.;            Position, drill hole 3, and return to point R.
X950.;             Position, drill hole 4, and return to point R.
Y-500.;            Position, drill hole 5, and return to point R.
G98 Y-700.;        Position, drill hole 6, and return to the initial level.
G80;               Cancel canned cycle
G28 G91 X0 Y0 Z0; Return to the reference position return
M5;                Spindle stop
```

After reaching the programmed depth of cut for each cutting feed Q, retraction to reference level R is performed in rapid traverse. Approach movement for a renewed cut is again carried out in rapid traverse up to a distance (d) which is set to GUD7_ZSFR[10]. Distance d and the depth of cut for each cutting feed Q are traversed with cutting feed. Specify Q incrementally implemented without sign.

Note

If _ZSFR[10]

- > 0 = value is used for anticipation distance ”d” (distance minimal 0.001)
- = 0 The anticipation distance is 30 mm, the value of the anticipation distance is always 0,6 mm. For larger drilling depths, the formula drilling depth/50 is used (maximum value 7 mm).

4.1.7 Boring cycle (G85)

Format

G85 X... Y... R... F... K... ;

X,Y: Hole position
Z: Distance from point R to the bottom of the hole
R: Distance from the initial level to R level
F: Cutting feed rate
K: Number of repeats

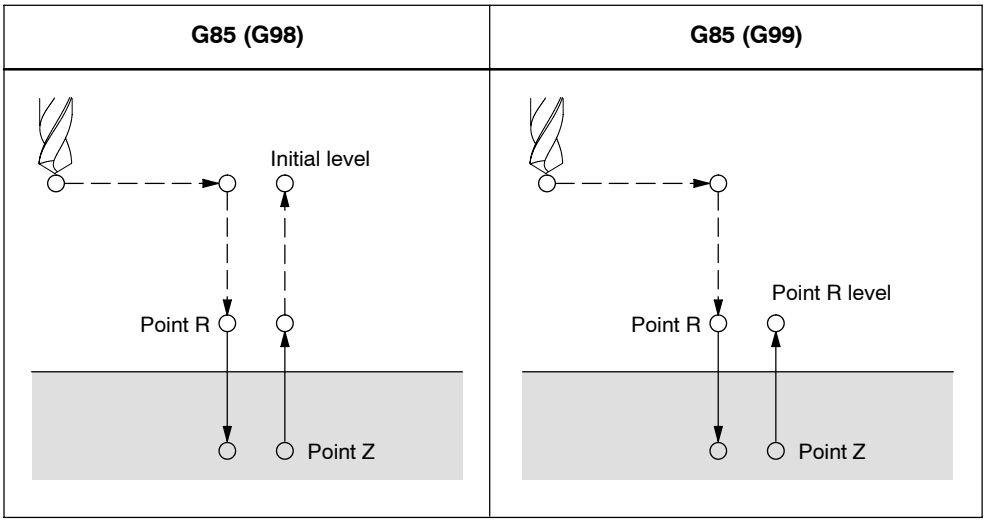


Fig. 4-12 Boring cycle (G85)

Explanations

Rapid traverse is carried out to point R after positioning along the X- and Y- axis. Drilling is performed from point R to point Z. After reaching point Z, cutting feed is performed back to point R.

Axis switching

Before the drilling axis can be changed the canned cycle must be canceled.

Drilling

Drilling is not performed in a block that does not contain X, Y, Z, R, or any other axes.

Cancel

G codes of group 01 (G00 to G03) and G85 must not be specified in a single block. Otherwise, G85 is canceled.

Tool offset

Tool offsets are ignored in the canned cycle mode.

Example

```
M3 S150;          Rotate spindle
G90 G0 Z100
G90 G99 G85 X200. Y-150. Z-100. R50. F150.;
                  Position, drill hole 1, and return to point R.
Y-500.;           Position, drill hole 2, and return to point R.
Y-700.;           Position, drill hole 3, and return to point R.
X950.;            Position, drill hole 4, and return to point R.
Y-500.;           Position, drill hole 5, and return to point R.
G98 Y-700.;       Position, drill hole 6, and return to the initial level.
G80;              Cancel canned cycle
G28 G91 X0 Y0 Z0; Return to the reference position return
M5;               Spindle stop
```

4.1.8 Boring cycle (G86)

Format

G86 X... Y... R... F... K... ;

X,Y: Hole position
Z: Distance from point R to the bottom of the hole
R: Distance from the initial level to point R
F: Cutting feed rate
K: Number of repeats

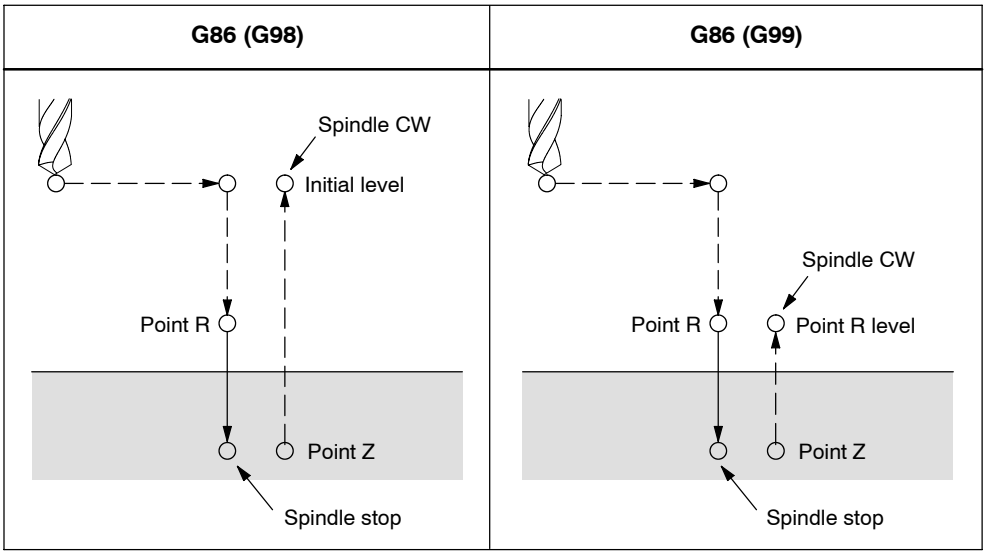


Fig. 4-13 Boring cycle (G86)

Explanations

Rapid traverse is performed to point R after positioning along the X and Y axes. Drilling is performed from point R to point Z. After the spindle is stopped at the bottom of the hole, the tool is retracted in rapid traverse.

Axis switching

The canned cycle must be canceled before the drilling axis can be changed.

Drilling

Drilling is not performed in a block that does not contain X, Y, Z, R, or any other axes.

Cancel

G codes of group 01 (G00 to G03) and G86 must not be specified in a single block. Otherwise, G86 is canceled.

Tool offset

Tool offsets are ignored in the canned cycle mode.

Example

```
M3 S1500;          Rotate spindle
G90 G0 Z100
G90 G99 G86 X200. Y-150. Z-100. R50. F150.;
                   Position, drill hole 1, and return to point R.
Y-500.;            Position, drill hole 2, and return to point R.
Y-700.;            Position, drill hole 3, and return to point R.
X950.;             Position, drill hole 4, and return to point R.
Y-500.;            Position, drill hole 5, and return to point R.
G98 Y-700.;        Position, drill hole 6, and return to the initial level.
G80;               Cancel canned cycle
G28 G91 X0 Y0 Z0; Return to the reference position return
M5;                Spindle stop
```

4.1.9 Boring cycle, back boring cycle (G87)

Accurate boring is performed by this cycle.

Format

G87 X... Y... R... Q... P... F... K... ;

X,Y: Hole position
Z: Distance from the bottom of the hole to point Z
R: Distance from the initial level to point R (the bottom of the hole)
Q: Tool shift amount
P: Dwell time
F: Cutting feed rate
K: Number of repeats

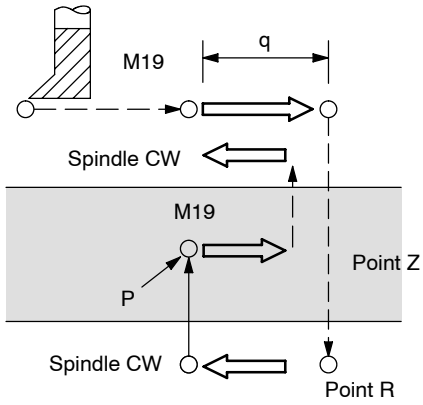
G87 (G98)	G87 (G99)
	**Not used**

Fig. 4-14 Boring cycle, back boring cycle (G87)

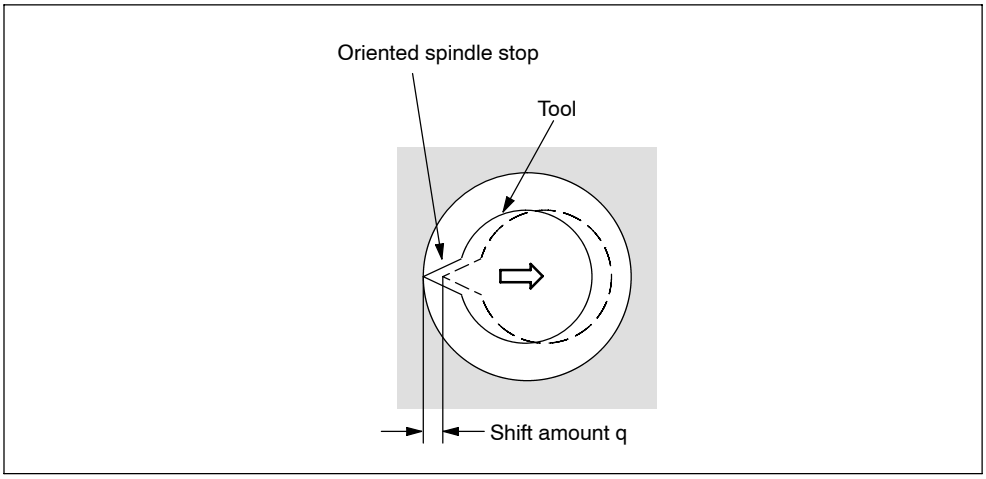


Fig. 4-15

Warning

Address Q (shift at the bottom of a hole) is a modal value wich is retained within canned cycles. Special care has to be taken because it is also used as the depth of cut in cycles G73 and G83.

Explanations

The spindle is stopped at the fixed rotation position after positioning along the X and Y axes. The tool is moved in the direction opposite the tip of the tool. Positioning (rapid traverse) is carried out to the bottom of the hole (point R).
Then the tool is shifted into the direction of the tool tip, and the spindle is rotated clockwise. Boring is carried out in the positive direction along the Z axis until point Z.
The spindle is stopped at the fixed rotation position again at point Z. The tool is then shifted into the direction opposite the tool tip, and the tool is shifted back to the initial level. Subsequently, the tool is shifted into the direction of the tool tip, and the spindle is rotated clockwise in order to proceed to the next block operation.
To enter a safety clearance, GUD _ZSFR[0] can be applied.
The lift-off path can be specified using _ZSFR[5].

	G17	G18	G19
_ZSFR[5] = 1	+X	+Z	+Y
_ZSFR[5] = 0 or 2	-X	-Z	-Y
_ZSFR[5] = 3	+Y	+X	+Z
_ZSFR[5] = 4	-Y	-X	-Z

G17, lift-off path in -X
G18, lift-off path in -Z
G19, lift-off path in -Y

Therefore, the angle has to be entered to GUD7 _ZSFR[2] in such a way that the tool tip points in the reverse direction for lift-path after the spindle has stopped.
Example:
If plane G17 is activated, the tool tip has to point into the +X direction.

Axis switching

The canned cycle must be canceled before the drilling axis can be changed.

Boring

Boring is not performed within a block that does not contain X, Y, Z, R, or any additional axes.

Q/R

By all means, specify a positive value at address Q. The sign is ignored if address Q is specified with a negative value. Q equal 0 is set whenever no lift-off amount is programmed. This leads to cycle execution without lift-off.

Cancel

G codes of group 01 (G00 to G03) and G87 must not be specified in a single block. Otherwise, G87 is canceled.

Tool offset

Tool offsets are ignored in the canned cycle mode.

Example

```
M3 S400;                Rotate spindle
G90 G0 Z100
G90 G87 X200. Y-150. Z-100. R150. Q3. P1000 F150.;
                        Position, bore hole 1,
                        orient at the initial level, then shift by 3 mm,
                        stop at point Z for 1 s
Y-500.;                 Position, drill hole 2.
Y-700.;                 Position, drill hole 3.
X950.;                  Position, drill hole 4.
Y-500.;                 Position, drill hole 5.
Y-700.;                 Position, drill hole 6
G80;                    Cancel canned cycle
G28 G91 X0 Y0 Z0;       Return to the reference position return
M5;                     Spindle stop
```

4.1.10 Drilling cycle (G89), retract using G01

Format

G89 X... Y... R... P... F... K... ;

X,Y: Hole position
Z: Distance from point R to the bottom of the hole
R: Distance from the initial level to point R
P: Dwell time at the bottom of a hole
F: Cutting feed rate
K: Number of repeats

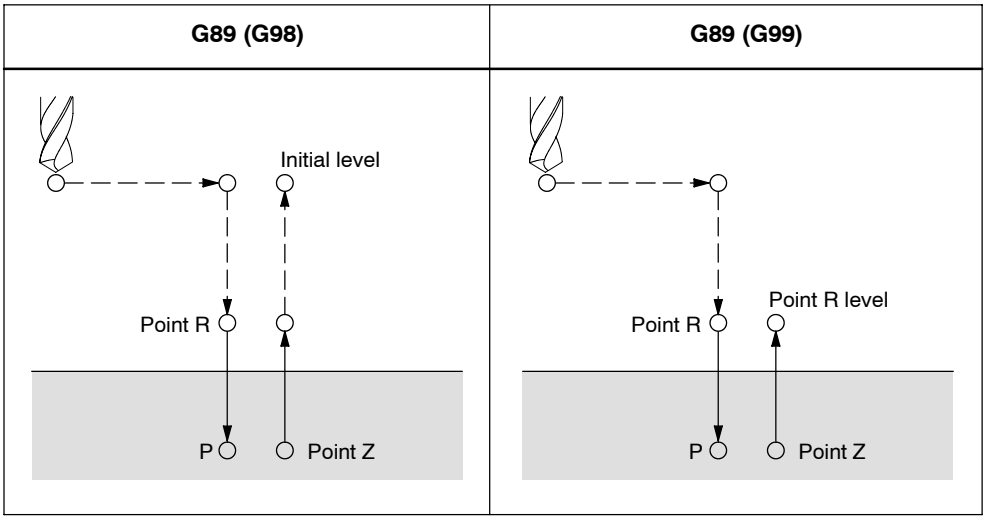


Fig. 4-16 Boring cycle (G89)

Explanations

This cycle is almost the same as G86 except that this cycle performs a dwell at the bottom of the hole.
Use a miscellaneous function (M code) to rotate the spindle before specifying G89.

Axis switching

The canned cycle must be canceled before the drilling axis can be changed.

Drilling

Drilling is not performed in a block that does not contain X, Y, Z, R, or any other axes.

Cancel

G codes of group 01 (G00 to G03) and G89 must not be specified in a single block. Otherwise, G89 is canceled.

Tool offset

Tool offsets are ignored in the canned cycle mode.

Example

```
M3 S150;          Rotate spindle
G90 G0 Z100
G90 G99 G89 X200. Y-150. Z-100. R50. P1000 F150.;
                  Position, drill hole 1, return to point R
                  then stop at the bottom of the hole for 1 s.
Y-500.;           Position, drill hole 2, and return to point R.
Y-700.;           Position, drill hole 3, and return to point R.
X950.;            Position, drill hole 4, and return to point R.
Y-500.;           Position, drill hole 5, and return to point R.
G98 Y-700.;       Position, drill hole 6, and return to the initial level.
G80;              Cancel canned cycle
G28 G91 X0 Y0 Z0; Return to the reference position return
M5;               Spindle stop
```

4.1.11 Rigid tapping cycle (G84)

When the spindle motor is controlled in rigid mode as if it were a servo motor, a tapping cycle can be sped up.

Format

G84 X... Y... Z... R... P... F... K... ;

X,Y: Hole position
Z: Distance from point R to bottom of the hole
R: Distance from the initial level to R level
P: Dwell time at bottom of the hole and at point R when a return is made
F: Cutting feedrate
K: Number of repeats (if required)

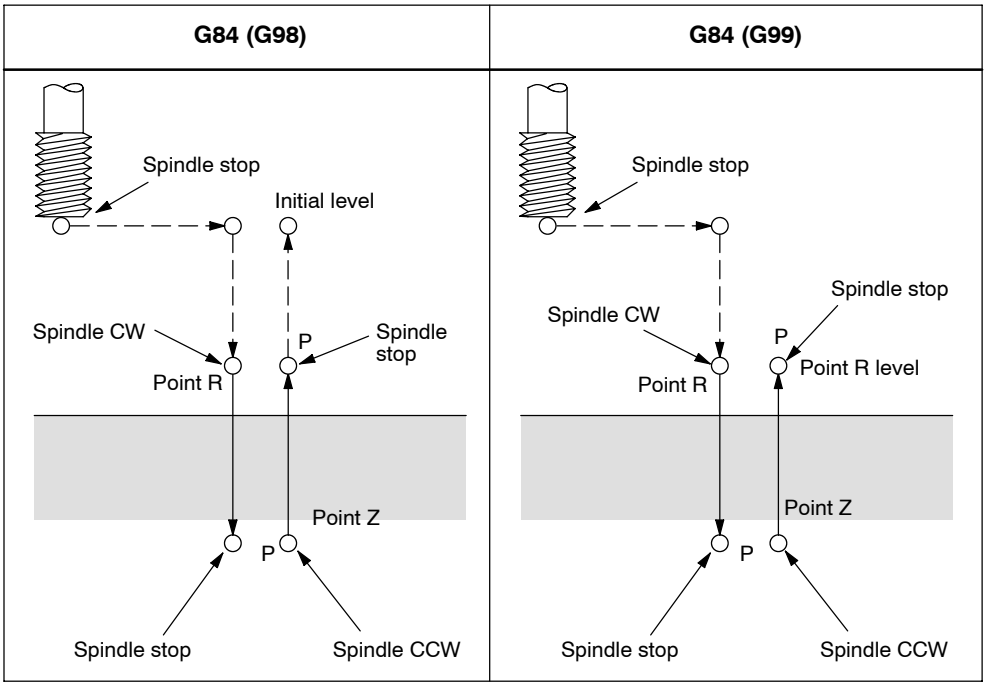


Fig. 4-17 Rigid tapping (G84)

Explanations

Rapid traverse is carried out to point R after positioning along the X and Y axes. Tapping is carried out from point R to point Z. The spindle is stopped, and a dwell is performed once tapping has been completed. Then the spindle is rotated in reverse direction. The tool is retracted to point R, and the spindle is stopped. Subsequently, rapid traverse to the initial level is carried out. The feedrate override and the spindle override are supposed to be 100% while tapping is being carried out.
Yet the rotation speed during retraction can be controlled through GUD _ZSFI[2].
Example: _ZSFI[2]=120, the retraction is performed at 120% of the tapping speed.

Thread lead

The thread lead is obtained from the expression 'feedrate spindle speed' in the feed-per-minute mode. The thread lead equals the feedrate speed within the speed-per-revolution mode.

Tool length compensation

The offset is applied at the time of positioning to point R if a tool length compensation (G43, G44, or G49) is determined in the canned cycle.

Axis switching

The canned cycle must be canceled before the drilling axis can be changed. An alarm is issued if the drilling axis is changed in rigid mode.

S command

An alarm is issued if a speed higher than the maximum speed for the gear being used is specified.

F command

An alarm is issued if a value exceeding the upper limit of cutting feedrate is specified.

Unit of F command

	Metric input	Inch input	Remarks
G94	1 mm/min	0.01 inch/min	Decimal point programming allowed
G95	0.01 mm/rev	0.0001 inch/rev	Decimal point programming allowed

Cancel

G codes of group 01 (G00 to G03) and G84 must not be specified in a single block. Otherwise, G84 is canceled.

Tool offset

Tool offsets are ignored in the canned cycle mode.

Example

Z axis feedrate 1000 mm/min
Spindle speed 1000 rpm
Thread lead 1.0 mm

<Programming of feed per minute>

```
S1000 M3;
G94;                          Feed-per-minute
G00 X100.0 Y100.0;            Positioning
G84 Z-50.0 R-10.0 F1000;      Rigid tapping
```

<Programming of feed per revolution>

```
G95;                          Feed-per-revolution
G00 X100.0 Y100.0;            Positioning
G84 Z-50.0 R-10.0 F1.0;       Rigid tapping
```

4.1.12 Left-handed rigid tapping cycle (G74)

When the spindle motor is controlled in rigid mode as if it were a servo motor, tapping cycles can be sped up.

Format

G74 X... Y... Z... R... P... F... K... ;

X,Y: Hole position
Z: The distance from point R to the bottom of the hole
R: The distance from the initial level to point R
P: Dwell time at the bottom of the hole and at point R when return is made.
F: Cutting feedrate
K: Number of repeats (if required)

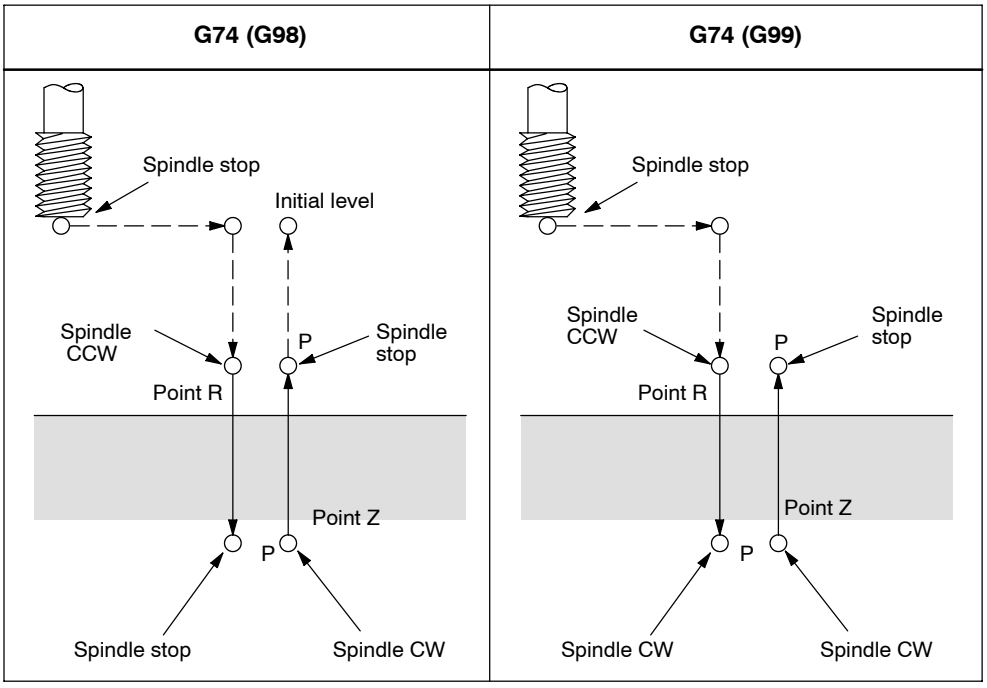


Fig. 4-18 Left-handed rigid tapping cycle (G74)

Explanations

Rapid traverse is performed to point R after positioning along the X and Y axes. Tapping is carried out from point R to point Z. The spindle is stopped and a dwell is performed once tapping has been completed. Subsequently, the spindle is rotated in the normal direction. The tool is retracted to point R, and the spindle is stopped. Rapid traverse to the initial level is then carried out.
The feedrate override and the spindle override are supposed to be 100% while tapping is being carried out.
The speed of rotation, however, can be controlled through GUD _ZSFI[2] during retraction. Example: _ZSFI[2]=120, the retraction takes place with 120% of the tapping speed.

Thread lead

The thread lead is obtained from the expression 'feedrate spindle speed' in feed-per-minute mode. The thread lead equals the feedrate speed when in feed-per-revolution mode.

Tool length compensation

The offset is applied at the time of positioning to point R whenever a tool length compensation (G43, G44, or G49) is specified in the canned cycle.

Axis switching

The canned cycle must always be canceled before the drilling axis can be changed. An alarm is issued if the drilling axis is changed into rigid mode.

S command

An alarm is issued if a speed that is higher than the maximum speed for the gear currently in use is specified.

F command

An alarm is issued if a value overshooting the upper limit of cutting feedrate is specified.

Unit of F command

	Metric input	Inch input	Remarks
G94	1 mm/min	0.01 inch/min	Decimal point programming allowed
G95	0.01 mm/rev	0.0001 inch/rev	Decimal point programming allowed

Cancel

A G code of the 01 group (G00 to G03) and G74 should not be specified in a single block or else G74 will be canceled.

Tool offset

Tool offsets are ignored In the canned cycle mode.

Example

Z axis feedrate 1000 mm/min
Spindle speed 1000 rpm
Thread lead 1.0 mm

<Programming of feed per minute>

```
S1000 M4
G94;                          Feed-per-minute
G00 X100.0 Y100.0;            Positioning
G74 Z-50.0 R-10.0 F1000;      Rigid tapping
```

<Programming of feed per revolution>

```
G95;                          Feed-per-revolution
G00 X100.0 Y100.0;            Positioning
G74 Z-50.0 R-10.0 F1.0;       Rigid tapping
```

4.1.13 Peck tapping cycle (G84 or G74)

Due to chips stuck to the tool or increased resistance to cutting, the tapping a deep hole in rigid tapping mode could be difficult. The peck rigid tapping cycle is useful when this is the case.
Cutting is carried out several times in this cycle until the bottom of the hole is reached. For this, two peck tapping cycles are available: High-speed peck tapping cycle (deep hole tapping with chip-breaking) as well as the standard peck tapping cycle (deep hole tapping with swarf removal.
By using GUD7 and setting data _ZSFI[1], these cycles are selected as follows:
_ZSFI[1] = 2: High-speed peck tapping cycle
_ZSFI[1] = 3: Standard peck tapping cycle

Format

G84 (or G74) X... Y... Z... R... P... Q... F... K... ;

X,Y: Hole position
Z: Distance from point R to the bottom of the hole
R: The distance from the initial level to point R level
P: Dwell time at the bottom of the hole and at point R when a return is made
Q: Depth of cut for each cutting feed
F: The cutting feedrate
K: Number of repeats

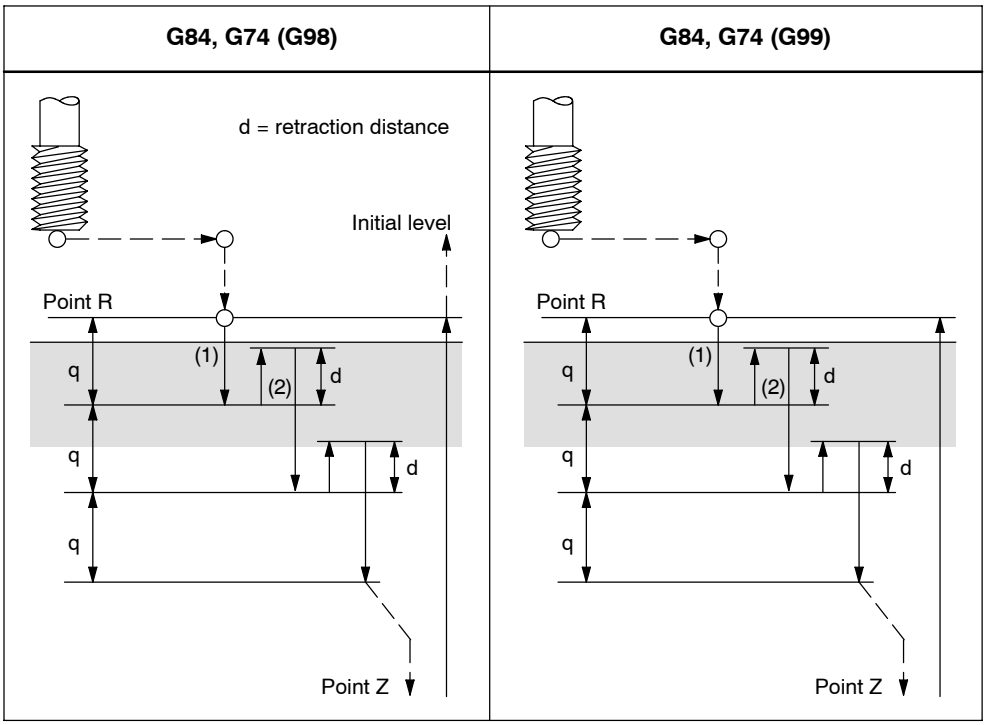


Fig. 4-19 High-speed peck tapping cycle (GUD7 _ZSFI[1] = 2)

1. The tool can operates at a normal cutting feedrate. Here the normal time constant is applied.
2. The retraction can be overridden. The retraction speed set to GUD7 _ZSFI[2] is applied in this case.

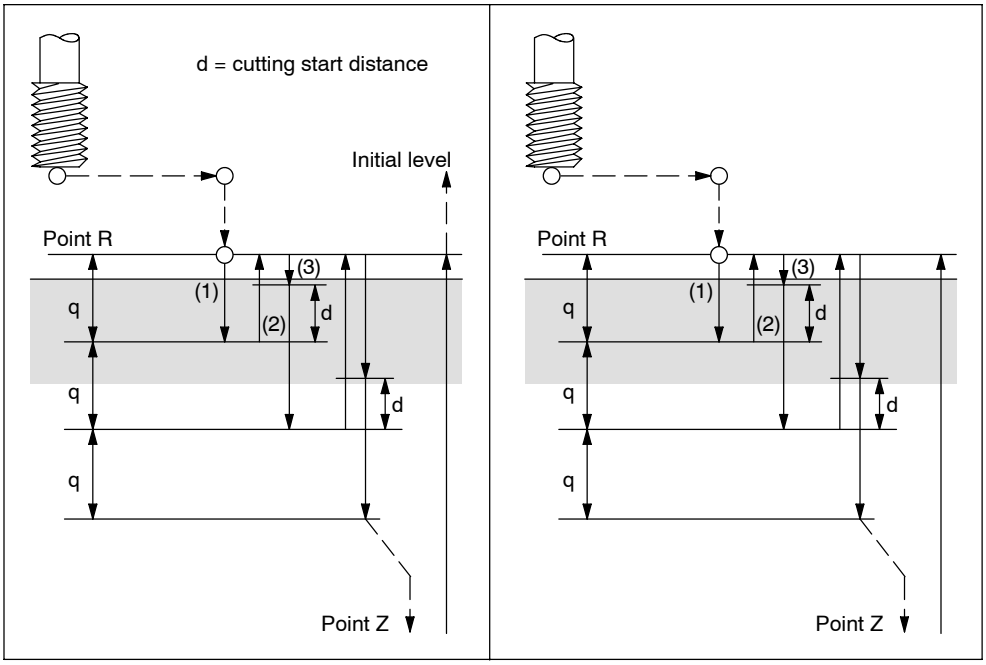


Fig. 4-20 Peck tapping cycle (GUD7 _ZSFI[1] = 3)

1. The tool can operate at a normal cutting feedrate. Here the normal time constant is applied.
2. The retraction can be overridden. The retraction speed set to GUD7 _ZSFI[2] is applied in this case.
3. The retraction can be overridden. Here the normal time constant is applied.

   An in-position check is carried out at the end of each operation of 1. and 2. in the peck tapping cycle during a rigid tapping cycle.

Explanation

High-speed peck tapping cycle

Rapid traverse is carried out to point R after positioning along the X and Y axes. Cutting is carried out from point R,with depth Q (depth of cut for each cutting feed). Subsequently, the tool is retracted by the distance d. Whether retraction is overridden or not is specified by a value other than 100% set to GUD7 _ZSFI[2]. The spindle is stopped once point Z is reached and then rotated in the reverse direction for retraction. The retraction distance d is to be set in GUD7 _ZSFR[1].

Note

I fthe value 0 is set into _ZSFR[1], 1 mm or 1 inch is used for the retraction distance as default.
If it is need to specify 0 mm or 0 inch, set a smaller value than the movable resolution.

Peck tapping cycle

Rapid traverse is performed to R level after positioning along the X and Y axes. Cutting is performed from point R with depth Q (depth of cut for each cutting feed). Subsequently a return is carried out to point R. Whether retraction is overridden or not is specified by a value other than 100% set to GUD7 _ZSFI[2]. Moving the cutting feedrate F is carried out from point R to a position distance d from the end point of the last cutting. This is where the cutting is restarted.
The spindle is stopped once point Z is reached, and, subsequently, rotated in the reverse direction for retraction.
Set d (distance to the point where the cutting is started) in GUD7 _ZSFR[1].

Note

I fthe value 0 is set into _ZSFR[1], 1 mm or 1 inch is used for the retraction distance as default.
If it is need to specify 0 mm or 0 inch, set a smaller value than the movable resolution.

Axis switching

The canned cycle must be canceled before the drilling axis can be changed. An alarm is issued whenever the drilling axis is changed in rigid mode.

S command

An alarm is issued whenever a speed higher than the maximum speed for the gear in use is specified.

F command

An alarm is issued, if a value overshooting the upper limit of the cutting feedrate is specified.

Unit of F command

	Metric input	Inch input	Remarks
G94	1 mm/min	0.01 inch/min	Decimal point programming allowed
G95	0.01 mm/rev	0.0001 inch/rev	Decimal point programming allowed

Cancel

A G code of the 01 group (G00 to G03) and G74/G84 should not be specified in a single block or else G74/G84 will be canceled.

Tool offset

Tool offsets are ignored in the canned cycle mode.

4.1.14 Canned cycle cancel (G80)

G80 cancels canned cycles.

Format

G80;

Explanations

The values of point R and point Z are cleared, all canned cycles are canceled and normal operation is performed. In addition, the values of all addresses programmed with drilling cycles are cleared.

4.1.15 Program example using tool length offset and canned cycles

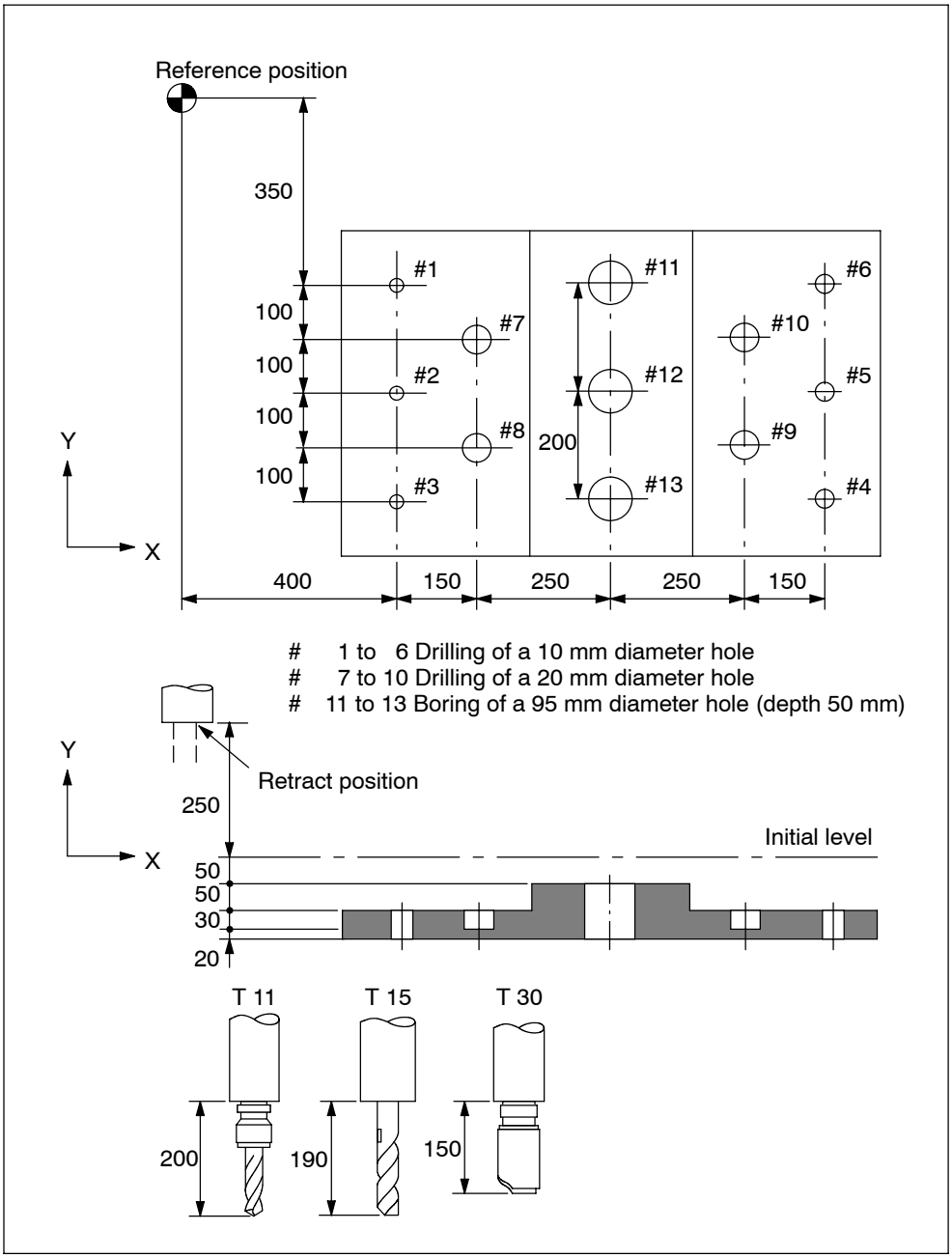


Fig. 4-21 Program example (drilling cycle)

Offset value +200.0 is set in offset No.11, +190.0 is set in offset No.15, and +150.0 is set in offset No.30

Program example

```
;
N001 G49                          ; Tool length offset cancel
N002 G10 L10 P11 R200.            ; Set tool offset 11 to +200.
N003 G10 L10 P15 R190.            ; Set tool offset 15 to +190.
N004 G10 L10 P30 R150.            ; Set tool offset 30 to +150.
N005 G92 X0 Y0 Z0                 ; Coordinate setting at reference
                                  ; position
N006 G90 G00 Z250.0 T11 M6        ; Tool change
N007 G43 Z0 H11;                  ; Initial level, tool length offset
N008 S30 M3                       ; Spindle start
N009 g99 G81 X400.0 Y-350.0       ; Positioning, then #1 drilling
Z-153.0 R-97.0 F1200;
N010 Y-550.0                      ; Positioning, then #2 drilling and
                                  ; point R level return
N011 G98 Y-750.0                  ; Positioning, then #3 drilling and
                                  ; initial level return
N012 G99 X1200.0                  ; Positioning, then #4 drilling and
                                  ; point R level return
N013 Y-550.0                      ; Positioning, then #5 drilling and
                                  ; point R level return
N014 G98 Y-350.0                  ; Positioning, then #6 drilling and
                                  ; initial level return
N015 G00 X0 Y0 M5                 ; Reference position return,
                                  ; spindle stop
N016 G49 Z250.0 T15 M6            ; Tool length offset cancel, tool
                                  ; change
N017 G43 Z0 H15                   ; Initial level, tool length offset
N018 S20 M3                       ; Spindle start
N019 G99 G82 X550.0 Y-450.0       ; Positioning, then #7 drilling,
Z-130.0 R-97.0 P300 F700          ; point R level return
N020 G98 Y-650.0                  ; Positioning, then #8 drilling,
                                  ; initial level return
N021 G99 X1050.0                  ; Positioning, then #9 drilling,
                                  ; point R level return
N022 G98 Y-450.0                  ; Positioning, then #10 drilling,
                                  ; initial level return
```

```
N023 G00 X0 Y0 M5                ; Reference position return,
                                 ; spindle stop
N024 G49 Z250.0 T30 M6           ; Tool length offset cancel, tool
                                 ; change
N025 G43 Z0 H30                  ; Initial level, tool length offset
N026 S10 M3                      ; Spindle start
N027 G85 G99 X800.0 Y-350.0      ; Positioning, then #11 drilling,
Z-153.0 R47.0 F500               ; point R level return
N028 G91 Y-200.0 K2              ; Positioning, then #12, 13 dril-
                                 ; ling. point R level return
N029 G28 X0 Y0 M5                ; Reference position return,
                                 ; spindle stop
N030 G49 Z0                      ; Tool length offset cancel
N031 M30                         ; Program End
```

4.1.16 Multiple threads with G33

Syntax G33 X.. Z.. F.. Q.. is used to program multiple threads in ISO dialect T and M mode, whereby:

X.. Z.. = Thread end position
F.. = Lead
Q.. = Initial angle

Threads with offset slides are programmed by entering starting points, which are offset from one another, in set G33. The starting point offset is entered at address “Q” as an absolute angular position. The corresponding setting data ($SD_THREAD_START_ANGLE) is changed accordingly.

Example: Q45000 means: Start offset 45.000 degrees

Range of values: 0.0000 to 359.999 degrees

The initial angle must always be programmed as an integer value. The input resolution for angular data is 0.001 degrees.

Example:

```
N200 X50 Z80 G01 F.8 G95 S500 M3
N300 G33 Z40 F2 Q180000
```

This produces a thread with a lead of 2 mm and a starting point offset of 180 degrees.

4.1.17 Threads with variable lead (G34)

Syntax G34 X.. Z.. F.. K.. is used to program threads with variable lead in ISO dialect T and M mode, whereby

X.. Z..	=	Thread end position
F..	=	Lead
K..	=	Lead increase (positive value)/ lead decrease (negative value)

G34 is used to increment or decrement the lead by the value programmed at address K on each spindle revolution.

Example:

```
N200 X50 Z80 G01 F.8 G95 S500 M3
N300 G91 G34 Z25.5 F2 K0.1
```

The programmed distance of 25.5 mm corresponds to 10 spindle revolutions.

4.2 Programmable data input (G10)

4.2.1 Changing of tool offset value

Existing tool offsets can be overwritten by using the G10. New tool offsets, however, cannot be created.

Format

G10 L10 P... R... ; Tool length compensation, geometry
G10 L11 P... R... ; Tool length compensation, wear
G10 L12 P... R... ; Tool radius compensation, geometry
G10 L13 P... R... ; Tool radius compensation, wear

P: Number of the compensation memory
R: Specifies the value

L1 can be programmed instead of L11.

Relevant machine data

Machine data 20382 $MC_TOOL_CORR_MOVE_MODE defines whether the compensation is applied in the block containing the selection or the next time the axis is programmed.
Machine data 20270 $MC_CUTTING_EDGE_DEFAULT = 0 defines that no tool length compensation is active initially on a tool change.

Setting data $SC_TOOL_LENGTH_CONST must contain the value 17 for the assignment of tool length offsets to geometry axes to be independent of the plane selection. Length 1 is then always assigned to the Z axis.

4.2.2 Setting the workpiece coordinate system shift data

With the commands of “G10 P00 X (U) ··· Y (V) ··· Z (W) ··· ;”, it is possible to write and update the workpiece coordinate system shift data using a part program. If an address is omitted in the designation of data input block, the offset amounts for the omitted addresses remain unchanged.

X, Z, C	:	Absolute or incremental setting data of the workpiece coordinate system shift amount
U, W, H	:	Incremental setting data of the workpiece coordinate system shift amount

4.3 Subprogram call up function (M98, M99)

This function can be used when subprograms are stored in the part program memory. Subprograms registered to the memory with program numbers assigned can be called up and executed as many times as required.

The created subprograms should be stored in the part program memory before they are called up.

Commands

The M codes indicated in Table 4-5 are used.

Table 4-5 Subprogram call M code

M code	Function
M98	Subprogram call up
M99	End of subprogram

Subprogram call (M98)

- M98 P nnn mmmm
  m: Program number (max. 4 digits)
  n: Number of repetitions (max. 3 digits)
- For example, if M98 P21 is programmed, the part program memory is searched for program name 21.mpf and the subprogram is executed once. To execute the subprogram 3 times, M98 P30021 needs to be programmed. If the specified program number is not found, an alarm occurs.
- Nesting of subprograms is possible - the allowable nesting level is four. If the nesting level exceeds this limit, an alarm occurs.

End of subprogram code (M99)

If M99 Pxxxx is programmed, execution resumes at block number xxxx on the return jump to the main program. The system initially searches forward for the block number (from the subprogram call towards the end of the program). If a matching block number is not found, the part program is then searched backwards (towards the head of the program).

If M99 is specified in a main program, the program returns to the beginning of that main program and the program is repeatedly executed.

4.4 Eight-digit program number

An eight-digit program number selection is activated with $MC_EXTERN_FUNCTION_MASK, bit6=1. This function affects M98, G65/66 and M96.

y: Number of program runs
x: Program number

Subroutine call M98

$MC_EXTERN_FUNCTION_MASK, bit6 = 0
M98 Pyyyyxxxx or
M98 Pxxxx Lyyyy
Program number max. 4-digit
Always add 0s to extend program number to 4 digits

E.g.:	M98 P20012	calls 0012.mpf 2 runs
	M98 P123 L2	calls 0123.mpf 2 runs

$MC_EXTERN_FUNCTION_MASK, bit6 = 1
M98 Pxxxxxxxx Lyyyy
No zeros are added, even if the program number has less than 4 digits.
The number of runs and program number cannot be programmed in P(Pyyyyxxxxx),
the number of runs must always be programmed with L!

e.g.:	M98 P123	calls 123.mpf 1 run
	M98 P20012	calls 20012.mpf 1 run,
	Important: No longer compatible with ISO Dialect Original	
	M98 P12345 L2	calls 12345.mpf 2 runs

Modal and block-by-block macro G65/G66

$MC_EXTERN_FUNCTION_MASK, bit6 = 0
G65 Pxxxx Lyyyy
Always add 0s to extend program number to 4 digits. Program number with more than 4 digits generates an alarm.

$MC_EXTERN_FUNCTION_MASK, bit6 = 1
M65 Pxxxx Lyyyy
No zeros are added, even if the program number has less than 4 digits. Program number with more than 8 digits generates an alarm.

Interrupt M96

Does not work with SINUMERIK 802D sl.

$MC_EXTERN_FUNCTION_MASK, bit6 = 0
M96 Pxxxx
Always add 0s to extend program number to 4 digits

$MC_EXTERN_FUNCTION_MASK, bit6 = 1
M96 Pxxxx
No zeros are added, even if the program number has less than 4 digits. Program number with more than 8 digits generates an alarm.

4.5 Polar coordinate command (G15, G16)

Using polar coordinate command it is possible to program the end point coordinate value in radius and angle. Any dimension word between G16 and G15 command is interpreted as the polar coordinate values for radius and angle in the current plane. The first axis of the plane represents the polar radius, while the second axis represents the polar angle.

Format

G17 (G18, G19) G90 (G91) G16;	Polar coordinate command ON
G90 (G91) X... Y... Z...	Polar coordinate command
... ;	
... ;	
G15;	Polar coordinate command CANCEL

G16:	Polar coordinate command
G15:	Polar coordinate command CANCEL
G17, G18, G19:	Plane selection
G90:	The pole is at the workpiece zero
G91:	The pole is at the current position
X, Y, Z:	First axis: radius of polar coordinate Second axis: angle of polar coordinate

Note

If the pole is moved from the current position to the workpiece zero, the radius is calculated as the distance from the current position to the workpiece zero.

Example

```
N5 G17 G90 X0 Y0;
N10 G16 X100. Y45.; Polar coordinates ON, pole is workpiece zero,
                    position X 70,711 Y 70,711 in Cartesian coordi
                    nate system
N15 G91 X100 Y0;    Pole is current position,
                    i.e. position X 170,711 Y 70,711
N20 G90 Y90.;       No X in block, pole is at workpiece zero,
                    Radius = SORT(X*X +Y*Y) = 184,776
G15;
```

The polar radius is always traversed as an absolute value while the polar angle can be interpreted as an absolute or incremental value.

4.6 Polar coordinate interpolation (G12.1, G13.1)

An interpolation between a rotary axis and a linear axis in the machining plane is switched on and off through G12.1 and G13.1. A further linear axis is perpendicular to this plane.
Linear or circular interpolation using coordinates in a Cartesian coordinate system is applied in order to program a linear axis together with a rotary axis (virtual axis).

This function corresponds to the TRANSMIT function within Siemens mode.

Note

For a detailed description of the TRANSMIT function see “SINUMERIK 840D/810D(CCU2)/FM” NC Functional Description, Extended Functions” chapter “Kinematic Transformation (M1)” and “SINUMERIK 840D/810D/FM–NC Programming Guide Production Scheduling (PGA)” chapter “Transformation”.

Format

```
G12.1;      Polar coordinate interpolation mode ON
...
...
G13.1;      Polar coordinate interpolation mode cancel
```

Caution

When specifying G12.1, the plane (G17, G18, G19) which has been used previously is canceled.
NC Reset will cancel the polar coordinate interpolation mode and re–establish the previously selected plane.

Possible G codes in the polar coordinate interpolation mode

G01	Linear interpolation
G02, G03	Circular interpolation
G04	Dwell, Exact stop
G40, G41, G42	Cutter compensation
G65, G66, G67	Custom macro command
G90, G91	Absolute command, incremental command
G94, G95	Feed per minute, feed per revolution

Using G02, G03 in the polar coordinate plane

The addresses used for the specification of the radius of an arc with respect to circular interpolation (G02 or G03) applied to a polar coordinate interpolation plane

are dependent on the first axis in the plane (linear axis).
- I and J in the Xp-Yp plane whenever the linear axis is the X axis or, alternatively, an axis parallel to the X axis.
- J and K in the Yp-Zp plane whenever the linear axis is the Y axis or, alternatively, an axis parallel to the Y axis.
- K and I in the Zp-Xp plane whenever the linear axis is the Z axis or, alternatively, an axis parallel to the Z axis.

Address R can also be used to specify the radius of an arc.

Example

X axis (linear axis), C axis (rotary axis)

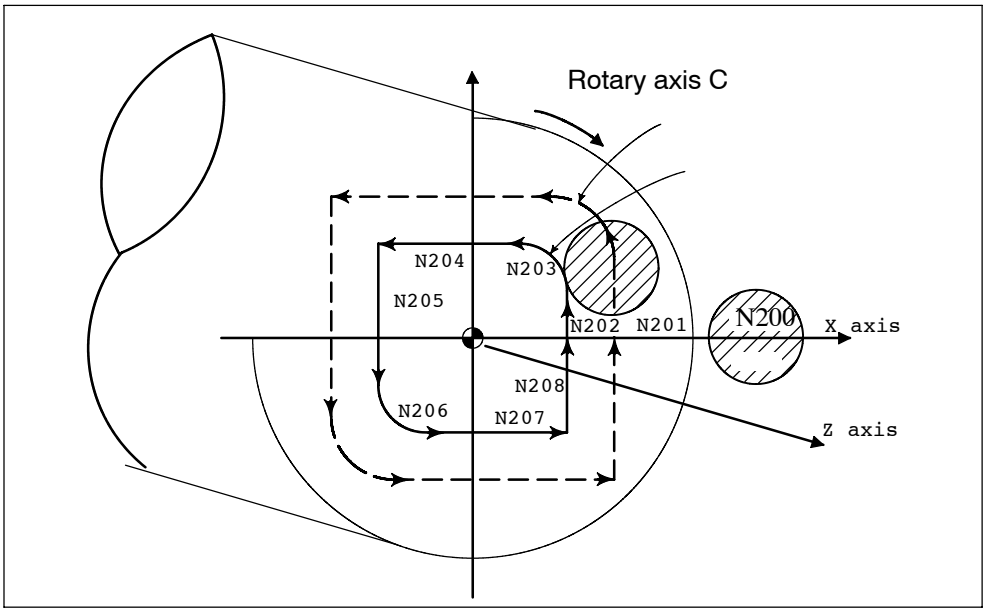


Fig. 4-22 Example of polar coordinate interpolation

```
N010 T0101;
N0100 G90 G00 X60.0 C0 Z..;
N0200 G12.1;              Polar coordinate interpolation ON
N0201 G42 G01 X20.0 F1000;
N0202 C10.0;
N0203 G03 X10.0 C20.0 R10.0
N0204 G01 X-20.0;
N0205 C-10.0;
N0206 G03 X-10.0 C-20.0 I10.0 J0;
N0207 G01 X20.0;
N0208 C0;
N0209 G40 X60.0;
N0210 G13.1;              Polar coordinate interpolation OFF
N0300 Z..;
N0400 X.. C..;
N0900 M30;
```

4.7 Cylindrical interpolation (G07.1)

This interpolation feature allows the machining to be accomplished by the combination of tool movements and rotation of a workpiece in the virtual orthogonal coordinate system. Machining is possible on the circumference of cylindrical workpiece by using the commands in an orthogonal coordinate system. To use this function, an additional axis of rotation is necessary in addition to the normal servo axes (X, Y, and Z axes).

Programming format

The cylindrical interpolation mode is turned ON and OFF by the G codes indicated below.

Table 4-6 G codes used for cylindrical interpolation

G code	Function	Group
G07.1	Cylindrical interpolation mode	16

Format

G07.1 A (B, C) r ;

Starts the cylindrical interpolation mode (enables cylindrical interpolation).

G07.1 A (B, C) 0 ;

The cylindrical interpolation mode is cancelled.

A, B, C: An address for the rotation axis
r: The radius of the cylinder

Specify the G07.1 command in a block without other commands.

The G07.1 command is modal and once the G07.1 command is specified, the cylindrical interpolation mode remains ON until the G07.1 A (B, C) command is specified. The NC is in the cylindrical interpolation OFF mode when the power is turned ON or when the NC is reset.

Note

- G07.1 is based on the Siemens option TRACYL. The relevant machine data need to be set accordingly.
- For details refer to the manual “Extended Functions”, chapter M1, 2.2 ”TRACYL”.

Example of programming

The following program is created on the cylindrical plane (the plane obtained by developing the circumference of the cylindrical workpiece) where the Z-axis is taken as the linear axis and the A-axis is taken as the rotary axis.

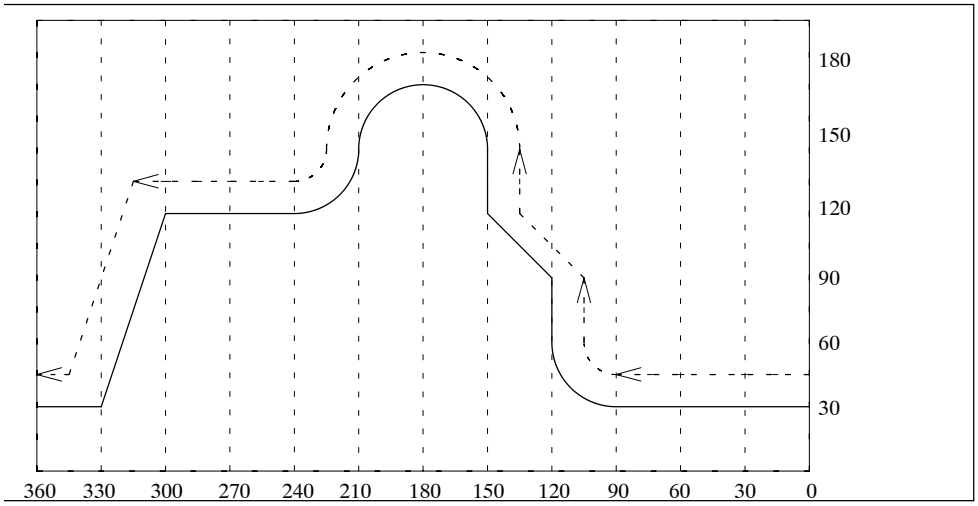


Fig. 4-23 G07.1 – Example of programming

Program

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

Programming in the cylindrical interpolation mode

In the cylindrical interpolation mode, only the following G codes can be used: G00, G01, G02, G03, G04, G40, G41, G42, G65, G66, G67, G90, G91, and G7.1. Concerning the G00 command, only the axes not included in the cylindrical plane can be designated in the G00 mode.

1. G00 (positioning command)
   The G00 command can be specified only for the axes which are not included in the cylindrical plane. Positioning is not possible on the cylindrical plane. If positioning is required for the axis which is included in the cylindrical plane, the cylindrical interpolation mode must be canceled once.
2. G01 (linear interpolation command)
   This command can be specified for all axes. However, it is not allowed to specify the axis included in the cylindrical plane and the one not included in the cylindrical plane in the same block.
   The designation of the end point for the linear interpolation should be made in either “mm” or “inch” for both the linear and rotary axes.
   Feedrates of the axes are controlled so that the vector sum (tangential velocity in the direction of tool movement) of the linear axis feedrate and the rotary axis feedrate will be the feedrate specified in the program.
3. G02/G03 (circular interpolation commands)
   The circular interpolation commands can be specified only for the axes included in the cylindrical plane.
   The designation of the end point for the circular interpolation should be made in either “mm” or “inch” for both the linear and rotary axes.
   The radius for the circular interpolation should be specified by an R command or by specifying the center of the arc. When an R command is used, designation of the radius should be made in either “mm” or “inch”. If the center of the arc should be designated instead of the R command, specify the distance from the start point to the center of the arc by signed incremental value using addresses I, J, and K.
   - If the linear axis is X-axis, use I and J assuming the XY plane.
   - If the linear axis is Y-axis, use J and K assuming the YZ plane.
   - If the linear axis is Z-axis, use K and I assuming the ZX plane.
4. G40/G41/G42
   The tool radius offset C function can be used only in the cylindrical plane. The D command specifying the offset memory number may be specified in any block. To execute tool radius offset in the cylindrical plane, turn ON the cylindrical interpolation mode and the tool radius offset mode.
   The tool path in the cylindrical plane is offset by the tool radius set in the tool offset data memory. The direction of offset is specified by G41 and G42.
   It is necessary to cancel the offset by specifying the G40 command before turning the cylindrical interpolation mode OFF.
5. G90/G91 (absolute/incremental commands)
   It is allowed to change the dimension data designation mode between absolute and incremental while in the cylindrical interpolation mode. Designation can be made in the same manner as in the normal mode.

Relationship between the Cylindrical Interpolation and Operations

- The following functions cannot be specified in the cylindrical interpolation mode. Similarly, it is not allowed to specify the G07.1 command while any of the functions indicated below is called.
  - Mirror image
  - Scaling (G50, G51)
  - Coordinate rotation (G68)
  - Base coordinate system setting
- Overrides (rapid traverse, jog, spindle speed) are valid.
- When the cylindrical interpolation mode is canceled, the interpolation plane selected before the call of the cylindrical interpolation mode is recovered.
- In the cylindrical interpolation mode, the stored stroke limit function is valid.
- To execute tool length offset, specify the tool length offset command before specifying the G07.1 command.
- The workpiece coordinate (G54 - G59) must be specified before specifying the G07.1 command.

4.8 Program support functions (2)

4.8.1 Working area limitation (G22, G23)

The working area limitation function checks whether the present position of axes operated manually or automatically enters the stored stroke limit (entry prohibited area) which is set by G22. If an axis has entered the stroke end limit, operation is stopped and an alarm occurs.

A protection zone predetermined by machine data setting must exist if G commands G22 and G23 are used. Further, the following machine data need to be set:
$MN_NUM_PROTECT_AREA_NCK = 2 (minimum)
$MC_NUM_PROTECT_AREA_ACTIVE = 2 (minimum)

When programming G22, the area inside the boundary becomes the forbidden area.

An upper (G23) and lower (G22) working area limit is defined for each axis. These values apply immediately and are not lost on Reset and when the control is switched on again. The tool (milling tool) radius can be changed in the channel–specific machine data $MC_WORKAREA_WITH_TOOL_RADIUS.

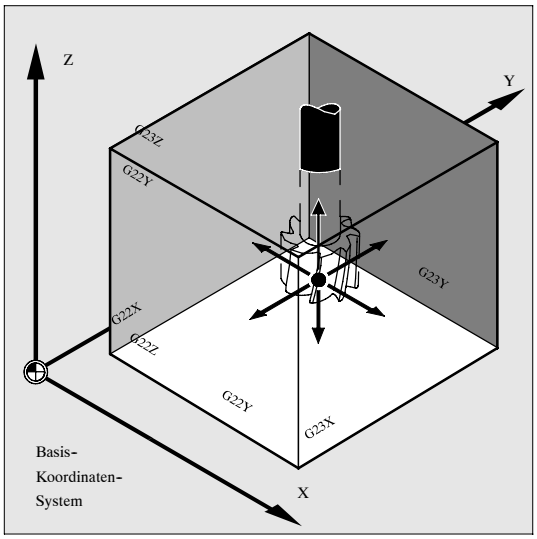


Fig. 4-24

Status at power-on

Whether working area limitation is enabled or disabled at power-on is decided by the following machine data:

$MC_EXTERN_GCODE_RESET_VALUES[3]

This MD is set to value 2 (G23) as default.

4.8.2 Chamfering and corner rounding commands

It is possible to insert chamfering and corner rounding blocks automatically between the following items:
- Linear interpolation and linear interpolation blocks
- Linear interpolation and circular interpolation blocks
- Circular interpolation and linear interpolation blocks
- Circular interpolation and circular interpolation blocks

Format

, C...; Champfering
, R...; Corner rounding

Explanations

A chamfering or corner rounding block is inserted whenever the above specification is added to the end of a block that specifies linear interpolation (G01) or circular interpolation (G02 or G03). It is possible to specify blocks applying chamfering and corner rounding consecutively.

Example

N10 G1 X10. Y100. F1000 G18
N20 A140 C7.5
N30 X80. Y70. A95.824, R10

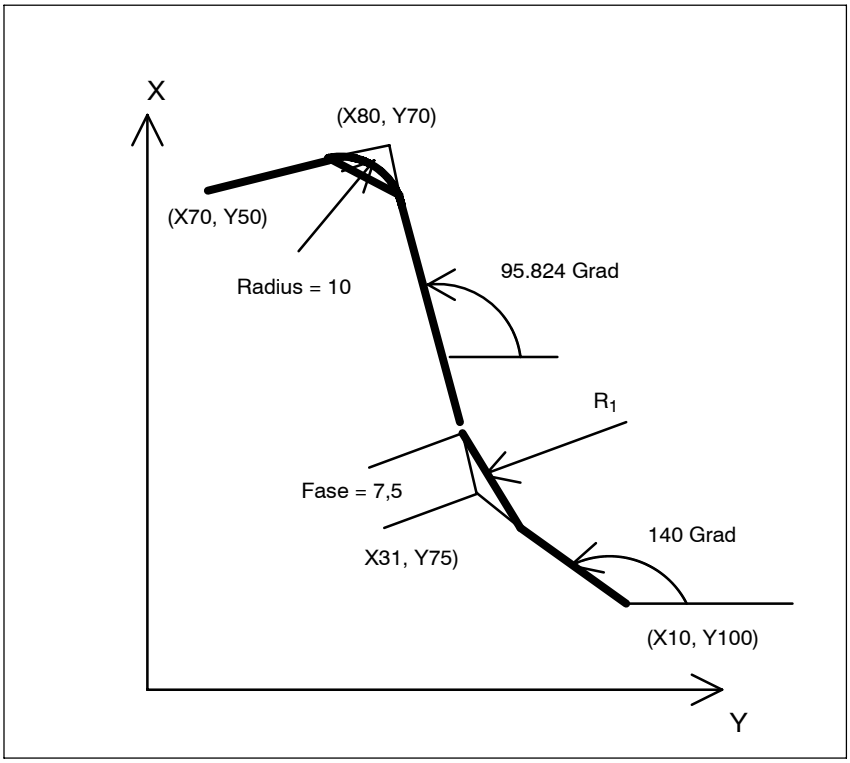


Fig. 4-25 Chamfer and corner R

Restrictions

ISO dialect mode

Address C is used in ISO Dialect0 mode both as an axis identifier and as an identifier for a chamfer on the contour.
Address R can be a cycle parameter or an identifier for the radius in a contour.
In order to distinguish between these two options, a “,” must be placed in front of the C or R address during contour definition programming.

Siemens mode

The identifiers for radius and chamfer are defined by machine data in Siemens mode. This prevents the occurrence of name conflicts. A comma must not be programmed before the identifier for radius or chamfer. The relevant MD are as follows:
MD for radius: $MN_RADIUS_NAME
MD for chamfer: $MN_CHAMFER_NAME

Plane selection

It is only possible to carry out chamfering and corner rounding in the plane specified via plane selection (G17, G18, or G19). Parallel axescannot be treated withthese functions.

Switching planes

A chamfering or corner rounding block can be inserted only for move commands which are performed in the same plane. In a block that comes immediately after plane switching (G17, G18, or G19 is specified), neither chamfering nor corner rounding can be specified.

Going to the next block

A block that specifies a move command using linear interpolation (G01) or circular interpolation (G02 or G03) must follow a block specifying chamfering or corner rounding. An alarm is issued whenever the next block does not contain these specifications.

Coordinate system

Neither chamfering nor corner rounding can be applied to a block that immediately succeds a change of the coordinate system (G92, or G52 to G59) or a specification of a return to the reference position (G28 to G30) .

Travel distance 0

Assuming the angle between the two straight lines is within +1, the chamfering or corner rounding block is regarded as having a travel distance of zero when two linear interpolation operations are performed. Assuming the angle between the straight line and the tangent to the arc at the intersection is within +1, the corner rounding block is regarded as having a travel distance of zero when linear interpolation and circular interpolation operations are performed. Assuming the angle between the tangents to the arcs at the intersection is within +1, the corner rounding block is regarded as having a travel distance of zero when two circular interpolation operations are carried out.

Cutting a thread

Within a threading block, corner rounding cannot be specified.

4.9 Automating support functions

4.9.1 Skip function (G31)

By specifying “G31 X... Y... Z... F... ;”, special linear interpolation is executed. If a skip signal is input during the execution of linear interpolation, linear interpolation is interrupted and the program advances to the next block without executing the remaining linear interpolation.

Delay from the input of the skip signal to the start of processing corresponding to the input signal is shorter than 0.5 msec; this is processed at extremely high speed.

Format

G31 X... Y... Z... F... ;
G31: One−shot G code (It is effective only in the block in which it is specified)

Explanations

The coordinate values when the skip signal is turned on can be used in a macro because they are stored as follows:
$AA_MW[X]: Position value in work coordinate system
$AA_MM[X]: Position value in machine coordinate system

In ISO Dialect mode, the PLC signals are evaluated in every block, irrespective of G31. G31 activates probe1. The deleted distance to go can be calculated via the PLC Var selector.

Note

An alarm is transmitted whenever the G31 command is issued while cutter compensation is being applied. Before the G31 command is specified, cancel cutter compensation through the G40 command.

Example

The next block to G31 represents an incremental command

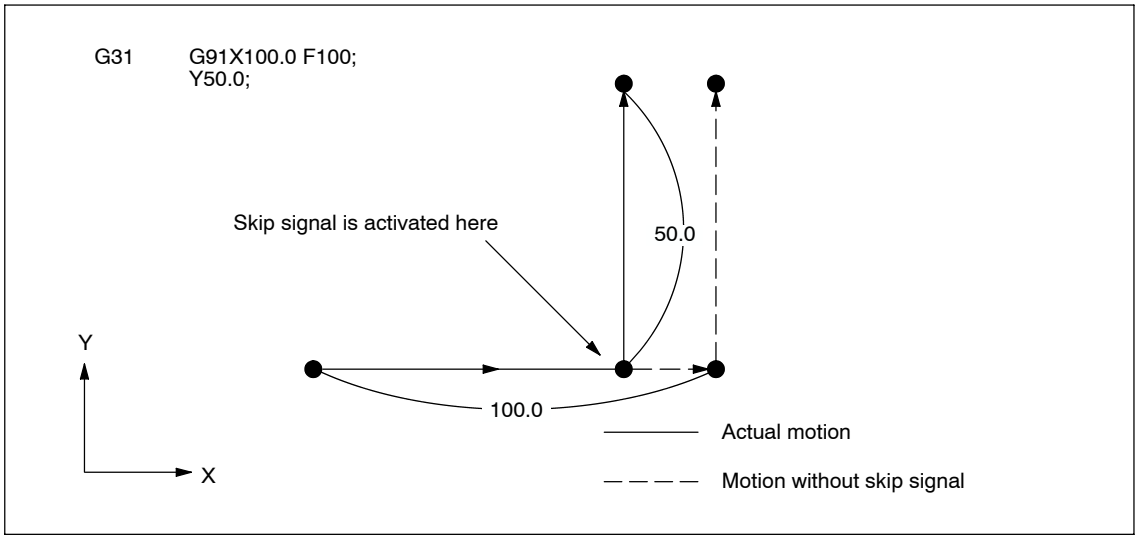


Fig. 4-26 The next block represents an incremental command

The next block to G31 represents an absolute command for 1 axis

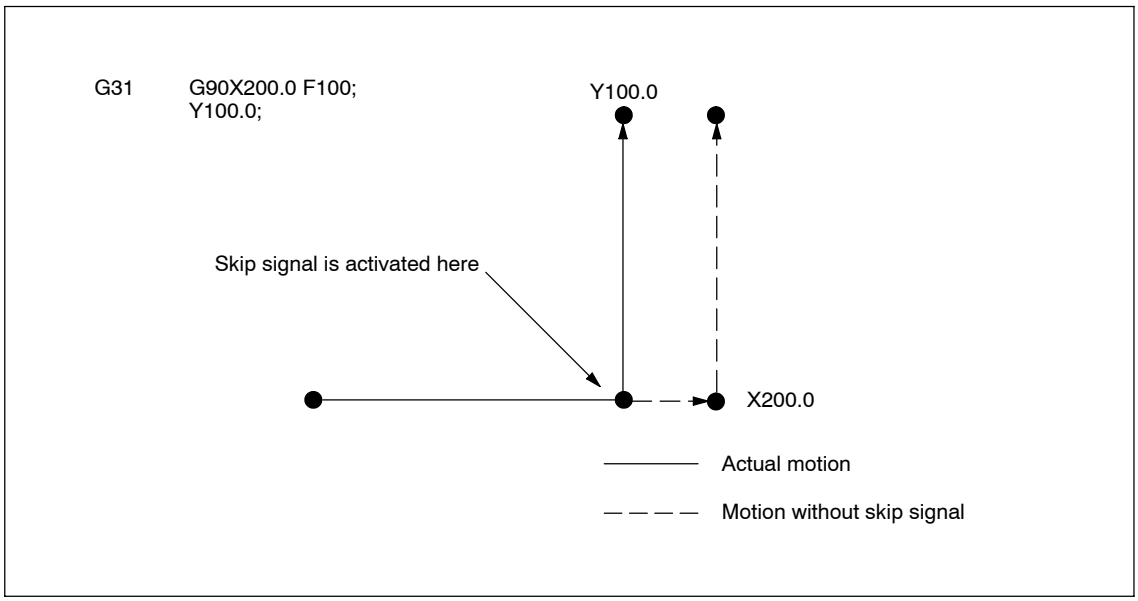


Fig. 4-27 The next block represents an absolute command for 1 axis

The next block to G31 represents an absolute command for 2 axes

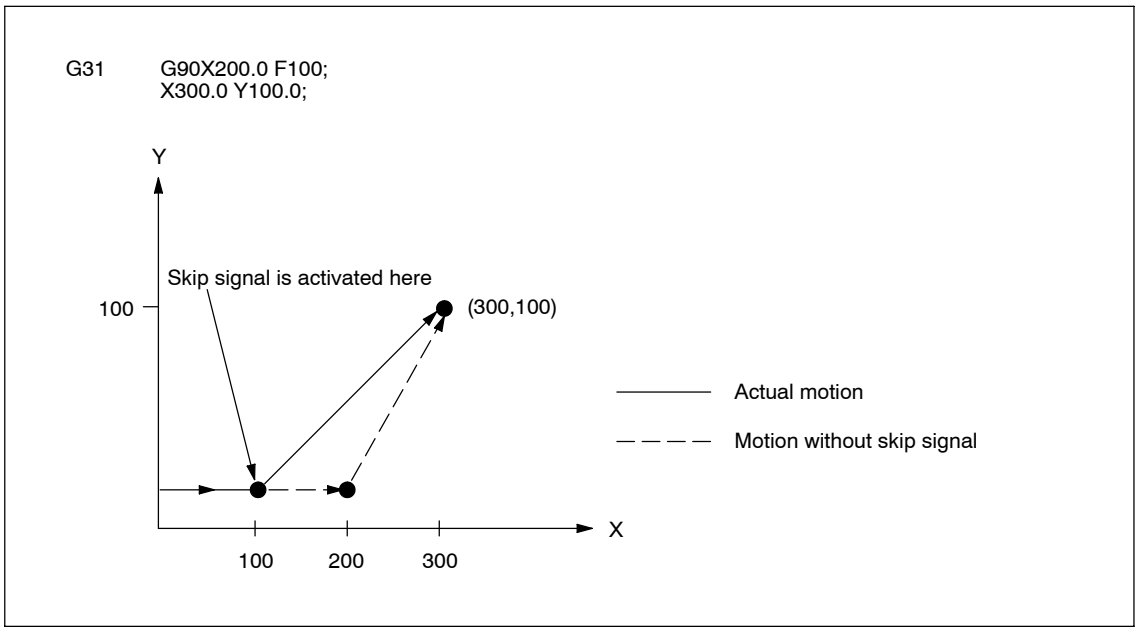


Fig. 4-28 The next block represents an absolute command for 2 axes

4.9.2 Multistage skip (G31, P1 - P4)

The multistage skip function stores coordinates in a macro variable within a block specifying P1 to P4 after G31 whenever a skip signal (4-point) is turned on. In order to match multiple Pn (n=1,2,3,4) as well as to match a Pn on a one-to-one basis, one skip signal can be set at a time.

Format

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

X, Y, Z : End point
F... : Feedrate
P... : P1-P4

Explanation

Multistage skip is caused by specifying P1, P2, P3, or P4 in a G31 block.
The digital inputs are assigned to addresses P1 – P4 through 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]

For an explanation of selecting (P1, P2, P3, or P4), refer to the manual supplied by the machine tool builder.

4.9.3 Program interrupt function (M96, M97)

By activating an external interrupt signal from the machine, another program can be called while a program is being executed. This function is referred to as program interrupt function. It is emulated using the Siemens syntax SETINT(1) <program name> [PRIO=1].
Program an interrupt command in the following format:

Format

M96 Pxxxx;	Enables program interrupt
M97;	Disables program interrupt

M97 and M96 P_ should be specified in a block without other commands. If other commands such as axis move commands are specified with M97 or M96 P_ in the same block, an alarm occurs.

Programming format

Start of interruption (M96)

By specifying “M96P · · · ;”, if the program interrupt signal goes ON during the execution of the program before the execution of M97, the program presently executed is interrupted (axis move is decelerated and stopped), and the program jumps to the one specified by P.

Example

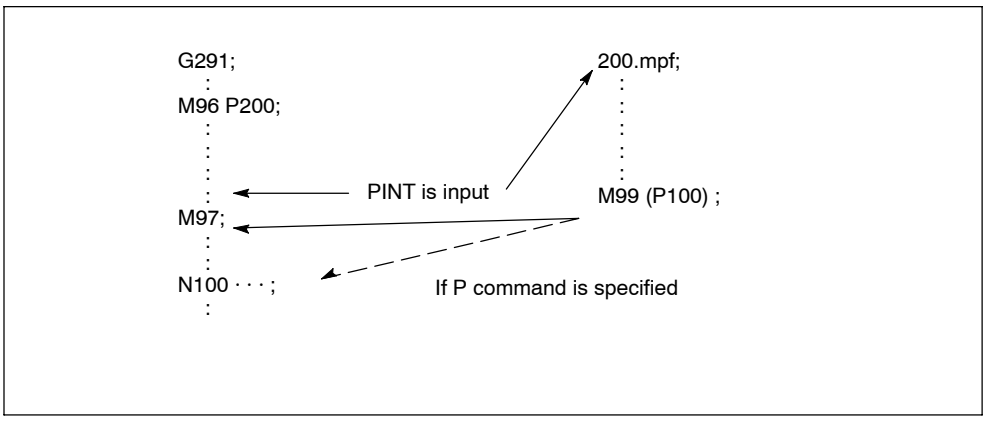


Fig. 4-29

- While the interrupt program, where jump has been made in response to the input of the interrupt signal during the execution of a program in the M96 mode, is executed, another interrupt signal is invalid.
- It is possible to specify the sequence number of the block where the interrupt program should start by using a Q command in the M96 P_ block.

End of interruption (M97)

The program interrupt function is canceled by specifying “M97;”.

Supplements to the program interrupt function

- The behaviour of the program interrupt function can be determined by setting the relevant bits of the following machine data:
  $MN_EXTERN_INTERRUPT_BITS_M96:
  Bit 0 = 0: No interrupt function possible.
  M96/M97 are treated as standard M functions
  Bit 0 = 1: Activation of program interrupt function possible

  Bit 1 = 0: Part program execution is continued with the end position of the NC block subsequent to the interruption block.
  Bit 1 = 1: Part program execution is continued from the interruption position

  Bit 2 = 0: NC block execution is interrupted immediatly and the subprogram is called.
  Bit 2 = 1: The subprogram is called after completion of the currently executed NC block.

  Bit 3 = 0: Machining cycle is interrupted if an interrupt signal occurs.
  Bit 3 = 1: Machining cycle is completed prior to subprogram call.

(The data bits are evaluated by the shell cycles)

- The M function to enable/disable the program interrupt function can be determined by machine data. However, M96, M97 is set as default.
  $MN_EXTERN_M_NO_SET_INT: enable
  $MN_EXTERN_M_NO_DISBLE_INT: disable
- In the program that is called up after interrupting the execution of another program, it is not allowed to specify M97 or M96. If specified, an alarm occurs.
- The M96 command can be specified in a subprogram. Jump to an interrupt program is not counted as a nesting level. Therefore, the level saved to the macro local variable does not change.
- By the execution of M99 specified in the interrupt program, the program returns to the block next to the one where the interrupt program has been called up. It is also possible to specify the return block by specifying a P command with M99. When returning to the previous program by the execution of M99, the modal information which was valid before the interruption, is recovered. However, if M99P_ is used to return to the previous program, the modal information changed during the execution of the interrupt program is used for the execution of the previous program.
- If the interrupt signal is input during the block stop state, the program jumps to the interrupt program when the operation is started by depressing of the cycle start switch.
- The program interrupt signal is invalid if input during the execution of high-speed cutting.
- If the program interrupt signal is input during the execution of G31 (skip), the skip mode is canceled and the program interrupt function is executed.
- If the program interrupt signal is input during the execution of a block including M, S, T, or B command, the program jumps to the interrupt program. Before jumping to the interrupt program, axis move is stopped after deceleration if the interrupt signal is input during axis move. If the M or T function is being executed when the interrupt signal is input, the program does not jump until the M or T function completion signal is input.
- If the program interrupt signal is input during the execution of tapping in the solid tap mode, execution of the interrupt program starts only after the completion of the solid tap block.

4.9.4 Tool life control function

Tool management, tool life and workpiece count monitoring can be reproduced with the Siemens tool management system.

4.10 Macroprograms

The NC has a set of instructions that can be used by the machine tool builders and the users to implement the original functions. The program created by using these instructions is called a macroprogram, which can be called and executed by the commands specified in a block with G65 or G66.

A macroprogram provides the following:

- Variables can be used.
- Arithmetic and logical operations using variables and constants are possible.
- Control commands for branch and repeat can be used.
- Commands to output messages and data can be used.
- Arguments can be specified.

This makes it possible to create a program in which complicated operations and operations requiring conditional judgment are included.

4.10.1 Differences from subprograms

Differences between macroprograms and subprograms are indicated below.

- With macroprogram call up commands (G65, G66), arguments can be specified. However, with subprogram call up command (M98), it is not possible to use arguments.
- If commands other than P, Q, and L are specified in the M98 block, the program jumps to the specified subprogram after executing these commands. With G65 and G66, commands other than P and L are regarded as argument specification and the program jumps to the specified macroprogram immediately. In this case, however, the commands specified preceding G65 and G66 are executed normally.

4.10.2 Macroprogram call (G65, G66, G67)

Macroprograms are usually executed after being called up.

The procedure used for calling up a macroprogram is indicated in Table 4-7.

Table 4-7 Macroprogram calling format

Calling up method	Command code	Remarks
Simple call up	G65	
Modal call up (a)	G66	Canceled by G67

Simple call up (G65)

Format

G65 P_ L_ ;

By specifying “G65 P ··· L ··· <argument specification>; ”, the macroprogram which is assigned the program number specified with P is called up and executed L times.

If it is necessary to pass arguments to the called up macroprogram, these arguments can be specified in this block.

Table 4-8 P and L commands

Address	Description	Number of digits
P	Program number	5 digits
L	Number of repetitions	9 digits

System variables for the addresses I, J, K

Because addresses I, J, and K can be programmed up to ten times in a block by macro call, an array index must be used to access the system variables for these addresses. The syntax for these three system variables is then $C_I[..], $C_J[..], $C_K[..]. The values are stored in the array in the order programmed. The number of addresses I, J, K programmed in the block is stored in variables $C_I_NUM, $C_J_NUM, $C_K_NUM.

The passed parameters I, J, K for macro calls are treated as one block, even if individual addresses are not programmed. If a parameter is programmed again or a following parameter has been programmed with reference to the sequence I, J, K, it belongs to the next block.
To recognize the programming sequence in ISO mode, system variables $C_I_ORDER, $C_J_ORDER, $C_K_ORDER are set. These are identical arrays to $C_I, $C_K and contain the associated number of parameters.

Note

The transfer parameters can only be read in the subroutine.

Example:

```
N5 I10 J10 K30 J22 K55 I44 K33
    set1    set2      set3
$C_I_NUM=2
$C_I[0]=10
$C_I[1]=44
$C_I_ORDER[0]=1
```

```
$C_I_ORDER[1]=3

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

$C_K_NUM=3
$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 ISO dialect 0 mode, the programmed values can be evaluated differently depending on the programming method (integer or real value). The different evaluation is activated via machine data.

If the MD is set, the control will behave as in the following example:

```
X100.    ;X axis is traveled 100 mm (100. with point => real value
Y200     ;Y axis is traveled  0.2 mm (200 without point => integer value
```

If the addresses programmed in the block are passed as parameters for cycles, the programmed values are always real values in the $C_x variables. In the case of integer values, the cycles do not indicate the programming method (real/integer) and therefore no evaluation of the programmed value with the correct conversion factor.

To indicate whether REAL or INTEGER has been programmed, there is the system variable $C_TYP_PROG. $C_TYP_PROG has the same structure as $C_ALL_PROG and $C_INC_PROG. For each address (A–Z) there is one bit. If the value is programmed as an INTEGER, the bit is set to 0, for REAL it is set to 1. If the value is programmed in variable $<number>, bit 2 = 1 is set.

Example:

M98 A100. X100 -> $C_TYP_PROG == 1.
Only bit 0 is set because only A is programmed as a REAL.

M98 A100. C20. X100 -> $C_TYP_PROG == 5.
Only bits 1 and 3 are set (A and C).

Restrictions:

Up to ten I, J, K parameters can be programmed in each block. Variable $C_TYP_PROG only contains one bit each for I, J, K. For that reason bit 2 is always set to 0 for I, J, and K in $C_TYP_PROG. It is therefore not possible to ascertain whether I, J or K have been programmed as REAL or INTEGER.

Parameters P, L, O, N can only be programmed as integers. A real value generates an NC alarm. For that reason the bit in $C_TYP_PROG is always 0.

Modal call up (G66, G67)

The modal call up commands set the mode for calling up a macroprogram. The specified macroprogram is called up and executed when the specified conditions are satisfied.

- By specifying “G66 P · · · L · · · <argument-specification>; ”, the mode for calling up the macroprogram is set. Once this block is executed, the macroprogram which is assigned the program number specified with P is called up and executed L times after the completion of move commands.

  If an argument is specified, the argument is passed to the macroprogram each time it is called up as with the simple call up of a macroprogram. The correspondence between the address of argument and local variables is the same as in the case of simple call up (G65).
- G67 cancels the G66 mode. When arguments are specified, G66 must be specified before all arguments. If G66 is specified, G67 must be specified in the same program corresponding to it.

Table 4-9 Modal call up conditions

Call up conditions	Mode setting code	Mode cancel code
After the execution of move command	G66	G67

Specifying argument

The term “to specify argument” means “assigning a real number” for local variables used in a macroprogram. There are two types of argument specifications: type I and type II. These types can be used as required, including a combination of the two types.

Correspondence between addresses and system variables (Type I)

Table 4-10 Address – variable correspondence and usable addresses for call up commands (type I)

Address - variable correspondence		Address - variable correspondence	
Address in Type I	**System variable**	**Address in Type I**	**System variable**
A	$C_A	Q	$C_Q
B	$C_B	R	$C_R
C	$C_C	S	$C_S
D	$C_D	T	$C_T
E	$C_E	U	$C_U
F	$C_F	V	$C_V

Table 4-10 Address – variable correspondence and usable addresses for call up commands (type I), continued

Address in Type I	System variable	Address in Type I	System variable
H	$C_H	W	$C_W
I	$C_I[0]	X	$C_X
J	$C_J[0]	Y	$C_Y
K	$C_K[0]	Z	$C_Z
M	$C_M		

Correspondence between addresses and system variables (Type II)

To use I, J, and K, they must be specified in the order of I, J, and K.

Since addresses I, J, K can be programmed up to ten times in a block with macro call, an array index must be used to access the system variables within the macro program for these addresses. The syntax for these three system variables is then $C_I[..], $C_j[..], $C_K[..]. The values are stored in the array in the order programmed. The number of addresses I, J, K programmed in the block is stored in variables $C_I_NUM, $C_J_NUM and $C_K_NUM.

Unlike the rest of the system variables, an array index must always be specified for these three variables. Array index 0 must always be used for cycle calls (e.g. G81); e.g. N100 R10 = $C_I[0]

Table 4-11 Address – variable correspondence and usable addresses for call up commands (type II)

Address - variable correspondence		Address - variable correspondence	
Address in Type II	**System variable**	**Address in Type II**	**System variable**
A	$C_A	K5	$C_K[4]
B	$C_B	I6	$C_I[5]
C	$C_C	J6	$C_J[5]
I1	$C_I[0]	K6	$C_K[5]
J1	$C_J[0]	I7	$C_I[6]
K1	$C_K[0]	J7	$C_J[6]
I2	$C_I[1]	K7	$C_K[6]
J2	$C_J[1]	I8	$C_I[7]
K2	$C_K[1]	J8	$C_J[7]
I3	$C_I[2]	K8	$C_K[7]
J3	$C_J[2]	I9	$C_I[8]
K3	$C_K[2]	J9	$C_J[8]
I4	$C_I[3]	K9	$C_K[8]
J4	$C_J[3]	I10	$C_I[9]
K4	$C_K[3]	J10	$C_J[9]

Table 4-11 Address – variable correspondence and usable addresses for call up commands (type II), continued

Address in Type II	System variable	Address in Type II	System variable
I5	$C_I[4]	K10	$C_K[9]
J5	$C_J[4]		

Note: If more than one set of I, J, or K is specified, the order of sets is determined for each I/J/K set, so that variable numbers are determined corresponding to that order.

Example of argument specification

When arguments are specified, the macroprogram call up code must always be specified before the specification of arguments. If argument specification is given before the macroprogram call up code, an alarm occurs. The value of argument specification can include a sign and decimal point independent of the address.

If no decimal point is used, the value is saved to the variable as the value with a decimal point according to the normal number of digits of that address.

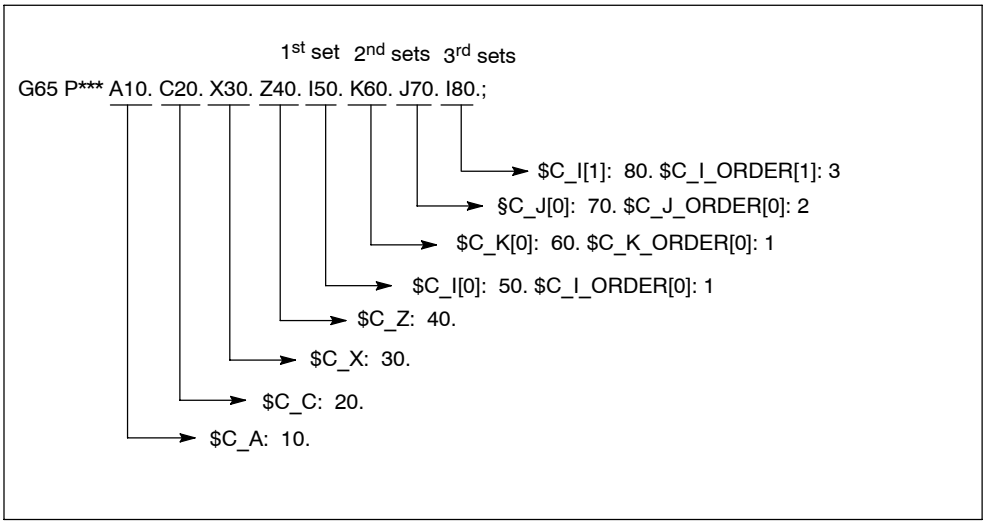


Fig. 4-30 Example of argument specification

Siemens mode/ISO mode macro program execution

The called macro program can either be executed in Siemens mode or ISO mode. The execution mode is decided in the first block of the macro program.
If a PROC <program name> instruction is included in the first block of the macro program, it is automatically switched to Siemens mode. If no such instruction is included, ISO mode is retained.

By executing a macro program in Siemens mode, transfer parameters can be stored into local variables using the DEF instruction. In ISO mode, however, transfer parameters cannot be stored into local variables.

In order to read transfer parameters within the macroprogram executed in ISO mode, switch to Siemens mode by G290 command.

Examples

Main program containing the macro call:

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

Macro program in Siemens mode:

```
_N_0010_SPF:
PROC 0010 ; Switching into Siemens mode
N10 DEF REAL X_AXIS ,Y_AXIS, S_SPEED, FEED
N15 X_AXIS = $C_X Y_AXIS = $C_Y S_SPEED = $C_S FEED = $C_F
N20 G01 F=FEED G95 S=S_SPEED
...
N80 M17
```

Macro program in ISO mode:

```
_N_0010_SPF:
G290 ; Switching into Siemens mode,
           ; if transfer parameters need to be read
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 ; Switching into ISO mode
N15 M3 G54 T1
N20
...
N80 M99
```

4.10.3 Macro Call via G Function

Analogous to G65, a macro can be called up via a G number.

50 G function replacements ca be configured by means of the machine data

$MN_EXTERN_G_NO_MAC_CYCLE and
$MN_EXTERN_G_NO_MAC_CYCLE_NAME.

The address A–F, H–K, M and P–Z can be use as the parameter. Only the integer value can be used for the address P and L. Address L is for the number of repition. Please refer to the chapter 4.10.2 ”Macroprogram call (G65, G66, G67)” for the parameter programmed in the block.
The number of the programmed G macro is stored in the variable $C_G. All further G functions programmed in the block are treated like ordinary G functions and performed before the execution of macro program. The programming sequence of the addresses and G functions in the block is not fixed and does not influence the functionality.

Restrictions

- The G function replacement can be execute in ISO mode (G290).
- The address O with G function replacement is signaled with an alarm.
- One single G/M function replacement (or generally only one subroutine call) can be executed per parts program line. Conflicts with other subroutine calls, e.g. when a modal subroutine call is active, are signaled with alarm 12722.
- The G function replacement with end of subprogram (M99) or end of program (M02, M30) is denied by the alarm.
- If a G macro is active, no other G/M macro or M subroutine is called up. In this case, M macros/subroutines are executed as M functions. G macros are executed as G function, if a corresponding G function exists; otherwise, alarm 12470 ”G function unknown” is output.
- In all other respects, the same restriction as for G65 are valid.

Configuration examples

Call of the MAKRO_G21 subroutine by the G21 function as well as G123.

$MN_EXTERN_G_NO_MAC_CYCLE[0] = 21

$MN_EXTERN_G_NO_MAC_CYCLE_NAME[0] = ”MAKRO_G21”

$MN_EXTERN_G_NO_MAC_CYCLE[1] = 123

$MN_EXTERN_G_NO_MAC_CYCLE_NAME[1] = ”MAKRO_G123”

$MN_EXTERN_G_NO_MAC_CYCLE[2] = 421

$MN_EXTERN_G_NO_MAC_CYCLE_NAME[2] = ”MAKRO_G123”

Programming example

```
PROC MAIN
...
N0090 G1 G21 X10 Y20 F1000 G90  ;call of MAKRO_G21.SPF, G1 and G90 are
                                ;activated before the MAKRO_G21.SPF call
...
N0500 G90 X20 Y30 G123 G1 G54   ;call of MAKRO_G123.SPF, G1, G54 and
                                ;G90 are activated before the
                                ;MAKRO_G123.SPF call
...
N0800 G90 X20 Y30 G421 G1 G54   ;Call of MAKRO_G123.SPF, G1, G54 and
                                ;G90 are activated before the
                                ;MAKRO_G123.SPF call
...
N0900 M30
```

PROC MAKRO_G21

```
...
N0010 R10 = R10 + 11.11
N0020 IF $C_X_PROG = = 0
N0030 SETAL (61000)             ;programmed variable transmitted
                                ;incorrectly
N0040 ENDIF
N0050 IF $C_V_PROG = = 0
N0060 SETAL (61001)
N0070 ENDIF
N0080 IF $C_F_PROG = = 0
N0090 SETAL (61002)
N0100 ENDIF
N0110 G90 X=$C_X V=$C_V
N0120 G291
N0130 G21 M6 X100               ;G21 -> activate metric measuring system
                                ;(no macro call)
N0140 G290
...
N0150 M17
```

PROC MAKRO_G123

```
...
N0010 R10 = R10 + 11.11
N0020 IF $C_G = = 421 GOTO label_G421 ;macro functionality for G123
N0040 G91 X=$C_X Y=$C_Y F500
...
...
N1990 GOTOF label_end
N2000 label_G421                      ;macro functionality for G421
N2010 G90 X=$C_X Y=$C_Y F1000
N2020 ...
...
```

```
N3000 G291
N3010 G123                    ;alarm 12470, as G123 is no G
                              ;function and a macro call is
                              ;not possible as long as the macro is
                              ;active. Exeption: The macro was
                              ;called up as subroutine with CALL
                              ;MAKRO_G123
```

4.11 Additional functions

4.11.1 Figure copy (G72.1, G72.2)

Does not work with SINUMERIK 802D sl.

Using the figure copy function, a once programmed contour can be easily repeated or, respectivly, copied. A linear (G72.2) or rotational (G72.1) copy can be carried out by means of this function.

Format

G72.1 X... Y... (Z...) P... L... R...

X, Y, Z: Reference point for the rotation of coordinates
P: Sub−program number
L: Number of sub−program repeats
R: Rotation angle

Through G72.1, a sub−program containing the contour to be copied can be called repeatedly. Prior to calling each sub−program, the coordinate system is rotated by a certain angle. The coordinate rotation is carried out along the axis perpendicular to the selected plane.

G72.2 I... J... K... P... L...

I, J, K: X, Y, Z position prior to sub−program call
P: Sub−program number
L: Number of sub−program repeats

Through G72.2, a sub−program, in which the contour to be repeated is programmed, is repeatedly called. Prior to each sub−program call, the axes programmed through I, J, K are traversed incrementally. The cycle calls the sub−program by the number of times specified by address L. Prior to each sub−program call, a path programmed in I, J, K and calculated from the initial point is traversed incrementally.

Examples

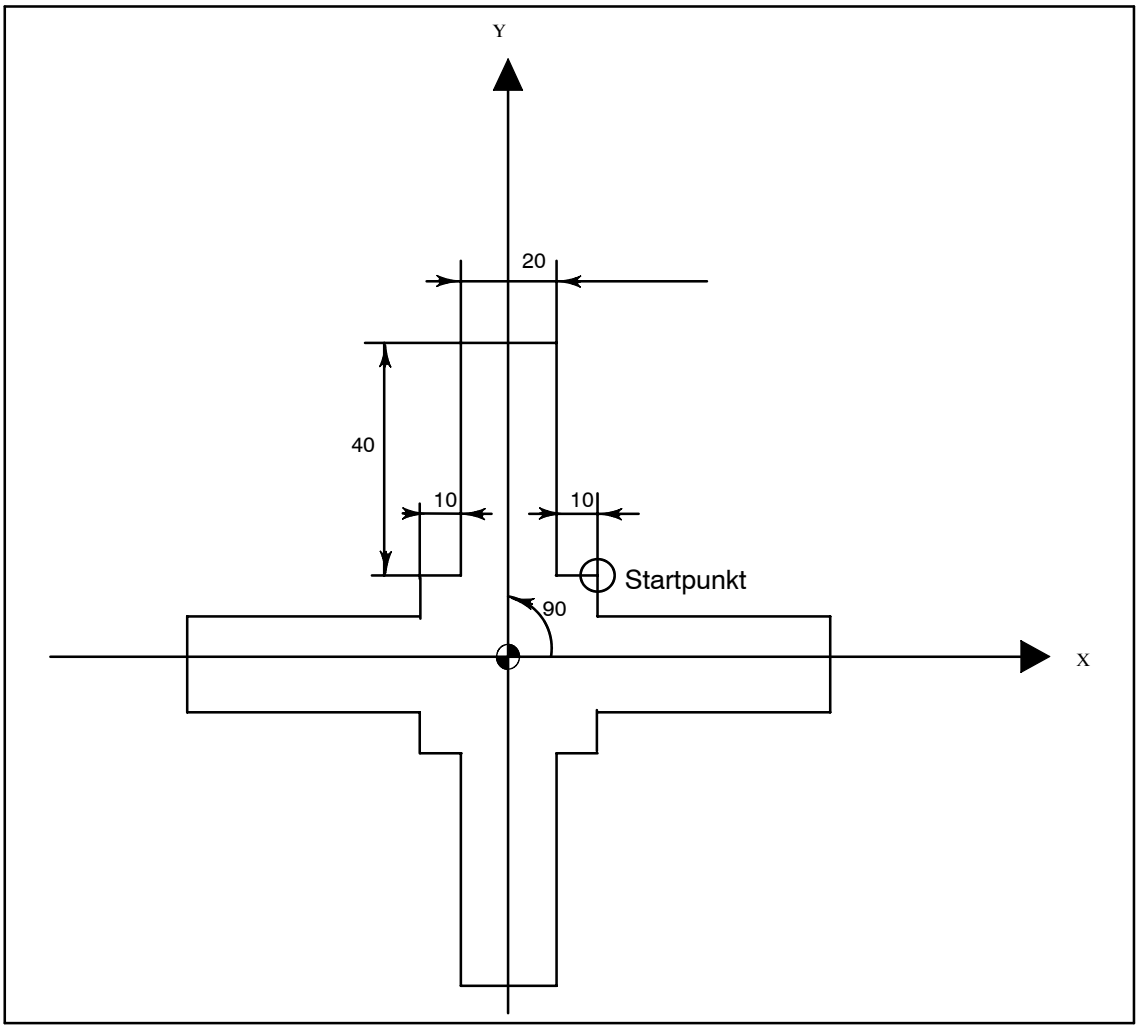


Fig. 4-31 Contour repetition using G72.1

```
Main program
N10 G92 X40.0 Y50.0 ;
N20 G01 G90 G17 G41 20 Y20 D01 F1000
N30 G72.1 P1234 L4 X0 Y0 R90.0
N40 G40 G01 X100 Y50 Z0
N50 G00 X40.0 Y50.0 ;
N60 M30 ;

Subprogram 1234.spf
N100 G01 X10
N200 Y50
N300 X-10
N400 Y10
```

```
N500 X-20
N600 M99
```

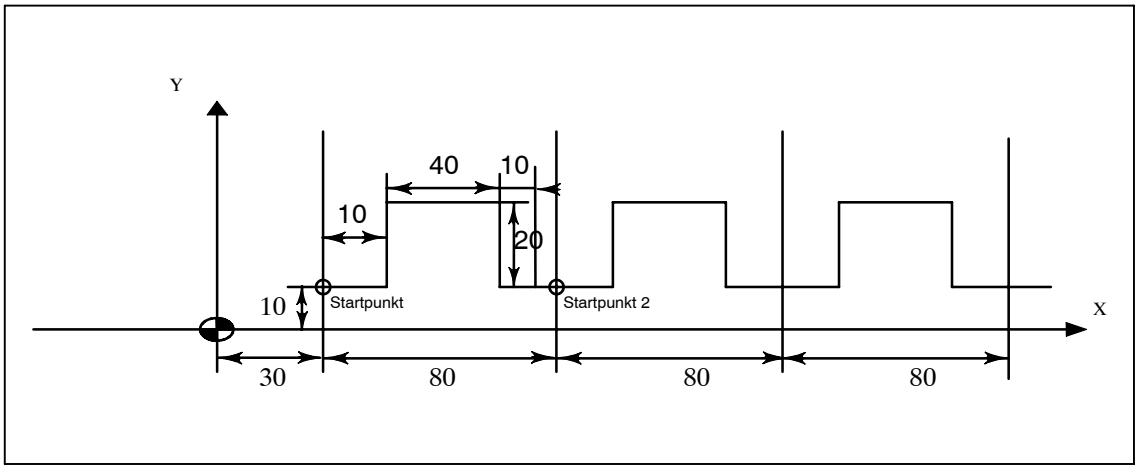


Fig. 4-32 Contour repetition using G72.2

```
N10 G00 G90 X0 Y0
N20 G01 G17 G41 X30. Y0 D01 F1000
N30 Y10.
N40 X30.
N50 G72.2 P2000 L3 I80. J0

O2000 G90 G01 X40.
N100 Y30.
N200 G01 X80.
N300 G01 Y10.
N400 X90.
N500 M99
```

4.11.2 Switchover modes for DryRun and skip levels

Switching over the skip levels (DB21 DBB2) always constitutes an intervention in the program run, resulting in a brief drop in velocity along the path in earlier SW versions. The same applies to the switchover of DryRun mode (DryRun = dry run feedrate DB21.DBB0.BIT6) from DryRunOff to DryRunOn or vice versa.

With a new switchover mode that has limited functionality, it is now possible to avoid the drop in velocity.

By setting machine data $MN_SLASH_MASK==2, it is no longer necessary to reduce the velocity when the skip levels are switched (i.e. a new value in the PLC->NCK Chan interface DB21.DBB2).

Note

The NCK processes blocks in two stages, the preprocessing and main runs. The result of the preprocessing run is transferred to the preprocessing memory from where the main run fetches the oldest block in each case and traverses its geometry.

Attention

When you set machine data $MN_SLASH_MASK==2, the preprocessing run is switched over when the skip levels are changed! All blocks stored in the preprocessing memory are traversed with the old skip level. As the user, you generally have no control over the fill level of the preprocessing memory. From your viewpoint, therefore, the new skip level will become **operative "at some point" after the levels are switched!**

Note

Part program command STOPRE clears the preprocessing memory. If you switch the skip level over before the STOPRE command, all blocks after the command will be reliably changed over. The same applies to an implicit STOPRE.

Switching over DryRun mode is subject to analogous restrictions.

If you set machine data $MN_DRYRUN_MASK==2, no drop in velocity will be necessary when you change over the DryRun mode. In this instance as well, however, it is only the preprocessing run that is switched over, resulting in the restrictions described above. In other words: **Watch out! DryRun mode will become active "at some time" after it has been switched over!**

4.12 Interrupt programm with M96 / M97 (ASUB)

M96

A subprogram can be defined as an interrupt routine with M96 P <program number>.

This program is started by an external signal. The first high-speed NC input of the 8 inputs available in Siemens mode is always used to start the interrupt routine. Machine data $MN_EXTERN_INTERRUPT_NUM_ASUP lets you select an other fast input (1 – 8).

The function is mapped onto standard syntax**:** SETINT(x) <CYCLE396> [PRIO=1].

In shell cycle CYCLE396, the interrupt program programmed with Pxxxx is called in ISO mode. The program number is in $C_PI. At the end of the shell cycle, machine data
10808: $MN_EXTERN_INTERRUPT_BITS_M96 BIT1 is evaluated, resulting either in positioning at the interruption point with REPOSA or in continuation with the next block. The new cycle variable $C_PI contains the value programmed with “P” without leading zeroes. These must be added to fill out to four digits in the shell cycle before the subprogram is called.

Example: N0020 M96 P5

Call in shell cycle
progName = “000” << $C_PI
ISOCALLprogName

See treatment of 8-digit program numbers, if MD $MC_EXTERN_FUNCTION_MASK, bit 6 is set.

M97

M97 is used to suppress starting of the interrupt routine. The interrupt routine can then only be started by the external signal following activation with M96.

This corresponds to Standard syntax**:** ENABLE(x).

x = content of $MN_EXTERN_INTERRUPT_NUM_ASUP

If the interrupt program programmed with M96 Pxx is called up directly with the interrupt signal (without the intermediate step with CYCLE396), machine data 20734: $MC_EXTERN_FUNCTION_MASK BIT10 must be set. The subprogram programmed with Pxx is then called on a 0 –> 1 signal transition in Siemens mode.

The M function numbers for the interrupt function are set via machine data. With machine data 10804: $MN_EXTERN_M_NO_SET_INT, the M number is used to activate an interrupt routine and with MD 10806:
$MN_EXTERN_M_NO_DISABLE_INT the M number is used to suppress an interrupt routine.

Only non-standard M functions are permitted to be set. M functions M96 and M97

are set as defaults. To activate the function, bit 0 must be set in machine data 10808: $MN_EXTERN_INTERRUPT_BITS_M96. These M functions will not be output to the PLC in this case. If bit 0 is not set, the M functions will be interpreted as conventional auxiliary functions.

On completion of the “Interrupt” program, the end position of the parts program block that follows the interruption block is approached. If processing of the parts program has to continue starting from the interruption point, there must be a REPOS instruction at the end of the “Interrupt” program, e.g. REPOSA.
For this purpose the interrupt program must be written in Siemens mode.

The M functions for activating and deactivating an interrupt program must be in a block of their own. If further addresses other than “M” and “P” are programmed in the block, alarm 12080 (syntax error) is output.

Note about machining cycles

For ISO dialect original, you can set whether a machining cycle will be interrupted by an interrupt routine immediately or not until the end. The shell cycles must evaluate machine data
10808: $MN_INTERRUPT_BITS_M96 bit 3 for that purpose. If bit=1, the interrupt must be disabled at the beginning of the cycle with DISABLE(1) and reactivated at the end of the cycle with ENABLE(1) to avoid interrupting the machining cycle. Because the interrupt program is only started on a 0/1 signal transition, the interrupt input must be monitored with a disabled interrupt during the cycle runtime with a synchronized action in the shell cycle. If the interrupt signal switches from 0 to 1, the interrupt signal after the ENABLE(1) must be set once again at the end of the shell cycle, so that the interrupt program will then start. To permit writing to the interrupt input in the shell cycle, the machine data
10361: $MN_FASTO_DIG_SHORT_CIRCUIT[1] must be parameterized.

Machine data

MD $MN_EXTERN_INTERRUPT_BITS_M96:

Bit 0: = 0: Interrupt program is not possible, M96/M97 are conventional M functions
= 1: Activation of an interrupt program with M96/M97 permitted

Bit 1: = 0: Execution of parts program continues from the final position of the next block after the interruption block
= 1: Continue parts program as from interruption position **(evaluated in interrupt program (ASUB), return with/without REPOSL)**

Bit 2: = 0: The interrupt signal interrupts the current block immediately and starts the interrupt routine
= 1: The interrupt routine is not started until the block has been completed.

Bit 3: = 0: The machining cycle is interrupted on an interrupt signal

= 1: The interrupt program is not started until the machining cycle has been completed.
(evaluated in the shell cycles)

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

Bit 1 must be evaluated in the interrupt program. If bit 1 = TRUE, on completion of the program, REPOSL must be used to reposition at the interruption point.

Example:

```
N1000 M96 P1234      ; Activate ASUB 1234.spf in the case of a rising
                     ; edge on the first high-speed input, program 1234.spf
                     ; is activated
        "
        "
N3000 M97            ; Deactivate the ASUB
```

Rapid lifting (LIFTFAST) is not performed before the interrupt program is called. On the rising flank of the interrupt signal, depending on machine data MD 10808: $MN_EXTERN_INTERRUPT_BITS_M96, the interrupt program is started immediately.

Limitations in Siemens mode

The interrupt routine is handled like a conventional subprogram. This means that in order to execute the interrupt routine, at least one subprogram level must be free. (12 program levels are available in Siemens mode, there are 5 in ISO Dialect mode.)

The interrupt routine is only started on a signal transition of the interrupt signal from 0 to 1. If the interrupt signal remains permanently set to 1, the interrupt routine will not be restarted.

Limitations in ISO Dialect mode

One program level is reserved for the interrupt routine so that all permissible program levels can be reserved before the interrupt program is called.

Depending on the machine data, the interrupt program will also be started when the signal is permanently on.

Abbreviations

A

ASCII	American Standard Code for Information Interchange
ASUB	Asynchronous Subroutine
BA	Mode of operation
BAG	Mode Group
BCD	Binary Coded Decimals
BCS	Basic Coordinate System
BIN	Binary Files
BP	Basic Program
C1 .. C4	Channel 1 to channel 4
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CNC	Computerized Numerical Control
COM	Communication
COR	Coordinate Rotation
CPU	Central Processing Unit
CR	Carriage Return

CRC	Cutter Radius Compensation
CSF	Control System Flowchart (PLC programming method)
CTS	Clear To Send (serial data interfaces)
CUTOM	Cutter Radius Compensation (Tool radius compensation)
DB	Data Block in the PLC
DBB	Data Block Byte in the PLC
DBW	Data Block Word in the PLC
DBX	Data Block Bit in the PLC
DC	Direct Control: The rotary axis is moved along the shortest path to the absolute position within one revolution.
DCE	Data Communications Equipment
DDE	Dynamic Data Exchange
DIO	Data Input/Output: Data transfer display
DIR	Directory
DLL	Dynamic Link Library: Module which can be accessed by a running program. Often contains program sections that are required by different programs.
DOS	Disk Operating System
DPM	Dual-Port Memory
DPR	Dual-Port RAM
DRAM	Dynamic Random Access Memory

DRF	Differential Resolver Function
DRY	Dry Run
DSB	Decoding Single Block
DTE	Data Terminal Equipment
DW	Data Word
EIA Code	Special punchtape code, number of punched holes per character always odd
ENC	Encoder
EPROM	Erasable Programmable Read Only Memory
FB	Function Block
FC	Function Call: Function block in the PLC
FDB	Product Database
FDD	Floppy Disk Drive
FDD	Feed Drive
FEPROM	Flash–EPROM
FIFO	First In First Out: Memory which operates without address specification from which data are read in the same order as they are stored.
FM	Function Module
FM–NC	Function Module – Numerical Control
FPU	Floating Point Unit

FRA	Frame Block
FRAME	Data Record (frame)
FST	Feed Stop
GUD	Global User Data
HD	Hard Disk
HEX	Abbreviation for hexadecimal
HHU	Handheld Unit
HMI	Human Machine Interface: SINUMERIK operating functions for operator control, programming and simulation. MMC and HMI are identical in meaning.
HW	Hardware
I	Input
I/O	Input/Output
I/RF	Infeed/Regenerative Feedback Unit (power supply) of SIMODRIVE 611(D)
IK (GD)	Implicit Communication (Global Data)
IKA	Interpolative Compensation
IM	Interface Module
IMR	Interface Module Receive
IMS	Interface Module Send
INC	Increment

INI	Initializing Data
IPO	Interpolator
IS	Interface Signal
ISO Code	Special punchtape code, number of punched holes per character always even
JOG	Jog mode
K Bus	Communication Bus
$K_Ü$	Transmission Ratio
K_v	Servo Gain Factor
LAD	Ladder Diagram (PLC programming method)
LEC	Leadscrew Error Compensation
LF	Line Feed
LUD	Local User Data
MB	Megabyte
MC	Measuring Circuit
MCP	Machine Control Panel
MCS	Machine Coordinate System
MD	Machine Data
MDA	Manual Data Automatic

MMC	Human Machine Communication: User interface on numerical control systems for operator control, programming and simulation. MMC and HMI are identical in meaning.
MPF	Main Program File: NC part program (main program)
MPI	Multi Port Interface
MSD	Main Spindle Drive
NC	Numerical Control
NCK	Numerical Control Kernel (with block preparation, traversing range, etc.)
NCU	Numerical Control Unit: Hardware unit of the NCK
NURBS	Non Uniform Rational B–Spline
O	Output
OB	Organization Block in the PLC
OEM	Original Equipment Manufacturer: The manufacturer of equipment that is marketed by another vendor, typically under a different name.
OI	Operator Interface
OP	Operator Panel
OPI	Operator Panel Interface
P Bus	I/O (Peripherals) Bus
PC	Personal Computer
PCIN	Name of SW for exchanging data with the control system

PCMCIA	Personal Computer Memory Card International Association
PG	Programming Device
PLC	Programmable Logic Control
PP	Production Planning
RAM	Random Access Memory (read-write memory)
REF	Reference Point Approach Function
REPOS	Reposition Function
ROV	Rapid Override
RPA	R Parameter Active: Memory area in the NCK for R-NCK for R parameter numbers
RPY	Roll Pitch Yaw: Type of coordinate system rotation
RTS	Request To Send (serial data interfaces)
SBL	Single Block
SD	Setting Data
SDB	System Data Block
SEA	Setting Data Active: Identification (file type) for setting data
SFB	System Function Block
SFC	System Function Call
SK	Softkey

SKP	Skip Block
SM	Stepper Motor
SOP	Shopfloor-Oriented Programming
SPF	Sub Program File (subroutine file)
SR	Subroutine
SRAM	Static RAM (battery-backed)
STL	Statement List
SSI	Serial Synchronous Interface
SW	Software
SYF	System Files
T	Tool
TC	Tool Change
TEA	Testing Data Active: Identifier for machine data
TLC	Tool length compensation
TNRC	Tool Nose Radius Compensation
TO	Tool Offset
TOA	Tool Offset Active: Identification (file type) for tool offsets
TRANSMIT	Transform Milling into Turning: Coordinate conversion on turning machines for milling operations
TRC	Tool Radius Compensation

UFR	User Frame: Zero offset
V.24	Serial Interface (definition of interchange lines between DTE and DCE)
WCS	Workpiece Coordinate System
WPD	Work Piece Directory
ZO	Zero Offset
ZOA	Zero Offset Active: Identification (file type) for zero offset data

Notes

Terms

B

Important terms are listed below in alphabetical order, accompanied by explanations. Cross-references to other entries in this glossary are indicated by the symbol "->".

A

A spline

The A spline runs tangentially through the programmed interpolation points (3rd degree polynomial).

Absolute dimension

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

AC control

(Adaptive Control)

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

Acceleration with jerk limitation

In order to obtain the optimum acceleration gradient for the machine while providing effective protection for the mechanical components, the machining program offers a choice between instantaneous acceleration and continuous (smooth) acceleration.

Access rights

The CNC program blocks and data are protected by a 7-level system of access restrictions:

- Three password levels for system manufacturers, machine manufacturers and users and
- Four keyswitch settings which can be evaluated via the PLC.

Activate/deactivate

Working area limitation is a means of restricting the axis movement over and above the restrictions imposed by the limit switches. A pair of values delimiting the protected zone area can be specified for each axis.

Address

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

Alarms

All -> messages and alarms are displayed in plain text on the operator panel. Alarm text also includes the date, time and corresponding symbol for the reset criterion.

Alarms and messages are displayed separately.

1. Alarms and messages in the part program
   Alarms and messages can be displayed directly from the part program in plaintext.
2. Alarms and messages from PLC
   Alarms and messages relating to the machine can be displayed from the PLC program in plaintext. No additional function block packages are required for this purpose.

Analog input/output module

Analog input/output modules are signal transducers for analog process signals.

Analog input modules convert analog measured values into digital values that can be processed in the CPU.

Analog output modules convert digital values into manipulated variables.

Approach fixed machine point

Approach motion towards one of the predefined -> fixed machine points.

Archiving

Exporting files and/or directories to an **external** storage device.

Asynchronous subroutine

- A part program that can be started asynchronously (or independently) by means of an interrupt signal (e.g. ”High-speed NC input” signal) while the part program is active (SW package 3 and earlier).

- A part program that can be started asynchronously (or independently) of the current program status by means of an interrupt signal (e.g. ”High-speed NC input” signal) (SW package 4 and later).

Automatic

Control system operating mode (block-sequential to DIN): Mode in NC systems in which a -> part program is selected and continuously executed.

Auxiliary functions

Auxiliary functions can be used to pass -> parameters to the -> PLC in -> part programs, triggering reactions there which are defined by the machine manufacturer.

Axes

CNC axes are classified according to their functional scope as:

- Axes: Interpolative path axes
- Positioning axes: Non-interpolative infeed and positioning axes with axis-specific feedrates; axes can move across block limits. Positioning axes need not be involved in workpiece machining as such and include tool feeders, tool magazines, etc.

Axis address

See -> axis identifier

Axis identifier

In compliance with DIN 66217, axes are identified as X, Y and Z for a right-handed rectangular -> coordinate system.

-> Rotary axes rotating around X, Y, Z are assigned the identifiers A, B, C. Additional axes, which are parallel to those specified, can be identified with other letters.

Axis name

See -> axis identifier

Axis/spindle replacement

An axis/spindle is permanently assigned to a particular channel via machine data. This MD assignment can be ”undone” by program commands and the axis/spindle then assigned to another channel.

B

B spline
The programmed positions for the B spline are not interpolation points, but merely ”check points”. The curve generated does not pass directly through these check points, but only in their vicinity (1st, 2nd or 3rd degree polynomial).

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

Backlash compensation
Compensation of a mechanical machine backlash, e.g. backlash due to reversal of leadscrews. The backlash compensation can be entered separately for each axis.

Backup battery
The backup battery provides non-volatile storage for the -> user program in the -> CPU and ensures that defined data areas and flags, timers and counters are retentive.

Base axis
Axis whose setpoint or actual value is employed in calculating a compensatory value.

Basic coordinate system
Cartesian coordinate system, is mapped onto machine coordinate system by means of transformation.

In the -> part program, the programmer uses the axis names of the basic coordinate system. The basic coordinate system exists in parallel to the -> machine coordinate system when no -> transformation is active. The difference between the systems relates only to the axis identifiers.

Baud rate
Rate at which data transmission takes place (bit/s).

Blank
The unmachined workpiece.

Block
A section of a -> part program terminated with a line feed. A distinction is made between -> main blocks and -> subblocks.

Block
All files required for programming and program execution are known as blocks.

Block search
The block search function allows selection of any point in the part program at which machining must start or be continued. The function is provided for the purpose of testing part programs or continuing machining after an interruption.

Booting
Loading the system program after Power ON.

Bus connector
A bus connector is an S7-300 accessory that is supplied with the -> I/O modules. The bus connector extends the -> S7-300 bus from the -> CPU or an I/O module to the next adjacent I/O module.

C

C axis
Axis about which the tool spindle describes a controlled rotational and positioning movement.

C spline
The C spline is the best known and the most widely used spline. The spline passes through each of the interpolation points at a tangent and along the axis of curvature. 3rd-degree polynomials are used.

Channel structure
The channel structure makes it possible to process the -> programs of individual channels simultaneously and asynchronously.

Circular interpolation
The -> tool is required to travel in a circle between defined points on the contour at a specified feed while machining the workpiece.

Clearance control (3D), sensor-driven
A position offset for a specific axis can be controlled as a function of a measured process variable (e.g. analog input, spindle current...). This function can automatically maintain a constant clearance to meet the technological requirements of the machining operation.

CNC
-> NC

CNC high-level language
The high-level language offers: -> user variables, -> predefined user variables, -> system variables, -> indirect programming, -> arithmetic and angular functions, -> relational and logic operations, -> program jumps and branches, -> program coordination (SINUMERIK 840D), -> macros.

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

COM
Numerical control component for the implementation and coordination of communication.

Command axis
Command axes are started from synchronized actions in response to an event (command). They can be positioned, started and stopped fully asynchronous to the part program.

Compensation axis
Axis having a setpoint or actual value modified by the compensation value.

Compensation table
Table of interpolation points. It supplies the compensation values of the compensation axis for selected positions of the base axis.

Compensation value
Difference between the axis position measured by the position sensor and the desired, programmed axis position.

Connecting cables
Connecting cables are pre-assembled or user-assembled 2-wire cables with a connector at each end. They are used to connect the -> CPU via the -> multipoint interface (MPI) to a -> programming device or to other CPUs.

Continuous-path mode
The purpose of continuous-path control mode is to prevent excessive deceleration of the -> path axes at the part program block limits that could endanger the operator or the control, machine or other assets of the plant and to effect the transition to the next block at as uniform a path speed as possible.

Contour
Outline of a -> workpiece.

Contour monitoring
The following error is monitored within a definable tolerance band as a measure of contour accuracy. Overloading of the drive, for example, may result in an unacceptably large following error. In such cases, an alarm is output and the axes stopped.

Coordinate system See -> machine coordinate system, -> workpiece coordinate system

CPU Central Processor Unit -> programmable controller

Cycle Protected subroutine for executing a recurring machining operation on the -> workpiece.

Cycles support The available cycles are listed in menu ”Cycle support” in the ”Program” operating area. Once the desired machining cycle has been selected, the parameters required for assigning values are displayed in plaintext.

D

Data block
1. Data unit of the -> PLC which can be accessed by -> HIGHSTEP programs.
2. Data unit of the -> NC: Data blocks contain data definitions for global user data. These data can be initialized directly when they are defined.

Data transfer program PCIN PCIN is a routine for transmitting and receiving CNC user data, e.g. part programs, tool offsets, etc. via the serial interface. The PCIN program can run under MS-DOS on standard industrial PCs.

Data word A data unit, two bytes in size, within a -> PLC data block.

Deletion of distance-to-go Command in part program which stops machining and clears the remaining path distance to go.

Design

- The SINUMERIK FM-NC is installed in the CPU tier of the SIMATIC S7-300. The 200 mm wide, fully encapsulated module has the same external design as the SIMATIC S7-300 modules.
- The SINUMERIK 840D is installed as a compact module in the SIMODRIVE 611D converter system. It has the same dimensions as a 50 mm wide SIMODRIVE 611D module. The SINUMERIK 840D comprises the NCU module and the NCU box.
- The SINUMERIK 810D has the same design as the SIMODRIVE 611D with a width of 150mm. The following components are integrated: SIMATIC S7-CPU, 5 digital servo drive controls and 3 SIMODRIVE 611D power modules.

Diagnosis

1. Control operating area
2. The control incorporates a self-diagnosis program and test routines for servicing: Status, alarm and service displays.

Digital input/output module

Digital modules are signal transducers for binary process signals.

Dimensions in metric and inch systems

Position and lead/pitch values can be programmed in inches in the machining program. The control is set to a basic system regardless of the programmable unit of measure (G70/G71).

DRF

Differential Resolver Function NC function which generates an incremental zero offset in AUTOMATIC mode in conjunction with an electronic handwheel.

Drift compensation

When the CNC axes are in the constant motion phase, automatic drift compensation is implemented in the analog speed control. (SINUMERIK FM-NC).

Drive

- SINUMERIK FM-NC has an analog $\pm$10V interface to the SIMODRIVE 611A converter system.
- The SINUMERIK 840D control system is linked to the SIMODRIVE 611D converter system via a high-speed digital parallel bus.

E

Editor
The editor makes it possible to create, modify, extend, join and insert programs/texts/program blocks.

Electronic handwheel
Electronic handwheels can be used to traverse the selected axes simultaneously in manual mode. The handwheel clicks are analyzed by the increment analyzer.

Exact stop
When an exact stop is programmed, a position specified in the block is approached accurately and, where appropriate, very slowly. In order to reduce the approach time, -> exact stop limits are defined for
rapid traverse and feed.

Exact stop limit
When all path axes reach their exact stop limits, the control responds as if it had reached its destination point precisely. The -> part program continues execution at the next block.

External zero offset
A zero offset specified by the -> PLC.

F

Fast retraction from contour
When an interrupt is received, it is possible to initiate a motion via the CNC machining program which allows the tool to be retracted quickly from the workpiece contour currently being machined. The retraction angle and the distance retracted can also be parameterized. An interrupt routine can be executed after the rapid retraction. (SINUMERIK FM-NC, 810D, 840D).

Feedforward control, dynamic
Contour inaccuracies resulting from following errors can be almost completely eliminated by the dynamic, acceleration-dependent feedforward control function. Feedforward control ensures an excellent degree of machining accuracy even at high tool path velocities. Feedforward control can only be selected or deselected for all axes together via the part program.

Feedrate override

The current feedrate setting entered via the control panel or by the PLC is overlaid on the programmed feedrate (0–200 %). The feedrate can also be corrected by a programmable percentage factor (1–200 %) in the machining program.

An offset can also be applied via motion-synchronous actions independently of the running program.

Finished-part contour

Contour of the finished workpiece. See also -> blank.

Fixed machine point

A point defined uniquely by the machine tool, such as the reference point.

Fixed-point approach

Machine tools can execute defined approaches to fixed points such as tool-change points, loading points, pallet-change points, etc. The coordinates of these points are stored on the control. Where possible, the control moves these axes in -> rapid traverse.

Frame

A frame is a calculation rule that translates one Cartesian coordinate system into another Cartesian coordinate system. A frame contains the components -> zero offset, -> rotation, -> scaling and -> mirroring.

G

General reset

The following memories of the -> CPU are erased by a general reset operation:

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

Geometry

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

Geometry axis

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

Global main run/subroutine

Each global main run/subroutine can be stored only once under its name in the directory. However, the same name can be used in different directories.

Ground

”Ground” is the term applied to all the electrically inactive, interconnected parts of a piece of equipment which cannot carry any hazardous contact voltage even in the event of a fault.

H

Helical interpolation

The helical interpolation function is ideal for machining internal and external threads using form milling cutters and for milling lubrication grooves. The helix comprises two movements:

1. Circular movement in one plane
2. Linear movement perpendicular to this plane.

High-speed digital inputs/outputs

As an example, high-speed CNC program routines (interrupt routines) can be started via the digital inputs. High-speed, program-driven switching functions can be initiated via the digital CNC outputs (SINUMERIK 840D). (SINUMERIK 840D).

HIGHSTEP

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

I

Identifier

In accordance with DIN 66025, identifiers (names) for variables (arithmetic variables, system variables, user variables), for subroutines, for vocabulary words and for words can contain several address letters. These letters have the same meaning as the words in the block syntax. Identifiers must be unique. Identical identifiers must not be used for different objects.

Inch system of measurement

System of measurement that defines distances in ”inches” and fractions thereof.

Inclined axis

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

Increment

A destination for axis traversal is defined by a distance to be covered and a direction referenced to a point already reached. See also -> absolute dimension.

Increment

Travel path length specification based on number of increments. The number of increments can be stored as a -> setting data or selected with keys labeled with 10, 100, 1000, 10 000.

Initialization block

Initialization blocks are special -> program blocks. They contain values which must be assigned before the program is executed. Initialization blocks are used primarily for initializing predefined data or global user data.

Initialization file

An initialization file can be created for each -> workpiece. In it, the various variable value instructions which apply exclusively to one workpiece can be stored.

Intermediate blocks

Movements with selected tool offset (G41/G42) can be interrupted by a limited number of intermediate blocks (blocks without axis motions in the offset plane). When such blocks are used, the tool offset can still be calculated correctly. The permissible number of intermediate blocks read in advance by the control can be set via system parameters.

Interpolation cycle

The interpolation cycle is a multiple of the basic system cycle. It specifies the cycle time for updating the setpoint interface to the position controllers. The interpolation cycle determines the resolution of the velocity profiles.

Interpolative compensation

Interpolative compensation provides a means of compensating for leadscrew errors (LEC) and measuring-system errors (MSEC) resulting from the production process.

Interpolator

Logical unit of the -> NCK which determines intermediate values for the movements to be traversed on the individual axes on the basis of destination positions specified in the part program.

Interrupt routine

Interrupt routines are special -> subroutines which can be started by events (external signals) in the machining process. The part program block being processed is aborted and the axis position at the instant of interruption is stored automatically.

See -> ASUB

Inverse-time feedrate

On SINUMERIK FM-NC and 840D controls, it is possible to program the time required to traverse the path of a block instead of the feedrate speed for the axis movement (G93).

I/O module

I/O modules create the link between the CPU and the process. I/O modules are:

- ->Digital input/output modules
- ->Analog input/output modules
- ->Simulator modules

J

Jog

Control system operating mode (setup): The machine can be set up in Jog mode. Individual axes and spindles can be jogged by means of direction keys. Other functions in Jog mode are -> reference point approach, -> Repos and -> Preset -> (set actual value).

K

Keyswitch

1. **S7-300**: The keyswitch is the mode selector switch on the -> CPU. The keyswitch is operated by means of a removable key.
2. **840D/FM-NC**: The keyswitch on the -> machine control panel has 4 positions which are assigned functions by the operating system of the control. There are also three keys of different colors belonging to the keyswitch that can be removed in the specified positions.

$K_Ü$

Transmission Ratio

K_v

Servo gain factor, control variable of a control loop

L

Languages

The user interface texts, system messages and alarms are available in five system languages (floppy disk):
German, English, French, Italian and **Spanish**.
The user can select **two** of the listed languages at a time in the control.

Leadscrew error compensation

Compensation of mechanical inaccuracies in a leadscrew involved in the feed motion. Errors are compensated by the control based on stored deviation measurements.

Limit speed

Minimum/maximum (spindle) speed: The maximum speed of a spindle can be limited by values defined in the machine data, the -> PLC or -> setting data.

Linear axis

The linear axis is an axis which, in contrast to a rotary axis, describes a straight line.

Linear interpolation

The tool travels along a straight line to the destination point while machining the workpiece.

Look Ahead

The **Look Ahead** function is a means of optimizing the machining velocity by looking ahead over a parameterizable number of traversing blocks.

Look Ahead for contour violations

The control detects and reports the following types of collision:

1. Path is shorter than tool radius.
2. Width of inside corner is less than the tool diameter.

M

Machine

Control operating area

Machine axes

Axes which exist physically on the machine tool.

Machine control panel

An operator panel on a machine tool with operating elements such as keys, rotary switches, etc. and simple indicators such as LEDs. It is used for direct control of the machine tool via the PLC.

Machine coordinate system — System of coordinates based on the axes of the machine tool.

Machine zero — A fixed point on the machine tool which can be referenced by all (derived) measurement systems.

Machining channel — A channel structure makes it possible to reduce downtimes by allowing sequences of motions to be executed in parallel. For example, a loading gantry can execute its movements during a machining operation. In this case, a CNC channel ranks as an autonomous CNC control complete with decoding, block preparation and interpolation.

Macros — Multiple programming language instructions can be combined in a single statement. This abbreviated sequence of instructions is called in the CNC program under a user-defined name. The macro executes the instructions sequentially.

Main block — A block prefixed by ":" containing all the parameters required to start execution of a -> part program.

Main program — -> Part program identified by a number or name in which other main programs, subroutines or -> cycles may be called.

Main run — Part program blocks which have been decoded and prepared by the preprocessor are executed during the "main run".

MDA — Control system operating mode: Manual Data Automatic. In the MDA mode, individual program blocks or block sequences with no reference to a main program or subroutine can be input and executed immediately afterwards through actuation of the NC Start key.

Measuring circuits

- SINUMERIK FM-NC: The requisite control circuits for axes and spindles are integrated in the control module as standard. A maximum total of 4 axes and spindles can be implemented, with no more than 2 spindles.
- SINUMERIK 840D: The signals from the sensors are analyzed in the SIMODRIVE 611D drive modules. The maximum total configuration is 8 axes and spindles, with no more than 5 spindles.

Messages

All messages programmed in the part program and -> alarms detected by the system are displayed in plain text on the operator panel. Alarms and messages are displayed separately.

Metric system

Standardized system of units for lengths in millimeters (mm), meters (m), etc.

Mirroring

Mirroring exchanges the leading signs of the coordinate values of a contour in relation to an axis. Mirroring can be performed simultaneously in relation to several axes.

Mode

An operating concept on a SINUMERIK control. The modes -> Jog, -> MDA, -> Automatic are defined.

Mode group

All axes/spindles are assigned to one and only one channel at any given time. Each channel is assigned to a mode group. The same -> mode is always assigned to the channels of a mode group.

Motion synchronization

This function can be used to initiate actions that are synchronized with the machining operation. The starting point of the actions is defined by a condition (e.g. status of a PLC input, time elapsed since beginning of a block). The start of motion-synchronous actions is not tied to block boundaries. Examples of typical motion-synchronous actions are:
Transfer M and H auxiliary functions to the PLC or deletion of distance-to-go for specific axes.

Multipoint interface

The multipoint interface (MPI) is a 9–pin sub–D port. A parameterizable number of devices can be connected to an MPI for the purpose of communicating with one another:

- Programming devices
- HMI systems
- Other automation systems

The ”Multipoint Interface MPI” parameter block of the CPU contains the –> parameters which define the properties of the multipoint interface.

N

NC

Numerical Control It incorporates all the components of the machine tool control system: –> NCK, –> PLC, –> MMC, –> COM.
Note: CNC (computerized numerical control) would be a more appropriate description for the SINUMERIK 840D or FM–NC controls. computerized numerical control.

NCK

Numerical Control Kernel: Component of the NC control which executes –> part programs and essentially coordinates the movements on the machine tool.

Network

A network is the interconnection of several S7–300s and other terminal devices such as a programming device, for example, interlinked by means of –> connecting cables. The networked devices interchange data via the network.

Node number

The node number is the ”contact address” of a –> CPU or the –> programming device or another intelligent I/O module if these devices are exchanging data with one another via a –> network. The node number is assigned to the CPU or the programming device by the S7 tool –> ”S7 Configuration”.

NRK

Numeric Robotic Kernel (operating system of the –> NCK)

NURBS

Motion control and path interpolation are implemented internally in the control on the basis of NURBS (Non–Uniform Rational B Splines). A standard procedure is thus available (SINUMERIK 840D) as an internal control function for all modes of interpolation.

O

Oblique-plane machining

Drilling and milling operations on workpiece surfaces which are oblique to the coordinate planes of the machine are supported by the ”Oblique surface machining” function. The position of the oblique plane can be defined by inclining the coordinate system (see FRAME programming).

OEM

The scope for implementing individual solutions (OEM applications) for the SINUMERIK 840D has been provided for machine manufacturers who wish to create their own operator interface or integrate process-oriented functions in the control.

Offset memory

Data area in the control in which tool offset data are stored.

Online tool offset

This function can be used for grinding tools only.

The reduction in size of the grinding wheel resulting from dressing is transferred as a tool offset to the currently active tool and immediately applied.

Operator interface

The operator interface (OI) is the human-machine interface of a CNC. It takes the form of a screen and has eight horizontal and eight vertical softkeys.

Oriented spindle stop

Stops the workpiece spindle at a specified orientation angle, e.g. to perform an additional machining operation at a specific position.

Oriented tool retraction

RETTOOL: If machining is interrupted (e.g. when a tool breaks), a program command can be used to retract the tool in a user-specified orientation by a defined distance.

Override

Manual or programmable control feature which enables the user to override programmed feedrates or speeds in order to adapt them to a specific workpiece or material.

P

Parameters

1. **S7-300**: The S7-300 uses two types of parameter:
   - Parameter of a STEP 7 statement
     A parameter of a STEP 7 statement is the address of the operand to be processed or a constant.
   - Parameter of a -> parameter block
     A parameter of a parameter block determines the behavior of a module.
2. **840D/810D**:
   - Control operating area
   - Computation parameter, can be set any number of times or queried by the programmer for any purpose in the part program.

Part program

A sequence of instructions to the NC control which combine to produce a specific -> workpiece by performing certain machining operations on a given -> blank.

Part program management

The part program management function can be organized according to -> workpieces. The quantity of programs and data to be managed is dependent on the control memory capacity and can also be configured via MD settings. Each file (programs and data) can be given a name consisting of a maximum of 16 alphanumeric characters.

Path axis

Path axes are all the machining axes in the -> channel which are controlled by the -> interpolator such that they start, accelerate, stop and reach their end positions simultaneously.

Path feed

The path feed acts on -> path axes. It represents the geometrical sum of the feeds on the participating -> path axes.

Path velocity

The maximum programmable path velocity depends on the input resolution. With a resolution of 0.1 mm, for example, the maximum programmable path velocity is 1000 m/min.

PG

Programming Device

PLC

Programmable Logic Control -> Speicherprogrammierbare Steuerung. Component of the -> NC: Programmable controller for processing the control logic on the machine tool.

PLC program memory

- SINUMERIK FM-NC: The PLC user program, the user data and the basic PLC program are stored together in the PLC user memory of the CPU 314.
  S7-CPU314 has a user memory of 24 KB for this purpose.
- SINUMERIK 840D: The PLC user program, the user data and the basic PLC program are stored together in the PLC user memory. The PLC user memory can be expanded up to 128 KB.
- SINUMERIK 810D: The PLC user program, the user data and the basic PLC program are stored together in the PLC user memory of the CPU 314. The basic
  version of the S7-CPU314 has a user memory of
  64 KB which can be optionally expanded up to 128 KB.

PLC programming

The PLC is programmed with the **STEP 7** software. The STEP 7 programming software is based on the standard **WINDOWS** operating system and incorporates the functionality of STEP 5 programming with innovative expansions and developments.

Polar coordinates

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

Polynomial interpolation

Polynomial interpolation provides a means of generating a very wide range of curves, including **straight-line, parabolic and exponential functions** (SINUMERIK 840D/810D).

Positioning axis

An axis which performs an auxiliary movement on a machine tool (e.g. tool magazine, pallet transport). Positioning axes are axes that do not interpolate with the -> path axes.

Power ON

The action of switching the control off and then on again.

Preprocessing memory, dynamic

The traversing blocks are preprocessed prior to execution and stored in a ”preprocessing memory”. Block sequences can be executed at a very fast rate from the memory. Blocks are uploaded continuously to the preprocessing memory during machining.

Preprocessing stop

Program command. The next block in a part program is not executed until all other blocks which have already been preprocessed and stored in the preprocessing memory have been executed.

See also ”Preprocessing memory”.

Preset

The control zero point can be redefined in the machine coordinate system by means of the Preset function. Preset does not cause the axes to move; instead, a new position value is entered for the current axis positions.

Program

1. Control operating area
2. Sequence of instructions to the control system.

Programmable frames

Programmable -> frames can be used to define new coordinate system starting points dynamically while the part program is running. A distinction is made between absolute definition using a new frame and additive definition with reference to an existing starting point.

Programmable logic controller

Programmable logic controllers (PLC) are electronic controllers whose functions are stored as a program in the control unit. The design and wiring of the unit are not, therefore, dependent on the control functions. Programmable logic controllers have the same structure as a computer, i.e. they consist of a CPU with memory, input/output modules and an internal bus system. The I/Os and programming language are selected according to the requirements of the control technology involved.

Programmable working area limitation

Limitation of the movement area of the tool to within defined, programmable limits.

Programming key
Characters and character sequences which have a defined meaning in the programming language
for -> part programs (see Programming Guide).

Protection zone
Three-dimensional area within a -> working area which the tool tip is not permitted to enter (programmable via MD).

Q

Quadrant error compensation
Contour errors on quadrant transitions caused by frictional fluctuations on guideways can be largely eliminated by means of quadrant error compensation. A circularity test is performed to parameterize the quadrant error compensation function.

R

R parameter
Calculation parameter. The programmer can assign or request the values of the R parameter in the -> part program as required.

Rail
This rail is used to mount the modules of the S7-300 system.

Rapid traverse
The highest traversing speed of an axis used, for example, to bring the tool from an idle position to the -> workpiece
contour or retract it from the workpiece contour.

Reference point
Point on the machine tool with which the measuring system of the -> machine axes is referenced.

Reference point approach
If the position measuring system used is not an absolute-value encoder, then a reference point approach operation is required to ensure that the actual values supplied by the measuring system are in accordance with the machine coordinate values.

REPOS

1. Reapproach contour, triggered by operator
   REPOS allows the tool to be returned to the interrupt position by means of the direction keys.

2. Programmed contour reapproach
   A selection of approach strategies are available in the form of program commands: Approach point of interruption, approach start of block, approach end of block, approach a point on the path between start of block and interruption.

Revolutional feedrate

The axis feedrate is adjusted as a function of the speed of the master spindle in the channel (programmed with G95).

Rigid tapping

This function is used to tap holes without the use of a compensating chuck. The spindle is controlled as an interpolative rotary axis and drill axis, with the result that threads are tapped precisely to the final drilling depth, for example, in blind tapped holes (precondition: Spindle axis mode).

Rotary axis

Rotary axes cause the tool or workpiece to rotate to a specified angle position.

Rotary axis, continuously turning

The range of motion of a rotary axis can be set to a modulo value (in machine data) or defined as continuous in both directions, depending on the application. Continuously turning rotary axes are used, for example, for eccentric machining, grinding and winding.

Rotation

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

Rounding axis

Rounding axes cause the workpiece or tool to rotate to an angle position described on a graduated grid. When the grid position has been reached, the axis is ”in position”.

S

S7 Configuration

S7 Configuration is a tool for parameterizing modules. S7 Configuration is used to set a variety of
-> parameter blocks of the -> CPU and the I/O modules on the

-> programming device. These parameters are uploaded to the CPU.

S7-300 bus

The S7-300 bus is a serial data bus which supplies modules with the appropriate voltage and via which they exchange data with one another. The connection between the modules is made by means of -> bus connectors.

Safety functions

The control includes continuously active monitoring functions which detect faults in the -> CNC, the programmable controller (-> PLC) and the machine so early that damage to the workpiece, tool or machine rarely occurs. In the event of a fault, the machining operation is interrupted and the drives stopped. The cause of the malfunction is logged and an alarm issued. At the same time, the PLC is notified that a CNC alarm is pending.

Safety Integrated

Effective personnel and machine protection integrated in the control in conformance with EC Directive >>89/392/EEC<< in >>Safety Category 3<< to EN-954-1 (Categories B. 1-4 are defined in this standard) for safe setup and testing.

Discrete fail-safety is assured. If an individual fault occurs, the safety function is still effective.

Scaling

Component of a -> frame which causes axis-specific scale alterations.

Services

Control operating area

Setting data

Data which provide the control with information about properties of the machine tool in a way defined by the system software.

Unlike -> machine data, setting data can be modified by the user.

Softkey

A key whose name appears on an area of the screen. The choice of softkeys displayed is adapted dynamically to the operating situation. The freely assignable function keys (softkeys) are assigned to functions defined in the software.

Software limit switches

Software limit switches define the limits of the travel range of an axis and prevent the slide contacting the hardware limit switches. Two pairs of values can be assigned per axis and activated separately via the -> PLC.

Spindles

The spindle functionality is a two-level construct:

1. Spindles: Speed-controlled or position-controlled spindle drives, analog
   digital (SINUMERIK 840D)
2. Auxiliary spindles: Speed-controlled spindle drives without actual position sensor, e.g. for power tools. ”Auxiliary spindle” function package, e.g. for power tools.

Spline interpolation

Using the spline interpolation function, the control is able to generate a smooth curve from just a small number of specified interpolation points along a setpoint contour.

Standard cycles

Standard cycles are used to program machining operations which repeat frequently:

- For drilling/milling
- For measuring tools and workpieces

The available cycles are listed in menu ”Cycle support” in the ”Program” operating area. Once the desired machining cycle has been selected, the parameters required for assigning values are displayed in plaintext.

Subblock

Block prefixed by ”N” containing information for a machining step such as a position parameter.

Subroutine

A sequence of instructions of a -> part program which can be called repeatedly with different initial parameters. A subroutine is called from within a main program. Every subroutine can be locked against unauthorized export and viewing (with MMC 102/103). -> Cycles are a type of subroutine.

Synchronization

Instructions in -> part programs for coordination of the operations in different -> channels at specific machining points.

Synchronized actions

1. Auxiliary function output
   While a workpiece is being machined, technological functions (-> auxiliary functions) can be output from the CNC program to the PLC. These auxiliary functions control, for example, ancillary equipment on the machine tool such as the sleeve, gripper, chuck, etc.

2. High-speed auxiliary function output
   The acknowledgement times for the -> auxiliary functions can be minimized and unnecessary halts in the machining process avoided for time-critical switching functions.

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

Synchronized axes

Synchronized axes require the same amount of time to traverse their path as -> geometry axes for their path.

Synchronous spindle

Accurate angular synchronism between one master spindle and one or more slave spindles. Enables flying transfer of a workpiece from spindle 1 to spindle 2 on turning machines.

In addition to speed synchronism, it is also possible to program the relative angular positions of the spindles, e.g. on-the-fly, position-oriented transfer of inclined workpieces.

Several pairs of synchronous spindles can be implemented.

System variable

A variable which exists although it has not been programmed by the -> part program programmer. It is defined by the data type and the variable name, which is prefixed with **$**. See also -> User-defined variable.

T

Teach In

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

Text editor

-> Editor

Tool

A tool employed to shape the workpiece, for example, a turning tool, milling cutter, drill, laser beam, grinding wheel, etc.

Tool nose radius compensation

A contour is programmed on the assumption that a pointed tool will be used. Since this is not always the case in practice, the curvature radius of the tool being used is specified so that the control can make allowance for it. The curvature centre point is guided equidistantly to the contour at an offset corresponding to the curvature radius.

Tool offset

A tool is selected by programming a **T function** (5 decades, integer) in the block. Up to nine tool edges (D addresses) can be assigned to each T number. The number of tools to be managed in the control is set in parameterization.

Tool length compensation is selected by programming D numbers.

Tool radius compensation

In order to program a desired -> workpiece contour directly, the control must traverse a path equidistant to the programmed contour, taking into account the radius of the tool used (G41/G42). (G41/G42).

Transformation

Programming in a Cartesian coordinate system, execution in a non-Cartesian coordinate system (e.g. with machine axes as rotary axes).

Employed in conjunction with Transmit, Inclined Axis, 5-Axis Transformation.

Transmit

This function is used to mill the outside contours on turned 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 benefits afforded by Transmit are simplified programming and improved machine efficiency through complete machining: Turning and milling on the same machine without reclamping.

Travel to fixed stop

This function allows axes (tailstocks, sleeves) to be traversed to a fixed stop position in order, for example, to clamp workpieces. The contact pressure can be defined in the part program.

Traversing range

The maximum permissible travel range for linear axes is ± 9 decades. The absolute value depends on the selected input and position control resolution and the unit of measurement (inch or metric).

U

User-defined variable

Users can define variables in the -> part program or data block (global user data) for their own use. A definition contains a data type specification and the variable name. See also -> system variable.

User memory

All programs and data such as part programs, subroutines, comments, tool offsets, zero offsets/frames and channel and program user data can be stored in the common CNC user memory.

User program

-> Part program

V

Variable definition

A variable is defined through the specification of a data type and a variable name. The variable name can be used to address the value of the variable.

Velocity control

In order to achieve an acceptable travel velocity in movements which call for very small adjustments of position in a block, the control can -> look ahead.

Vocabulary words

Words with a specific notation which have a defined meaning in the programming language for -> part programs.

W

Working memory The working storage is a Random Access Memory in the -> CPU which the processor accesses as it executes the application program.

Working space Three-dimensional zone into which the tool tip can be moved on account of the physical design of the machine tool.
See also -> protection zone.

Workpiece Part to be produced/machined by the machine tool.

Workpiece contour Setpoint contour of the -> workpiece to be produced/machined.

Workpiece coordinate system The origin of the workpiece coordinate system is the ->workpiece zero. In machining operations programmed in the workpiece coordinate system, the dimensions and directions refer to this system.

Workpiece zero The workpiece zero is the origin for the -> workpiece coordinate system. It is defined by its distance from the machine zero.

X

Y

Z

Zero offset

Specification of a new reference point for a coordinate system through reference to an existing zero and a -> frame.

1. Settable
   SINUMERIK 840D: A parameterizable number of settable zero offsets is available for each CNC axis. Each of the zero offsets can be selected by G functions and selection is exclusive.
2. External
   All offsets which define the position of the workpiece zero can be overlaid with an external zero offset
   - defined by handwheel (DRF offset) or
   - defined by the PLC.
3. Programmable
   Zero offsets can be programmed for all path and positioning axes by means of the TRANS instruction.

G Code Table

C

Appendix C describes the G codes and the functions.

C.1 G code table

Table C-1 G code table

G code		Description	840D sl	802D sl
Group 1				
G00 [1)]	1	Rapid traverse	x	x
G01	2	Linear motion	x	x
G02	3	Circle/helix, clockwise	x	x
G02.2	6	Involute interpolation, clockwise	x	x
G03	4	Circle/helix, counterclockwise	x	x
G03.2	7	Involute interpolation, counterclockwise	x	x
G33	5	Thread cutting with constant lead	x	x
Group 2				
G17 [1)]	1	XY plane	x	x
G18	2	ZX plane	x	x
G19	3	YZ plane	x	x
Group 3				
G90 [1)]	1	Absolute programming	x	x
G91	2	Incremental programming	x	x
Group 4				
G22	1	Working area limitation, protection zone 3 on	x	x
G23 [1)]	2	Working area limitation, protection zone 3 off	x	x
Group 5				
G93	3	Inverse time feed (rev/min)	x	x
G94 [1)]	1	Feed in [mm/min, inch/min]	x	x
G95	2	Feed in [mm/rev, inch/rev]	x	x
Group 6				
G20[1)] (G70)	1	Input system inch	x	x
G21 (G71)	2	Input system metric	x	x

Table C-1 G code table

G code		Description	840D sl	802D sl
Group 7				
G40 [1]	1	Deselect cutter radius compensation	x	x
G41	2	Compensation to left of contour	x	x
G42	3	Compensation to right of contour	x	x
Group 8				
G43	1	Tool length compensation positive on	x	x
G44	2	Tool length compensation negative on	x	x
G49 [1]	3	Tool length compensation off	x	x
Group 9				
G73	1	High speed deep hole drilling cycle with chipbreaking	x	x
G74	2	Counterclockwise tapping cycle	x	x
G76	3	Fine drilling cycle	x	x
G80 [1]	4	Cycle off	x	x
G81	5	Counterbore drilling cycle	x	x
G82	6	Counterboring drilling cycle	x	x
G83	7	Deep hole drilling cycle with swarf removal	x	x
G84	8	Clockwise tapping cycle	x	x
G85	9	Drilling cycle	x	x
G86	10	Drilling cycle, retract using G00	x	x
G87	11	Back boring cycle	x	x
G89	12	Drilling cycle, retract using G01	x	x
Group 10				
G98 [1]	1	Return to starting point for fixed cycles	x	x
G99	2	Return to point R for fixed cycles	x	x
Group 11				
G50 [1][2]	1	Scaling off	x	x
G51[2]	2	Scaling on	x	x
Group 12				
G66[2]	1	Modal macro call	x	x
G67 [1][2]	2	Delete modal macro call	x	x
Group 13				
G96	1	Constant cutting rate on	x	x
G97 [1]	2	Constant cutting rate off	x	x
Group 14				
G54 [1]	1	Select zero offset	x	x
G55	2	Select zero offset	x	x

Table C-1 G code table

G code		Description	840D sl	802D sl
G56	3	Select zero offset	x	x
G57	4	Select zero offset	x	x
G58	5	Select zero offset	x	x
G59	6	Select zero offset	x	x
G54P{1...48}	1	Extended zero offsets	x	x
G54.1	7	Extended zero offset	x	x
G54 P0	1	“externel zero offsets EXOFS”	x	x
Group 15				
G61	1	Exact stop modal	x	x
G62	4	Automatic corner override	x	x
G63	2	Tapping mode	x	x
G64[1)]	3	Continuous-path mode	x	x
Group 16				
G68 [2)]	1	Coordinate rotation ON, 2D/3D	x	--
G69 [2)]	2	Coordinate rotation OFF	x	--
Group 17				
G15 [1)]	1	Polar coordinates off	x	x
G16	2	Polar coordinates on	x	x
Group 18 (non-modal)				
G04	1	Dwell	x	x
G05	18	High-speed cycle cutting	x	x
G05.1[2)]	22	High-speed cycle -> Calling CYCLE305	x	x
G07.1[2)]	16	Cylindrical interpolation	x	x
G08	12	Look-ahead control	x	--
G09	2	Exact stop	x	x
G10[2)]	3	Write zero offset/tool offset	x	x
G10.6	17	Lift off from contour (POLF)	x	x
G11	4	Terminate parameter input	x	x
G27	13	Referencing check	x	x
G28	5	Approach 1st reference point	x	x
G30	6	Approach 2nd/3rd/4th reference point	x	x
G30.1	19	Floating reference position	x	x
G31	7	Measurement with touch-trigger probe	x	x
G52	8	Additive zero offset	x	x
G53	9	Approach position in machine coordinate system	x	x
G60	22	Oriented positioning	x	x

Table C-1 G code table

G code		Description	840D sl	802D sl
G65[2)]	10	Call macro	x	x
G72.1[2)]	14	Contour repeating - rotational copy	x	--
G72.2[2)]	15	Contour repeating - linear copy	x	--
G92	11	Preset actual value memory / spindle speed limitation	x	x
G92.1	21	Delete actual value memeory, reset of WCS	x	x
Group 22				
G50.1	1	Programmable mirror image cancel	x	x
G51.1	2	Programmable mirror image	x	x
Group 25				
G13.1	1	Polar coordinate interpolation mode cancel	x	x
G12.1	2	Polar coordinate interpolation mode	x	x
Group 31				
G290 [1)]	1	Select Siemens mode	x	x
G291	2	Select ISO dialect mode	x	x
x means that G-Code is applicable, -- means that G-Code is not applicable				

Note: In general, the NC establishes the G code modes identified by 1), when the power is turned ON or when the NC is reset. However, please refer to the machine tool builders documentation for actual setting.
The G codes identified by 2) are optional. Please refer to the machine tool builders documentation for the availability of the function.

D.1 Machine/Setting data

10604 MD number	**WALIM_GEOAX_CHANGE_MODE** Work area limitation when switching geometrical axes		
Default setting: 0	Minimum input limit: 0	Maximum input limit: 1	
Changes effective after Power On	Protection level: 2/7	Unit: -	
Data type: BYTE	Applies with effect from SW version: 6.2		
Meaning:	Retain or deactivate work area limitation when switching geometrical axes. The MD is bit-coded and has the following meaning: Bit = =0: Deactivate work area limitation when switching geometrical axes =1: Retain work area limitation when switching geometrical axes		
This MD cannot SINUMERIK 802D sl.			

10615 MD number	**NCFRAME_POWERON_MASK** Delete global base frames on Power On		
Default setting: 0	Minimum input limit: 0	Maximum input limit: 0	
Changes effective after Power On	Protection level: 2/7	Unit: -	
Data type: DWORD	Applies with effect from SW version: 5.2		
Meaning:	This machine data defines whether global base frames are deleted on a Power On reset. The selection can be made separately for the individual base frames. Bit 0 corresponds to base frame 0, bit 1 to base frame 1, etc. 0: Base frame is retained on Power On 1: Base frame is deleted on Power On.		
This MD cannot SINUMERIK 802D sl.			

10652	**CONTOUR_DEF_ANGLE_NAME**		
MD number	Definable name for angle in the contour short description		
Default setting: "ANG"	Minimum input limit: -	Maximum input limit: -	
Changes effective after Power On	Protection level: 2/7	Unitt: -	
Datentype: STRING	Applies with effect from SW version: 5		
Meaning:	The setting is effective for Siemens G code programming only, i.e. G290. The name used to program the angle in the contour short description is definable. This allows, for example, identical programming in different language modes: If the angle is named "A", it is programmed in the same way with Siemens and ISO Dialect0. The name must be unique, i.e. axes, variables, macros, etc. must not exist with the same name.		
This MD cannot SINUMERIK 802D sl.			

10654	**RADIUS_NAME**		
MD number	Definable name for radius non-modally in the contour short description		
Default setting: "RND"	Minimum input limit: -	Maximum input limit: -	
Changes effective after Power On	Protection level: 2/7	Unit: -	
Data type: STRING	Applies with effect from SW version: 5		
Meaning:	The name used to program the radius in the contour short description is definable. This allows, for example, identical programming in different language modes: If the radius is named "R", it is programmed in the same way with Siemens and ISO Dialect0. The name must be unique, i.e. axes, variables, macros, etc. must not exist with the same name. The setting is effective for Siemens G code programming, i.e. G290.		
This MD cannot SINUMERIK 802D sl.			

10656	**CHAMFER_NAME**		
MD number	Definable name for chamfer in the contour short description		
Default setting: "CHR"	Minimum input limit: -	Maximum input limit: -	
Changes effective after Power On	Protection level: 2/7	Unit: -	
Data type: STRING	Applies with effect from SW version: 5		
Meaning:	The name used to program the chamfer in the contour short description is definable. This allows, for example, identical programming in different language modes: If the chamfer is named "C", it is programmed in the same way with Siemens and ISO Dialect0. The name must be unique, i.e. axes, variables, macros, etc. must not exist with the same name. The setting is effective for Siemens G code programming, i.e. G290. The chamfer in the original direction of movement. Alternatively, the chamfer length can be programmed with the name CHF.		
This MD cannot SINUMERIK 802D sl.			

10704	DRYRUN_MASK		
MD number	Activating dry run feed		
Default setting:	Minimum input limit: -	Maximum input limit: -	
Changes effective after:	Protection level:	Unit: -	
Data type: BYTE	Applies with effect from SW version:		
Meaning:	DRYRUN_MASK == 0 Dryrun must only be activated or deactivated at the end of a block. DRYRUN_MASK == 1 Dry run feed may be activated or deactivated even during program execution Note: Once dry run feed has been activated, the axes are stopped for the duration of the reorganization. DRYRUN_MASK == 2 Dryrun can be activated or deactivated in any phase and the axes are not stopped. **Note:** However, the function is only effective upon using a block which comes ”later” in the program run. The function takes effect on the next (implicit) Stop Reset.		
This MD cannot SINUMERIK 802D sl.			

10706	SLASH_MASK		
MD number	Activating the block skip function		
Default setting:	Minimum input limit: -	Maximum input limit: -	
Changes effective after:	Protection level:	Unit: -	
Data type: BYTE	Applies with effect from SW version:		
Meaning:	SLASH_MASK == 0 The block skip function can only be switched over at the end of a block. SLASH_MASK == 1 When SLASH_MASK == 1 the block skip function may be activated even during program execution. Note: Once block skip has been activated, the axes are stopped for the duration of the reorganization. SLASH_MASK == 2 Block switchover is possible in any phase. **Note:** However, the function is only effective upon using a block which comes ”later” in the program run. The function takes effect on the next (implicit) Stop Reset.		
This MD cannot SINUMERIK 802D sl.			

10715 MD number	**M_NO_FCT_CYCLE[0]** M function number for cycle call		
Default setting: -1	Minimum input limit: -1	Maximum input limit: -	
Changes effective after Power On	Protection level: 2/7	Unit: -	
Data type: DWORD	Applies with effect from SW version: 5.2		
Meaning:	M number with which a subprogram is called. 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, the subprogram defined in M_NO_FCT_CYCLE_NAME is started at the end of the block. If the M function is programmed again in the subprogram, the substitution no longer takes place by means of a subprogram call. $MN_M_NO_FCT_CYCLE is effective both in Siemens mode G290 and in external language mode G291. A subprogram call may not be superimposed on M functions with fixed meanings. In the event of a conflict, alarm 4150 is output: - M0 to M5, - M17, M30, - M40 to M45, - M function for spindle/axis mode switchover according to $MC_SPIND_RIGID_TAPPING_M_NR (default M70) - M functions for nibbling/punching according to configuration via $MC_NIBBLE_PUNCH_CODE if activated via $MC_PUNCHNIB_ACTIVATION. - With applied external language ($MN_MM_EXTERN_LANGUAGE) M19, M96-M99. Exception: The M functions defined for the tool change with $MC_TOOL_CHANGE_M_CODE. $MN_M_NO_FCT_CYCLE_NAME and $MN_T_NO_FCT_CYCLE_NAME may not be active in the same block (part program line), i.e. only one M/T function substitution can be active per block. Neither an M98 call nor a modal subprogram call can be programmed in the block with the M function substitution. A subprogram return jump or end of part program is not allowed. Alarm 14016 is output in the event of a conflict.		

10716	M_NO_FCT_CYCLE_NAME[0]		
MD number	Name of tool-changing cycle for M functions from MD $MN_MFCT_CYCLE		
Default setting: -	Minimum input limit: -	Maximum input limit: -	
Changes effective after Power On	Protection level: 2/7	Unit: -	
Data type: STRING	Applies with effect from SW version: 5.2		
Meaning:	The name of the cycle is stored in the machine data. This cycle is called when the M function from machine data $MN_M_NO_FCT_CYCLE is programmed. If the M function is programmed in a motion block, the cycle is executed after the move-ment. $MN_M_NO_FCT_CYCLE is effective both in Siemens mode G290 and in external language mode G291. If a T number is programmed in the calling block, the programmed T number can be scanned in the cycle in variable $P_TOOL. $MN_M_NO_FCT_CYCLE_NAME and $MN_T_NO_FCT_CYCLE_NAME may not be active in the same block, i.e. only one M/T function substitution can be active per block. Neither an M98 call nor a modal subprogram call can be programmed in the block with the T function substitution. A subprogram return jump or end of part program is not allowed. Alarm 14016 is output in the event of a conflict.		

10717	T_NO_FCT_CYCLE_NAME		
MD number	Name for tool-changing cycle with T number		
Default setting: -	Minimum input limit: -	Maximum input limit: -	
Changes effective after Power On	Protection level: 2/7	Unit: -	
Data type: STRING	Applies with effect from SW version: 5.2		
Meaning:	If a T function is programmed in a part program block, the subprogram defined in T_NO_FCT_CYCLE_NAME is called at the end of the block. System variable $C_T / $C_T_PROG can be used in the cycle to scan the programmed T no. as a decimal value, and $C_TS / $C_TS_PROG as a string (only with tool management). If a T number is programmed with the D number, it can be scanned in the cycle in system variable $C_D/$C_D_PROG. System variable $C_T_PROG or $C_D_PROG can be used in the subprogram to check whether the T or D command was programmed. The values can be read out with system variable $C_T or $C_D. If another T command is programmed in the subprogram, no substitution takes place, but the T word is output to the PLC. $MN_T_NO_FCT_CYCLE_NAME and system variables $C_T / $C_TS_PROG are effective both in Siemens mode G290 and in external language mode G291. $MN_M_NO_FCT_CYCLE_NAME and $MN_T_NO_FCT_CYCLE_NAME may not be active in the same block i.e. only one M/T function substitution can be active per block. Neither an M98 call nor a modal subprogram call can be programmed in the block with the T function substitution. A subprogram return jump or end of part program is not allowed. Alarm 14016 is output in the event of a conflict.		

10718	**M_NO_FCT_CYCLE_PAR**		
MD number	M function substitution with parameters		
Default setting: -1	Minimum input limit: -	Maximum input limit: -	
Changes effective after Power ON	Protection level: 2/7	Unit: -	
Data type: DWORD	Applies with effect from SW version: 6.3		
Meaning:	If an M function substitution has been configured with MD 10715: M_NO_FCT_CYCLE[n] / MD 10716: M_NO_FCT_CYCLE_NAME[n], a parameter transfer for each system variable as for the T function substitution can be specified for one of these M functions with MD 10718: M_NO_FCT_CYCLE_PAR. The parameters stored in the system variables always refer to the parts program line in which the M function to be substituted was programmed. The following system variables are available: $C_ME : Address expansion of the substituted M function $C_T_PROG : TRUE if address T was programmed $C_T : Value of address T (integer) $C_TE : Address expansion of address T $C_TS_PROG : TRUE if address TS was programmed $C_TS : Value of address TS (string, with tool management only) $C_D_PROG : TRUE if address D was programmed $C_D : Value of address D $C_DL_PROG : TRUE if address DL was programmed $C_DL : Value of address DL		

10719	**T_NO_FCT_CYCLE_MODE**		
MD number	Parameterization of T function substitution		
Default setting: 0	Minimum input limit: -	Maximum input limit: -	
Changes effective after Power ON	Protection level: 2/7	Unit: -	
Data type: DWORD	Applies with effect from SW version: 6.4		
Meaning:	This machine data is used to set whether D or DL is transferred as a parameter to the T substitution cycle when D or DL and T are programmed in a single block (default) or whether it is to be executed before the T substitution cycle is called. Value 0: as previously, the D or DL number is transferred to the cycle (default value) Value 1: the D or DL number is calculated directly in the block This function is only active if tool change has been configured with the M function (MD 22550: TOOL_CHANGE_MODE = 1), otherwise the D or DL values are always transferred.		

10760	G53_TOOLCORR		
MD number	Mode of action when G53, G153 and SUPA is specified		
Default setting: 2	Minimum input limit: 2	Maximum input limit: 4	
Changes effective after Power On	Protection level: 2/7	Unit: -	
Data type: BYTE	Applies with effect from SW version: 5.2		
Meaning:	The MD is effective in both Siemens mode and in external language mode. This machine data defines whether tool length compensation and tool radius compensation are suppressed with language commands G53, G153 and SUPA. 0 = G53/G153/SUPA is non-modal suppression of zero offsets, tool length compensation and tool radius compensation remain active. 1= G53/G153/SUPA is non-modal suppression of zero offsets, and active tool length and tool radius compensation.		

10800	EXTERN_CHAN_SYNC_M_NO_MIN		
MD number	First M code for channel synchronization		
Default setting: -1	Minimum input limit: 100	Maximum input limit:	
Changes effective after Power On	Protection level: 2/7	Unit: -	
Data type: DWORD	Applies with effect from SW version: 6.2		
Meaning:	Lowest number M code out of an M code number area which is reserved for channel synchronization.		
This MD cannot SINUMERIK 802D sl.			

10802	EXTERN_CHAN_SYNC_M_NO_MAX		
MD number	Last M code for channel synchronization		
Default setting: -1	Minimum input limit: 100	Maximum input limit:	
Changes effective after Power On	Protection level: 2/7	Unit: -	
Data type: DWORD	Applies with effect from SW version: 6.2		
Meaning:	Highest number M code out of an M code number area which is reserved for channel sysnchronization. The number of M codes must not exceed a number of 10 times the number of channels (for example: 2 channels results in max. 20 M codes). Alarm 4170 is issued if an excessive M code area is specified.		
This MD cannot SINUMERIK 802D sl.			

10804	EXTERN_M_NO_SET_INT		
MD number	ASUP activating M code		
Default setting: 96	Minimum input limit: 0	Maximum input limit:	
Changes effective after Power On	Protection level: 2/7	Unit: -	
Data type: DWORD	Applies with effect from SW version: 6.2		
Meaning:	M code to activate interruption type subprogram call in ISO dialect T/M mode (ASUP).		

10806	**EXTERN_M_NO_DISABLE_INT**		
MD number	ASUP deactivating M code		
Default setting: 97	Minimum input limit: 0	Maximum input limit:	
Changes effective after Power On	Protection level: 2/7	Unit: -	
Data type: DWORD	Applies with effect from SW version: 6.2		
Meaning:	M code to activate interruption type subprogram call in ISO dialect T/M mode (ASUP).		

10808	**EXTERN_INTERRUPT_BITS_M96**	
MD number	Interrupt program - Execution (M96)	
Default setting: 0	Minimum input limit: 0	Maximum input limit: 8
Changes effective after Power On	Protection level: 2/7	Unit: -
Data type: WORD	Applies with effect from SW version: 6.2	
Meaning:	Using the data bits described below, the behaviour of the interruption type subprogram activated by M96 P .. can be specified. Bit 0: =0, Disable interruption type subprogram; M96/M97 are treated as standard M codes =1, Enable activation/deactivation of interruption type subprogram using M96/M97 Bit 1: =0, Execution of the part program is continued at the target position of the NC block subsequent to the NC block where the interruption took place =1, Execution of the part program is continued at the interruption position Bit 2: =0, The current NC block is interrupted immediatly and the subprogram is called if the interrupt signal is detected. =1, The subprogram is called after completion of the current NC block Bit 3: =0, When detecting an interrupt signal during execution of a machining cycle, the machining cycle is interrupted. =1, Interrupt after machining cycle completion	

10810	**EXTERN_MEAS_G31_P_SIGNAL**	
MD number	Measuring signal input assignment for G31 P..	
Default setting: 1	Minimum input limit: 0	Maximum input limit: 3
Changes effective after Power On	Protection level: 2/7	Unit: -
Datentype: BYTE	Applies with effect from SW version: 6.2	
Meaning:	Measuring inputs 1 and 2 are assigned to the arguments P .. of G31 P1 to P4 command. It is a bit coded MD. Only bit 0 and bit 1 are evaluated. For example: $MN_EXTERN_MEAS_G31_P_SIGNAL[1], Bit 0=1, the 1st measuring input is activated by G31 P2. $MN_EXTERN_MEAS_G31_P_SIGNAL[3] = 2, the 2nd measuring input is activated by G31 P4. Bit 0: =0: Deactivate measuring input 1 for G31 P1 (-P4) =1 Activate measuring input 1 for G31 P1 (-P4) Bit 1: =0 Deactivate measuring input 2 for G31 P1 (-P4) =1 Activate measuring input 2 for G31 P1 (-P4)	

10814 MD number	EXTERN_M_NO_MAC_CYCLE Macro call via M function		
Default setting:	Minimum input limit:	Maximum input limit:	
Change effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: DWORD	Applies with effect from SW version:		
Meaning:	M number with which a macro is called. The name of the subprgram is stated in $MN_EXTERN_M_NO_MAC_CYCLE_NAME[n]. If the M 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 addresses programmed in the block are written into the appropriate variables. If the M function is programmed again in the subprogram, it will no more be replaced by a subprogram call. $MN_EXTERN_M_NO_MAC_CYCLE_NAME[n] is active only in the external language mode G291. M functions with defined meaning may not be overlaid by a subprogram call. In the case of a conflict, this is reported by alarm 4150: - M0 to M5, - M17, M30, - M19, - M40 to M45, - M function up to switchover of spindle mode/axis mode according to $MC_SPIND_RIGID_TAPPING_M_NR (default: M70), - M function for nibbling/punching acc. to configuration via $MC_NIBBLE_PUNCH_CODE if they have been activated via $MC_PUNCHNIB_ACTIVATION. - with external language applied ($MN_MM_EXTERN_LANGUAGE) additionally M96 to M99 - M functions which are defined by $MN_M_NO_FCT_CYCLE. Exeption: The M function defined with $MC_TOOL_CHANGE_M_CODE for tool change. The subprograms configured with $MN_EXTERN_M_NO_MAC_CYCLE_NAME[n] may not become simultaneously within one block (part program line), i.e. a maximum of one M function replacement, neither an M98 nor a modal subprogram call may be programmed. Return jump to subprogram or end of part program are not allowed either. In the case of a conflict, alarm 14016 is output.		

10815 MD number	EXTERN_M_NO_MAC_CYCLE_NAME UP name for M function macro call		
Default setting:	Minimum input limit:	Maximum input limit:	
Change effective after POWER ON	Protection level:	Unit: -	
Data type: STRING	Applies with effect from SW version:		
Meaning:	Cycle name when calling via the M function defined with $MN_EXTERN_M_NO_MAC_CYCLE[n].		

10818	**EXTERN_INTERRUPT_NUM_ASUP**		
MD number	Interruptnumber for ASUP start (M96)		
Default setting: 2	Minimum input limit: 1	Maximum input limit: 8	
Changes effective after	Protection level:	Unit: -	
Data type: BYTE	Applies with effect from SW version: 6.2		
Meaning:	Number of the interrupt input with which an asynchronous subprogram activated in ISO mode is started (M96<program number>.		

10820	**EXTERN_INTERRUPT_NUM_RETRAC**		
MD number	Interruptnumber for retract (G10.6)		
Default setting: 1	Minimum input limit: 1	Maximum input limit:: 8	
Changes effective after	Protection level:	Protection level: -	
Data type: BYTE	Applies with effect from SW version: 6.2		
Meaning:	Number of the interrupt input with which, in ISO mode, fast retraction to the position programmed with G10.6 is triggered (M96 <Programmnummer>).		

10880	**EXTERN_CNC_SYSTEM**	
MD number	External control system whose programs are executed	
Default setting: 0	Minimum input limit: 0	Maximum input limit: 2
Changes effective after Power On	Protection level: 2/7	Unit: -
Data type: WORD	Applies with effect from SW version: 5	
Meaning:	Selection of the external language 1 = ISO-2: System Fanuc0 Milling (from 5.1) 2 = ISO-3: System Fanuc0 Turning (from 5.2) The functional scope defined in the current Siemens documentation is valid. This data is only evaluated if machine data $MN_MM_EXTERN_LANGUAGE is set.	

10882	**NC_USER_EXTERN_GCODES_TAB [n]:0...59**	
MD number	List of user-specific G codes of an external NC language	
Default setting: -	Minimum input limit: -	Maximum input limit: -
Changes effective after Power On	Protection level: 2/2	Unit: -
Datentype: STRING	Applies with effect from SW version: 5	
Meaning:	Code B is implemented by default for external programming language ISO Dialect0-T. Code A and Code C have different G function names. $MN_NC_USER_EXTERN_GCODES_TAB can be used to rename the G functions. The G command codes can be changed for external NC languages. The G group and the position within the G group remain the same. Only the G command codes can be changed. Up to 30 code changes are possible. Example: $MN_NC_USER_EXTERN_GCODES_TAB[0]="G20" $MN_NC_USER_EXTERN_GCODES_TAB[1]="G70" --> G20 is reassigned to G70; If G70 already exists, an error message appears on NCK reset.	

10884	EXTERN_FLOATINGPOINT_PROG		
MD number	Valuation of programmed values not containing a decimal point		
Default setting: 1	Minimum input limit: 0	Maximum input limit: 1	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: BOOLEAN	Applies with effect from SW version: 5.2		
Meaning:	This machine data is effective for external programming languages, i.e. if MD 18800: MM_EXTERN_LANGUAGE = 1. The machine data defines how programmed values without decimal points are evaluated. 0: Standard Notation: Values without decimal points are interpreted in internal units IS-B, IS-C (see MD EXTERN_INCREMENT_SYSTEM). Values without decimal points are interpreted in internal units e. g. X1000 = 1 mm (with 0.001 mm input resolution) X1000.0 = 1000 mm 1: Pocket Calculator Notation: Values without decimal points are interpreted as mm, inch or degrees. Values without decimal points are interpreted as mm, inch or degrees e.g. X1000 = 1000 mm X1000.0 = 1000 mm		

10886	EXTERN_INCREMENT_SYSTEM		
MD number	Increment system		
Default setting: 0	Minimum input limit: 0	Maximum input limit: 1	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: BOOLEAN	Applies with effect from SW version: 5.2		
Meaning:	This machine data is effective for external programming languages, i.e. if MD 18800: MM_EXTERN_LANGUAGE = 1. This machine data defines which increment system is active 0: Increment system IS-B= 0.001 mm/degree = 0.0001 inch 1: Increment system IS-C = 0.0001 mm/degree = 0.00001 inch		

10888	EXTERN_DIGITS_TOOL_NO		
MD number	Number of digits for T number in external language mode		
Default setting: 2	Minimum input limit: 2	Maximum input limit: 4	
Changes effective after Power On	Protection level: 2/7	Unit: -	
Data type: BYTE	Applies with effect from SW version: 5.2		
Meaning:	The machine data is only effective with $MN_EXTERN_CNC_SYSTEM = 2. Number of digits for tool number in programmed T value. The number of leading digits specified in $MN_EXTERN_DIGITS_TOOL_NO is interpreted as the tool number from the programmed T value. The trailing digits address the compensation memory.		

18800 MD number	MM_EXTERN_LANGUAGE External language active in the control		
Default setting: 0	Minimum input limit: 0	Maximum input limit: 1	
Changes effective after Power On	Protection level: 2/7	Unit: -	
Datentype: DWORD	Applies with effect from SW version: 5		
Meaning:	This MD must be set to enable ISO Dialect0-T and ISO Dialect0-M programs to run on the control. Only one external language can be selected at a time. Please refer to the latest documentation for the available command range. Bit 0 (LSB): Execution of part programs in ISO_2 or ISO_3 mode. For coding see $MN_MM_EXTERN_CNC_SYSTEM (10880)		
This MD cannot SINUMERIK 802D sl.			

D.2 Channel-specific machine data

20050 MD number	AXCONF_GEOAX_ASSIGN_TAB Assignment between geometry axis and channel axis		
Default setting: 70	Minimum input limit: 0	Maximum input limit: 20	
Changes effective after Power ON	Protection level: 2/2	Unit: -	
Data type: BYTE	Applies with effect from SW version: 5.2		
Meaning:	This MD assigns a geometry axis to a channel axis. The assignment must be made for all 3 geometry axes (X,Y,Z). If a geometry axis is not assigned, the value 0 should be entered. The geometry axis is therefore not available and cannot be programmed, e.g. if the second geometry axis is not required for the “turning” technology Y -> entry: value 0 (see default setting for turning).		

20060 MD number	AXCONF_GEOAX_NAME_TAB Geometry axis name in channel		
Default setting: X, Y, Z	Minimum input limit: -	Maximum input limit: -	
Changes effective after Power ON	Protection level: 2/7	Unit: -	
Data type: STRING	Applies with effect from SW version:		
Meaning:	This MD is used to enter the names of the geometry axes for the channel separately. Geometry axes can be programmed in the part program using the names specified here.		

20070 MD number	AXCONF_MACHAX_USED Machine axis number valid in channel		
Default setting: 1, 2, 3, 4	Minimum input limit: 0	Maximum input limit: 31	
Changes effective after Power ON	Protection level: 2/7	Unit: -	
Data type: BYTE	Applies with effect from SW version:		
Meaning:	This MD assigns a machine axis to a channel axis. SINUMERIK 802D has 5 channel axes. Channel axis identifiers for the axes activated in the channel must be specified in MD 20080: AXCONF_CHANAX_NAME_TAB. The axes can be programmed. A machine axis that has not been assigned to a channel axis is not active i.e. no axis control, no display on the screen.		

20080	AXCONF_CHANAX_NAME_TAB		
MD number	Channel axis name		
Default setting: X, Y, Z, A, B, C, U, V, X11, Y11, ...	Minimum input limit: -	Maximum input limit: -	
Changes effective after Power ON	Protection level: 2/7	Unit: -	
Data type: STRING	Applies with effect from SW version:		
Meaning:	In this MD you can set the name of the channel axis. The channel axis is displayed with this identifier in the WCS. This identifier is also written in the program. Generally, the first two or three channel axes are used as geometry axes (see also MD 20050: AXCONF_GEOAX_ASSIGN_TAB). The remaining channel axes are defined as special axes. SINUMERIK 802D has 5 channel axes.		

20094	SPIND_RIGID_TAPPING_M_NR		
MD number	M number for switchover to controlled spindle mode (Siemens mode)		
Default setting: 70	Minimum input limit: 0	Maximum input limit: 0xFF	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: BYTE	Applies with effect from SW version: 5.2		
Meaning:	The machine data is effective in Siemens mode and in external language mode. This machine data defines the M function number used to switch the spindle to controlled spindle mode (axis mode). This number is substituted by M70 in Siemens mode and by M29 in external language mode. Only M numbers which have not already been defined as defaults are allowed. M codes M1, M2, M3, M4, M5, M30, etc. are not allowed, for example.		

20095	EXTERN_RIGID_TAPPING_M_NR		
MD number	M number for switchover to controlled spindle mode (external language mode)		
Default setting: 29	Minimum input limit: 6	Maximum input limit: 0xFF	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: BYTE	Applies with effect from SW version:		
Meaning:	This machine data defines the M function number used to switch the spindle to controlled spindle mode (axis mode) in external language mode. This number can be used in external language mode to substitute M29 with another M function. Only M numbers which have not already been defined as defaults are allowed. M codes M0, M1, M3, M4, M5, M30, M99 etc. are not allowed, for example.		

20150 MD Number	**GCODE_RESET_VALUES** Initial setting of G groups		
Default setting: 2, 0, 0, 1, 0, ...	Minimum input limit: -	Maximum input limit: -	
Changes effective after RESET	Protection level: 2/7	Unit: -	
Data type: BYTE	Applies with effect from SW version:		
Bedeutung:	Definition of G codes which become active on runup and reset or at part program end. The index of the G codes in the respective groups must be programmed as the default value.		

Title Group	Default	
GCODE_RESET_VALUES[0]	1	2 (G01)
GCODE_RESET_VALUES[1]	2	0 (inaktiv)
GCODE_RESET_VALUES[2]	3	0 (inaktiv)
GCODE_RESET_VALUES[3]	4	1 (START FIFO)
GCODE_RESET_VALUES[4]	5	0 (inaktiv)
GCODE_RESET_VALUES[5]	6	1 (G17) bei Fräsen
GCODE_RESET_VALUES[6]	7	1 (G40)
GCODE_RESET_VALUES[7]	8	1 (G500)
GCODE_RESET_VALUES[8]	9	0 (inaktiv)
GCODE_RESET_VALUES[9]	10	1 (G60)
GCODE_RESET_VALUES[10]	11	0 (inaktiv)
GCODE_RESET_VALUES[11]	12	1 (G601)
GCODE_RESET_VALUES[12]	13	2 (G71)
GCODE_RESET_VALUES[13]	14	1 (G90)
GCODE_RESET_VALUES[14]	15	2 (G94)
GCODE_RESET_VALUES[15]	16	1 (CFC)
...		

20152 MD number	**GCODE_RESET_MODE** Reset response in the G groups		
Default setting:	Minimum input limit: 0	Maximum input limit: 1	
Changes effective after Reset	Protection level: 2/7	Unit: -	
Data type: BYTE	Applies with effect from SW version:		
Meaning:	This machine data is only evaluated if bit 0 is set in $MC_RESET_MODE_MASK. For every entry in machine data $MN_GCODE_RESET_VALUES (and thus for every G group) this MD defines whether the setting corresponding to $MC_GCODE_RESET_VALUES will be resumed upon the occurrence of a reset/parts program end (MD = 0), or if the setting valid at that moment will be retained (MD = 1). Example: In this case whenever there is a reset/parts program end, the initial setting for the sixth G group (current plane) will be read from machine data $MC_GCODE_RESET_VALUES: $MC_GCODE_RESET_VALUE(5)=1; Reset value of sixth G group is M17 $MC_GCODE_RESET_MODE(5)=0; Initial setting for sixth G group after a reset/parts program end is as in $MC_GCODE_RESET_VALUES(5) If it is required that the current setting for the sixth G group (current plane) be retained in the event of a reset/parts program end, the setting is as follows: $MC_GCODE_RESET_VALUE(5)=1; Reset value of sixth G group is M17 $MC_GCODE_RESET_MODE(5)=1; Current setting for the sixth G group is retained even after a reset/part program end		
This MD cannot SINUMERIK 802D sl.			

20154 MD number	**EXTERN_GCODE_RESET_VALUES[n]: 0, ..., 30** Defines the G codes which are activated on startup if the NC channel is not running in Siemens mode.		
Default setting: -	Minimum input limit: -	Maximum input -	
Changes effective after Power On	Protection level: 2/2	Unit: -	
Data type: BYTE	Applies with effect from SW version: 5		
Meaning:	The following external programming languages are possible: - ISO dialect milling - ISO dialect turning The G group classification to be used is specified in the current SINUMERIK documentation. The following groups can be defined within MD EXTERN_GCODE_RESET_VALUES: **ISO dialect M**: G code group 2: G17/G18/G19 G code group 3: G90/G91 G code group 5: G94/G95 G code group 6: G20/G21 G code group 13: G96/G97 G code group 14: G54-G59 **ISO dialect T**: G code group 2: G96/G97 G code group 3: G90/G91 G code group 5: G94/G95 G code group 6: G20/G21 G code group 16: G17/G18/G19		
This MD cannot SINUMERIK 802D sl.			

20380	**TOOL_CORR_MODE_G43/G44**		
MD number	Processing of prog. length offsets G43/G44		
Default setting: 0	Minimum input limit: 1	Maximum input limit: 2	
Changes effective after RESET	Protection level: 2/7	Unit: -	
Data type: BYTE	Applies with effect from SW version: 5.2		
Meaning:	The machine data is only effective if MD EXTERN_CNC_LANGUAGE = 1. When G43/G44 is active, it determines how length offsets programmed with H are pro-cessed. 0: mode A The tool length H always acts on the Z axis, independent of the current plane. 1: mode B The tool length H acts on one of the three geometry axes depending on the active plane: G17 on the 3rd geometry axis (usually Z) G18 on the 2nd geometry axis (usually Y) G19 on the 1st geometry axis (usually X) By multiple programming, length offsets can be established in all three geometrical axes in this mode, i.e. by activating an offset, the existing length offset of another axis will not be cancelled. 2: mode C The tool length offset becomes valid in the axis programmed together with the H code regardless of the selected plane. Further, the behaviour is as discribed under mode B.		
This MD cannot SINUMERIK 802D sl.			

20382	**TOOL_CORR_MOVE_MODE**		
MD number	Traversing the tool length offset		
Default setting: FALSE	Minimum input limit: -	Maximum input limit: -	
Changes effective after RESET	Protection level: 2/7	Unit: -	
Data type: BOOLEAN	Applies with effect from SW version: 5.2		
Meaning:	The machine data determines how the tool length offsets are applied. FALSE: A tool length offset is only applied if the associated axis was programmed. (Same behaviour as in previous SW versions) TRUE: Tool length offsets are always applied, regardless of whether the associated axes were programmed.		
This MD cannot SINUMERIK 802D sl.			

20732	**EXTERN_G0_LINEAR_MODE**		
MD number	Rapid traverse interpolation selection		
Default setting: 1	Minimum input limit: 0	Maximum input limit: 1	
Changes effective after POWER ON	Protection level: 2/4	Unit: -	
Data type: BOOLEAN	Applies with effect from SW version:		
Meaning:	This MD determines G00 interpolation behaviour. 0: axes move like positioning axes 1: linear interpolation		

20734	EXTERN_FUNCTION_MASK		
MD number	Function mask for external language		
Default setting:	Minimum input limit: 0	Maximum input limit: 16	
Changes effective after Reset	Protection level: 2/7	Unit: -	
Data type: DWORD	Applies with effect from SW version: 6.2		

Meaning:

This machine data is used to influence functions in ISO mode.

Bit 0 =0: ISO mode T: “A” and “C” are interpreted as axes. If contour definition is programmed, a comma must precede “A” or “C”.

=1: “A” and “C” in the parts program are always interpreted as contour definition.
Neither axis A nor axis C is permitted to exist.

Bit 1 =0: ISO mode T G10 P<100 tool geometry
>100 tool wear

=1: G10 P<10 000 tool geometry
>10 000 tool wear

Bit 2 =0: G04 dwell time: always either [s] or [ms]

=1: if G95 is active, dwell time in spindle revolutions

Bit 3 =0 ISO scanner errors result in an alarm

```
Example:        N5 G291         ; ISO Dialect mode
                N10 WAIT        ; Alarm 12080 “WAIT unknown”
                N15 G91 G500    ; Alarm 12080 “G500 unknown”
```

=1: ISO scanner errors are not output. The block will be transferred to the Siemens translator

```
Example:        N5 G291         ; ISO Dialect mode
                N10 WAIT        ; The Siemens translator will process
                                ; the block
                N15 G91 G500    ; The Siemens translator will process
                                ; the block
                N20 X Y         ; Block processed by ISO translator due to
                                ; G291, G91 off N15 is active
```

Bit 4 =0: G00 is traversed into the exact stop function.
Example: In G64, G00 blocks are also traversed with G64

=1 G00 blocks are always traversed with G09, even when G64 is active

Bit 5 =0: Movements of the rotary axis are carried out along the shortest path

=1: Depending on the sign, movements of the rotary axis are carried out in the positive or negative direction of rotation

Bit 6 =0: Only 4-digit program number allowed

Bit 6 =1: 8-digit program number allowed. Numbers shorter than 4 digits are expanded to 4 digits .

Bit 7 =0: Axis programming with geo axis replacement/parallel axes is compatible in ISO mode

=1: Axis programming with geo axis replacement/parallel axes is compatible with Siemens mode in ISO mode

Bit8 =0: In cycles, the F value is always interpreted as a feedrate for transfer

=1: In thread cycles, the F value is interpreted as a pitch for transfer

Bit9 =0: In ISO Mode T for G84, G88 and in standard mode F for G95, multiplication is by 0.01 mm or 0.0001 inch

=1: In ISO Mode T for G84, G88 and in standard mode F for G95, multiplication is by 0.01 mm or 0.0001 inch

Bit 10 = 0: In M96 Pxx the Pxx program is called when interrupted.

= 1: In M96 Pxx CYCLE396.spf is always called when interrupted.

Bit 11 = 0: When G54 Pxx is programmed, G54.1 is displayed.

= 1: When G54 Pxx or G54.1 Px is programmed, G54Px is always displayed.

Bit 12 = 0: When the UP defined by M96 Pxx is called, $P_ISO_STACK is not changed.

=1: When the UP defined by M96 Pxx is called, $P_ISO_STACK is incremented.

Bit 13 = 0: alle G10 Befehle ohne internem STOPRE

Bit 13 = 1: alle G10 Befehle mit internem STOPRE

22420 MD number	**FGROUP_DEFAULT_AXIS[n]: 0, ..., 7** Default value for FGROUP command		
Default setting: 0	Minimum input limit: 0	Maximum input limit: 8	
Changes effective after Power On	Protection level: 7/7	Unit: -	
Data type: BYTE	Applies with effect from SW version: 5.2		
Meaning:	You can specify up to 8 channel axes whose resulting velocity corresponds to the programmed path feed. If all 8 values are set to zero (default), the geometry axes entered in $MC_AXCONF_GEOAX_ASSIGN_TAB are activated as the default setting for the FGROUP command. Example: The first 4 axes in the channel are relevant for the path feed: $MC_FGROUP_DEFAULT_AXES[0] = 1 $MC_FGROUP_DEFAULT_AXES[2] = 2 $MC_FGROUP_DEFAULT_AXES[3] = 3 $MC_FGROUP_DEFAULT_AXES[4] = 4		
This MD cannot SINUMERIK 802D sl.			

22512 MD number	**EXTERN_GCODE_GROUPS_TO_PLC[n]: 0, ..., 7** Specifies the G groups which are output to the NCK/PLC interface when an external NC language is active		
Default setting: -	Minimum input limit: -	Maximum input limit: -	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: BYTE	Applies with effect from SW version: 5		
Meaning:	The user can select the G groups of an external NC language with channel MD $MC_EX-TERN_ GCODE_GROUPS_TO_PLC . The active G command is then signaled from the NCK to the PLC for these groups. Default 0: No output The NCK/PLC interface is updated on every block change and after a Reset. It cannot always be assured that a block-synchronous relationship exists between the NC block and the signaled G functions (e.g. if very short blocks are used in continuous-path mode). The same applies to $MC_GCODE_GROUPS_TO_PLC		
This MD cannot SINUMERIK 802D sl.			

22515 MD number	**GCODE_GROUPS_TO_PLC_MODE** Behavior of G group to PLC		
Default setting: -	Minimum input limit: -	Maximum input limit: -	
Changes effective after Power ON	Protection level: 2/7	Unit: -	
Data type: DWORD	Applies with effect from SW version: 6.3		
Meaning:	For setting how the G groups are to be interpreted in the PLC as data. The current behavior (bit 0=0) was for the G group to be the array index of a 64 byte field (DBB 208 - DBB 271). That way, up to the 64th G group can be reached. The new behavior (bit 0=1)) is for the data storage in the PLC to be up to 8 bytes (DBB 208 - DBB 215). With this behavior, the array index of this byte array is identical with the index of the MD $MC_GCODE_GROUPS_TO_PLC[Index] and $MC_EXTERN_GCODE_GROUPS_TO_PLC[Index]. Each index (0-7) must only be entered in one of the two machine data, the other must contain the value 0. Bit 0(LSB = 0: Behavior as before, the 64 byte array is used for the G code. Bit 0(LSB = 1: The user sets for which G groups the first 8 bytes will be used		
This MD cannot SINUMERIK 802D sl.			

22900	STROKE_CHECK_INSIDE		
MD number	Determine enternal/external protection zone		
Default setting: 0	Minimum input limit: 0	Maximum input limit: 1	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: BYTE	Applies with effect from SW version: 5.2		
Meaning:	This machine data applies in combination with external programming languages. It is effective with $MN_MM_EXTERN_LANGUAGE = 1. It defines whether protection zone 3 is an internal or external protection zone. Meaning: 0: Protection zone 3 is an internal protection zone 1: Protection zone 3 is an external protection zone		
This MD cannot SINUMERIK 802D sl.			

22910	WEIGHTING_FACTOR_FOR_SCALE		
MD number	Input unit for scaling factor		
Default setting: 0	Minimum input limit: 0	Maximum input limit: 1	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: BOOLEAN	Applies with effect from SW version: 5.2		
Meaning:	This machine data applies in combination with external programming languages. It is active with $MN_MM_EXTERN_LANGUAGE = 1. It defines the unit for the scale factor P and the axial scale factors I, J, K Meaning: 0: Scale factor in 0.001 1: Scale factor in 0.00001		

22914	AXES_SCALE_ENABLE		
MD number	Enable axial scaling (G51)		
Default setting: 0	Minimum input limit: 0	Maximum input limit: 1	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Datentype: BOOLEAN	gültig ab SW-Stand: 5.2		
Meaning:	This MD enables axial scaling. Meaning: 0: Axial scaling not possible 1: Axial scaling possible, (MD DEFAULT_SCALE_FACTOR_AXIS becomes effective)		

22920	**EXTERN_FIXED_FEEDRATE_F1_ON**		
MD number	Activation of fixed feedrates F1 - F9		
Default setting: FALSE	Minimum input limit:	Maximum input limit:	
Changes effective after Power ON	Protection level: 2/7	Unit:	
Data type: BOOLEAN	Applies with effect from SW version: 6.2		
Meaning:	This MD enables the fixed feedrates from the setting data $SC_EXTERN_FIXED_FEEDRATE_F1_F9 [ ]. 0: no fixed feedrates with F1 - F9 1: the feedrates from the setting data $SC_EXTERN_FIXED_FEEDRATE_F1_F9 are activated by programming F1 -F9		

22930	**EXTERN_PARALLEL_GEOAX**		
SD number	Assignment of parallel channel geometry axis		
Default setting: 0	Minimum input limit: 0	Maximum input limit: 3	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: BYTE	Applies with effect from SW version: 6.2		
Meaning:	Assignment of axes parallel to the geometrical axes. Using this table, parallel channel axes can be assigned to geometrical axes. Within the ISO dialect mode, the parallel axes can then be activated as geometrical axes by commanding a G code for plane selection (G17 - G19) together with the axis designation of the relevant parallel axis. Axis interchange is then carried out with the axis defined in $MC_AXCONF_GEOAX_ASSIGN_TAB[ ]. Prerequisite: The channel axes in use must be active. Entering a zero deactivates the relevant parallel geometrical axis.		

24004	**CHBFRAME_POWERON_MASK**		
MD number	Delete channel-specific base frame on Power On		
Default setting: 0	Minimum input limit: 0	Maximum input limit: 0xFF	
Changes effective after POWER ON	Protection level: 2/7	Unit: -	
Data type: DWORD	Applies with effect from SW version: 5.2		
Meaning:	This machine data defines whether channel-specific base frames are deleted on a Power On reset, i.e. work shifts and rotations are reset to 0, scaling is set to 1. Mirroring is switched off. The selection can be made separately for the individual base frames. Bit 0 corresponds to base frame 0, bit 1 to base frame 1, etc. 0: Base frame is retained on Power On 1: Base frame is deleted on Power On.		
This MD cannot SINUMERIK 802D sl.			

24006	CHSFRAME_RESET_MASK		
MD number	Active system frames after reset		
Default setting: 0	Minimum input limit: 0	Maximum input limit: 0x7FF	
Changes effective after RESET	Protection level: 2/7	Unit: -	
Data type: DWORD	Applies with effect from SW version: 5.2		
Meaning:	Bit mask used for the reset setting of the channel-specific system frames included in the channel. Bit 0: System frame for actual value setting and scratching is active after reset. Bit 1: System frame for external work offset is active after reset. Bit 2: Reserved, for TCARR and PAROT see $MC_GCODE_RESET_VALUES[]. Bit 3: Reserved, for TOROT and TOFRAME see $MC_GCODE_RESET_VALUES[]. Bit 4: System frame for workpiece reference points is active after reset. Bit 5: System frame for cycles is active after reset. Bit 6: Reserved; reset behavior dependent on $MC_RESET_MODE_MASK. Bit 7: System frame $P_ISO1FR (ISO G51.1 Mirror) is active after reset. Bit 8: System frame $P_ISO2FR (ISO G68 2DROT) is active after reset. Bit 9: System frame $P_ISO3FR (ISO G68 3DROT) is active after reset. Bit 10: System frame $P_ISO4FR (ISO G51 Scale) is active after reset. Related to: MM_SYSTEM_FRAME_MASK		
This MD cannot SINUMERIK 802D sl..			

28082	MM_SYSTEM_FRAME_MASK		
MD number	System frames (SRAM)		
Default setting: 0	Minimum input limit: 0	Maximum input limit: 0x7FF	
Changes effective after Power ON	Protection level: 2/7	Unit: -	
Data type: DWORD	Applies with effect from SW version: 5.2		
Meaning:	Bit mask for configuring channel-specific system frames included in the channel. The following applies: Bit 0: System frame for setting actual value and scratching Bit 1: System frame for external work offset Bit 2: System frame for TCARR aund PAROT Bit 3: System frame for TOROT and TOFRAME Bit 4: System frame for workpiece reference points Bit 5: System frame for cycles Bit 6: System frame for transformations Bit 7: System frame for $P_ISO1FR for ISO G51.1 Mirror Bit 8: System frame for $P_ISO2FR for ISO G68 2DROT Bit 9: System frame for $P_ISO3FR for ISO G68 3DROT Bit 10: System frame for $P_ISOFR for ISO G51 Scale		
This MD cannot SINUMERIK 802D sl..			

D.3 Axis-specific setting data

43120 MD number	**DEFAULT_SCALE_FACTOR_AXIS** Default axial scale factor for G51 active		
Default setting: 1	Minimum input limit: -99999999	Maximum input limit: 99999999	
Changes effective IMMEDIATELY	Protection level: 7/7	Unit: -	
Data type: DWORD	Applies with effect from SW version: 5.2		
Meaning:	This machine data applies in combination with external programming languages. It is effective with $MN_MM_EXTERN_LANGUAGE = 1. If no axial scale factor I, J or K is programmed in the G51 block, DEFAULT_SCALEFACTOR_AXIS is effective. This MD is valid only if MD AXES_SCALE_ENABLE is set.		

43240 MD number	**M19_SPOS** Position of spindle (degree) when commanding M19		
Default setting: 0	Minimum input limit: -359.999	Maximum input limit: 359.999	
Changes effective IMMEDIATELY	Protection level: 7/7	Unit: -	
Data type: DOUBLE	Applies with effect from SW version: 5.2		
Meaning:	Das Settingdatum ist auch im Siemens-Mode wirksam.		

D.4 Channel-specific setting data

42110	**DEFAULT_FEED**		
SD number	Default value for path feed		
Default setting: 0	Minimum input limit: 0	Maximum input limit: -	
Changes effective IMMEDIATELY	Protection level: 7/7	Unit: -	
Data type: DOUBLE	Applies with effect from SW version: 5.2		
Meaning:	If no path feed is programmed in the part program, the value stored in $SC_DEFAULT_FEED is used. The setting data is evaluated at the start of the part program allowing for the feed type active at the time (see $MC_GCODE_RESET_VALUES and/or $MC_EXTERN_GCODE_RESET_VALUES).		

42140	**DEFAULT_SCALE_FACTOR_P**		
SD number	Default scale factor for address P		
Default setting: 0	Minimum input limit: -99999999	Maximum input limit: 99999999	
Changes effective IMMEDIATELY	Protection level: 7/7	Unit: -	
Data type: DWORD	Applies with effect from SW version: 5.2		
Meaning:	This machine data applies in combination with external programming languages. It is effective with $MN_MM_EXTERN_LANGUAGE = 1. If no scale factor P is programmed in the block, the value in this machine data is applied.		

42150	**DEFAULT_ROT_FACTOR_R**		
SD number	Default angle of rotation R		
Default setting: 0	Minimum input limit: 0	Maximum input limit: 360	
Changes effective IMMEDIATELY	Protection level: 2/7	Unit: degree	
Data type: DOUBLE	Applies with effect from SW version:		
Meaning:	When commanding coordinate rotation G68 without specifying an angle of rotation, the value set to this setting data is applied.		

42160	**EXTERN_FIXED_FEEDRATE_F1_F9**		
SD number	F1 digit feed F1 - F9		
Default setting: 0	Minimum input limit: 0	Maximum input limit:	
Changes effective IMMEDIATELY	Protection level: 2/7	Unit: mm/min	
Data type: DOUBLE	Applies with effect from SW version:		
Meaning:	Pre-defined feedrates which are selected by commanding F1 - F9 when G01 is active.		

42520	**CORNER_SLOWDOWWN_START**		
SD number	Distance before corner		
Default setting: 0	Minimum input limit 0	Maximum input limit: arbitrary	
Change effective immediately	Protection level: 7/7	Unit POSN_LIN	
Data type: DOUBLE	Applies with effect from SW version: 6		
Meaning:	Path length as from which the feed before the corner at G62 is reduced.		

42522	**CORNER_SLOWDOWN_END**		
SD number	Distance after corner		
Default setting: 0	Minimum input limit: 0	Maximum input limit:: arbitrary	
Change effective immediately	Protection level: 7/7	Unit: POSN_LIN	
Data type: DOUBLE	Applies with effect from SW version: 6		
Meaning:	Path length as from which the feed after a corner at G62 remains reduced.		

42524	**CORNER_SLOWDOWN_OVR**		
SD number	Feed override at corner with G62		
Default setting: 0	Minimum input limit: 0	Maximum input limit: arbitrary	
Change effective immediately	Protection level: 7/7	Unit: PERCENT	
Data type: DOUBLE	Applies with effect from SW version: 6		
Meaning:	Override by which the feed at the corner G62 is multiplied.		

42526	**CORNER_SLOWDOWN_CRIT**		
SD number	Criterium for corner detection with G62		
Default setting: 0	Minimum input limit: 0	Maximum input limit: arbitrary	
Change effective immediately	Protection level: 7/7	Unit: POSN_ROT	
Data type: DOUBLE	Applies with effect from SW version: 6		
Meaning:	Angle as from which a corner at feed reduction with G62, G21 is taken into account.		

43340	**EXTERN_REF_POSITION_G30_1**		
MD number	Reference point position for G30.1		
Default setting:	Minimum input limit:	Maximum input limit:	
Changes effective immediately	Protection level:	Unit:	
Data type: DOUBLE	Applies with effect from SW version:		
Meaning:	Setting data Reference point position for G30.1. This setting data is evaluated in CYCLE328.		

Data Fields, Lists

E

E.1 Machine data

Number	Identifier	Name	Refer-ence
General ($MN_ ...)			
10604	WALIM_GEOAX_CHANGE_MODE	Work area limitation when switching geometri-cal axes	
10615	NCFRAME_POWERON_MASK	Delete global base frames on Power On	K2
10652	CONTOUR_DEF_ANGLE_NAME	Definable name for angle in the contour short description	
10654	RADIUS_NAME	Definable name for radius non-modally in the contour short description	
10656	CHAMFER_NAME	Definable name for chamfer in the contour short description	
10704	DRYRUN_MASK	Activating dry run feed	
10706	SLASH_MASK	Activating the block skip function	
10715	M_NO_FCT_CYCLE[n]: 0, ..., 0	M function number for tool change cycle call	
10716	M_NO_FCT_CYCLE_NAME[ ]	Name of tool-changing cycle for M functions-from MD $MN_MFCT_CYCLE	
10717	T_NO_FCT_CYCLE_NAME	Name for tool-changing cycle with T no.	
10718	M_NO_FCT_CYCLE_PAR	M function substitution with parameters	
10719	T_NO_FCT_CYCLE_MODE	Parameterization of T function substitution	
10760	G53_TOOLCORR	Behaviour of G53, G153 and SUPA	
10800	EXTERN_CHAN_SYNC_M_NO_MIN	First M code for channel synchronization	
10802	EXTERN_CHAN_SYNC_M_NO_MAX	Last M code for channel synchronization	
10804	EXTERN_M_NO_SET_INT	ASUP activation M code	
10806	EXTERN_M_NO_DISABLE_INT	ASUP deactivation M code	
10808	EXTERN_INTERRUPT_BITS_M96	Interrupt program execution (M96)	
10810	EXTERN_MEAS_G31_P_SIGNAL	Measuring input assignment for G31 P..	
10814	EXTERN_M_NO_MAC_CYCLE	Macro call via M function	
10815	EXTERN_M_NO_MAC_CYCLE_NAME	UP name for M function macro call	
10818	EXTERN_INTERRUPT_NUM_ASUP	Interruptnumber for ASUP start (M96)	
10820	EXTERN_INTERRUPT_NUM_RETRAC	Interruptnumber for retract (G10.6)	
10880	EXTERN_CNC_SYSTEM	External control system whose programs are to be executed	

General ($MN_ ...)			
10881	EXTERN_GCODE_SYSTEM	ISO mode T: G code system selection	
10882	NC_USER_EXTERN_GCODES_TAB[n]: 0-59	List of user defined G commands for external NC language	
10884	EXTERN_FLOATINGPOINT_PROG	Evaluation for progr. without decimal point	
10886	EXTERN_INCREMENT_SYSTEM	Defines the increment system	
10888	EXTERN_DIGITS_TOOL_NO	Number of digits for T number in external language mode	
10890	EXTERN_TOOLPROG_MODE	Tool change programming with external programming language	
18190	MM_NUM_PROTECT_AREA_NCK	Number of files for machine related protection zones (SRAM)	S7
18800	MM_EXTERN_LANGUAGE	External language active in the control	
Channel-specific ($MC_ ...)			
20050	AXCONF_GEOAX_ASSIGN_TAB[ ]	Assignment geometrical axis / channel axis	K2
20060	AXCONF_GEOAX_NAME_TAB[ ]	Geometrical axis in channel	K2
20070	AXCONF_MACHAX_USED[ ]	Valid machine axis number in channel	K2
20080	AXCONF_CHANAX_NAME_TAB[ ]	Channel axis designation in channel	K2
20094	SPIND_RIGID_TAPPING_M_NR	M function number with which the spindle switches to controlled spindle mode	
20095	EXTERN_RIGID_TAPPING_M_NR	M function number in external language mode for spindle switchover to controlled spindle mode	
20100	DIAMETER_AX_DEF	Geometrical axis with cross axis functionality	P1
20150	GCODE_RESET_VALUES[n]: 0 bis max. Anzahl G-Codes	G code groups reset values	K1
20152	GCODE_RESET_MODE	Reset response in the G group	
20154	EXTERN_GCODE_RESET_VALUES[n]: 0-30	G code groups reset values	
20380	TOOL_CORR_MODE_G43G44	Behaviour of tool length compensation G43/G44	
20382	TOOL_CORR_MOVE_MODE	Traversing tool length compensation	
20732	EXTERN_G0_LINEAR_MODE	Determine traverse movement of G00	
20734	EXTERN_FUNCTION_MASK	External language function mask	
22420	FGROUP_DEFAULT_AXES[ ]	FGROUP command default value	
22512	EXTERN_GCODE_GROUPS_TO_PLC[n]: 0-7	Specifies the G groups which are output to the NCK/PLC interface when an external NC language is active	
22900	STROKE_CHECK_INSIDE	Protection zone direction (inside/outside)	
22910	WEIGHTING_FACTOR_FOR_SCALE	Unit of scale factor	
22914	AXES_SCALE_ENABLE	Enable axial scaling (G51)	
22920	EXTERN_FEEDRATE_F1_F9_ACTIV	Enable F 1-digit feed (F0 - F9)	
22930	EXTERN_PARALLEL_GEOAX	Assign parallel channel geometry axis	
24004	CHBFRAME_POWERON_MASK	Delete channel-specific base frame on Power On	
24006	CHSFRAME_RESET_MASK	Active system frames after reset	

Channel-specific ($MC_ ...)			
28080	NUM_USER_FRAMES	Number of zero offsets	
28082	MM_SYSTEM_FRAME_MASK	System frames (SRAM)	
29210	NUM_PROTECT_AREA_ACTIVE	Activate protection zone	
34100	REFP_SET_POS[0]	Reference position / not used when absolute measuring system is applied	
35000	SPIND_ASSIGN_TO_MACHAX	assign spindle / machine axis	

E.2 Setting data

Number	Identifier	Name	Refer-ence
Axis-specific			
43120	$SC_DEFAULT_SCALE_FACTOR_AXIS	Default axial scale factor when G51 active	
43240	$SA_M19_SPOS	Position of spindle when programming M19	
42890	$SA_M19_SPOSMODE	Positioning mode of spindle when commanding M19	
Channel-specific			
42110	$SC_DEFAULT_FEED	Default value for path feed	V1
42140	$SC_DEFAULT_SCALE_FACTOR_P	Default scale factor for address P	
42150	$SC_DEFAULT_ROT_FACTOR_R	Default angle of rotation R	
42520	$SC_CORNER_SLOWDOWN_START	Distance before corner	
42522	$SC_CORNER_SLOWDOWN_END	Distance after corner	
42524	$SC_CORNER_SLOWDOWN_OVR	Feed override at corner with G62	
42526	$SC_CORNER_SLOWDOWN_CRIT	Criterium for corner detection with G62	
43340	$SC_EXTERN_REF_POSITION_G30_1	Reference point position for G30.1	

E.3 Variables

Identifier	Type	Description
$C_A	REAL	Value of programmed address A in ISO Dialect mode for cycle programming
$C_B	REAL	Value of programmed address B in ISO Dialect mode for cycle programming
....	....	.....
$C_G	INT	G number for cycle calls in external mode
$C_H	REAL	Value of programmed address H in ISO Dialect mode for cycle programming
$C_I[ ]	REAL	Value of programmed address I in ISO Dialect mode for cycle programming and macro programming with G65/G66. Up to 10 items are possible in one block for macro programming. The values are stored in the array in the order in which they are programmed.
$C_I_ORDER[ ]	REAL	For description see $C_I[ ], used to define the programming sequence
$C_J[ ]	REAL	For description see $C_I[ ]
$C_J_ORDER[ ]	REAL	For description see $C_I[ ], used to define the programming sequence
$C_K[ ]	REAL	For description see $C_I[ ]
$C_K_ORDER[ ]	REAL	For description see $C_I[ ], used to define the programming sequence
$C_L	INT	Value of programmed address L in ISO Dialect mode for cycle programming
....	....	....
$C_Z	INT	Value of programmed address Z in ISO Dialect mode for cycle programming
$C_TS	STRING	String of tool name programmed at address T
$C_A_PROG	INT	Address A is programmed in a block with a cycle call. 0 = not programmed 1 = programmed (absolute) 3 = programmed (incremental)
$C_B_PROG	INT	Address B is programmed in a block with a cycle call. 0 = not programmed 1 = programmed (absolute) 3 = programmed (incremental)
....	....	....
$C_G_PROG	INT	The shell cycle call is programmed with a G function
$C_Z_PROG	INT	Address Z is programmed in a block with a cycle call. 0 = not programmed 1 = programmed (absolute) 3 = programmed (incremental)
$C_TS_PROG	INT	A tool name was programmed at address T TRUE = programmed, FALSE = not programmed
$C_ALL_PROG	INT	Bitmap of all programmed addresses in a block with a cycle call Bit 0 = address A Bit 25 = address Z Bit = 1 address programmed in incremental dimensions Bit = 0 address not programmed
$P_EXTGG[n]	INT	Active G code of the external language
$C_INC_PROG	INT	Bitmap of all programmed incremental addresses in a block with a cycle call Bit 0 = address A Bit 25 = address Z Bit = 1 address programmed in incremental dimensions Bit = 0 address programmed in absolute dimensions

Identifier	Type	Description
$C_I_NUM	INT	Cycle programming: Value is always 1 if bit 0 set in $C_I_PROG. Macro programming: Number of I addresses programmed in 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 map of all programmed addresses in a block with a cycle call Bit 0 = A Bit 25 = Z Bit = 0 axis programmed as INT Bit = 1 axis programmed as REAL
$C_PI	INT	Program number of the interrupt routine that was programmed with M96

Notes

Alarms

F

If error states are detected in cycles, an alarm is generated and cycle execution is interrupted.
The cycles continue to output messages in the dialog line of the control. These messages do not interrupt execution.
Alarms with numbers between 61000 and 62999 are generated in the cycles This number range is subdivided further according to alarm reactions and cancelation criteria.

Table F-1 Alarm number and alarm description

Alarm no.	Brief description	Source	Explanation/remedy
General alarms			
61001	Pitch of thread not correct	CYCLE376T	Pitch of thread is not specified correctly
61003	No feed programmed in cycle	CYCLE371T, CYCLE374T, CYCLE383T, CYCLE384T, CYCLE385T, CYCLE381M, CYCLE383M, CYCLE384M, CYCLE387M	No feed F word was programmed in the calling block before the cycle call, see standard Siemens cycles
61004	Configuration of geometry axis not correct	CYCLE328	The order of the geometry axes is incorrect, see standard Siemens cycles
61101	Reference plane improperly defined	CYCLE375T, CYCLE81, CYCLE83, CYCLE84, CYCLE87	See standard Siemens cycles
61102	No spindle direction programmed	CYCLE371T, CYCLE374T, CYCLE383T, CYCLE384T, CYCLE385T, CYCLE381M, CYCLE383M, CYCLE384M, CYCLE387M	Spindle direction M03 or M04 missing, see standard Siemens cycles
61107	First drilling depth incorrectly defined		First drilling depth counter to total drilling depth
61603	Grooving incorrectly defined	CYCLE374T	Grooving depth value 0
61607	Start point incorrect	CYCLE376T	The start point is not outside of the area to be machined
61610	No in-feed programmed	CYCLE374T	In-feed value = 0
ISO alarms			
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_EX-TERN_ LANGUAGE or option bit 19800 $ON_EXTERN_LAN-GUAGE is not set

Table F-1 Alarm number and alarm description, continued

Alarm no.	Brief description	Source	Explanation/remedy
61801	Incorrect G code selected	CYCLE300, CYCLE371T, CYCLE374T, CYCLE376T, CYCLE383T, CYCLE384T, CYCLE385T	An illegal numerical value for the CNC system was programmed in the program call CYCLE300<value> or in the cycle setting data an incorrect value for the G code system was specified.
61802	Incorrect axis type	CYCLE328, CYCLE330	The programmed axis is assigned to a spindle
61803	Programmed axis does not exist	CYCLE328, CYCLE330	The programmed axis does not exist in the system. Check MD20050-20080
61804	Programmed position beyond reference point	CYCLE328, CYCLE330	The programmed intermediate position or current position is located behind the reference point
61805	Value programmed in absolute and incremental dimensions	CYCLE328, CYCLE330, CYCLE371T, CYCLE374T, CYCLE376T, CYCLE383T, CYCLE384T, CYCLE385T	The intermediate position is programmed using both absolute and incremental dimensions
61806	Incorrect axis assignment	CYCLE328	The order of the axis assignment is incorrect
61807	Incorrect spindle direction programmed	CYCLE384M	The programmed spindle direction conflicts with the spindle direction used for the cycle
61808	Final drilling depth or single drilling depth missing	CYCLE383T, CYCLE384T, CYCLE385T, CYCLE381M, CYCLE383M, CYCLE384M, CYCLE387M	Total depth Z or single drilling depth Q missing from G8x block (first call of cycle)
61809	Drilling position not admissible	CYCLE383T, CYCLE384T, CYCLE385T	
61810	ISO G cde not possible	CYCLE383T, CYCLE384T, CYCLE385T	
61811	ISO axis designation not admissible	CYCLE328, CYCLE330. CYCLE371T, CYCLE374T, CYCLE376T, CLE383T, CYCLE384T, CYCLE385T	The calling NC block contains an ISO axis designation which is not admissible
61812	Incorrect numeral value(s) in cycle call	CYCLE371T, CYCLE376T,	The calling NC block contains a numeral value which is not admissible
61813	Incorrect GUD value	CYCLE376T	Not admissable numeral value in cycle setting data
61814	Polar coordinatea not possible	CYCLE381M, CYCLE383M, CYCLE384M, CYCLE387M	
61815	G40 not active	CYCLE374T, CYCLE376T	G40 was not active prior to the cycle call

Commands

C

G

M

S

Index

To
SIEMENS AG
A&D MC MS1
P.O. Box 3180

D-91050 Erlangen, Germany

Fax: +49-(0)9131 / 98 - 63315 [documentation]
mailto:documotioncontrol.docu@siemens.com
http://www.siemens.com/automation/service&support

Suggestions
Corrections

for Publication/Manual:

SINUMERIK 802D sl/840D sl /840D/840Di sl/840Di/810D Programming Manual ISO Milling

User Documentation

Programming Guide

Edition: 04.2007

From

Name

Address of your company/department

Street

Telephone: /

Telefax: /

Should you come across any printing errors when reading this publication, please notify us on this sheet. Suggestions for improvement are also welcome.

Suggestions and/or corrections