

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

### Programming Manual ISO Turning

#### Programming Guide

#### Validity

<i>Software</i>	<i>Version</i>
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

Programming Basics 1

Commands Calling  
Axis Movements 2

Movement Control  
Commands 3

Enhanced Level  
Commands 4

#### Appendix

Abbreviations A

Terms B

G Code Table C

MDs and SDs D

Data Fields, Lists E

Alarms F

Index

# 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.

<b>Edition</b>	<b>Order No.</b>	<b>Remarks</b>
02.01	6FC5 298-6AC10-0BP0	<b>A</b>
12.01	6FC5 298-6AC10-0BP1	<b>C</b>
11.02	6FC5 298-6AC10-0BP2	<b>C</b>
04.05	6FC5 298-7AC10-0BP0	<b>C</b>
04.07	6FC5398-5BP10-0BA0	<b>C</b>

## 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.

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

It may be possible to run functions 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	<a href="http://www.siemens.com/automation/support-request">http:// www.siemens.com/automation/support-request</a>		
E-Mail	<a href="mailto:adsupport@siemens.com">mailto:adsupport@siemens.com</a>		

---

### 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	<a href="mailto:motioncontrol.docu@siemens.com">mailto:motioncontrol.docu@siemens.com</a>

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 referring to 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

<b>1</b>	<b>Programming Basics</b>	<b>1-11</b>
1.1	Introductory explanations	1-11
1.1.1	Siemens mode	1-11
1.1.2	ISO Dialect mode	1-11
1.1.3	Switchover	1-12
1.1.4	G code display	1-12
1.1.5	Maximum number of axes / axis designation	1-12
1.1.6	Selection of G code system A, B, or C	1-13
1.1.7	Decimal point programming	1-13
1.1.8	Comments	1-15
1.1.9	Block skip	1-16
1.2	Basics of feed function	1-17
1.2.1	Rapid traverse	1-17
1.2.2	Cutting feed (F command)	1-17
1.2.3	Switching between feed per minute mode and feed per revolution mode (G94/G95)	1-21
<b>2</b>	<b>Commands Calling Axis Movements</b>	<b>2-23</b>
2.1	Interpolation commands	2-23
2.1.1	Positioning (G00)	2-23
2.1.2	Linear interpolation (G01)	2-25
2.1.3	Circular interpolation (G02, G03)	2-27
2.1.4	Cylindrical interpolation (G07.1)	2-33
2.1.5	Polar coordinate interpolation (G12.1, G13.1)	2-35
2.2	Using the thread cutting function	2-38
2.2.1	Thread cutting and continuous thread cutting (G33)	2-38
2.2.2	Continuous thread cutting	2-41
2.2.3	Multiple-thread cutting (G33)	2-43
2.2.4	Variable lead thread cutting (G34)	2-46
2.3	Reference point return	2-47
2.3.1	Automatic return to reference point (G28)	2-47
2.3.2	Reference point return check (G27)	2-49
2.3.3	Second to fourth reference point return (G30)	2-50
2.3.4	Rapid lift with G10.6	2-51
2.4	Tool retract (G10.6)	2-53
<b>3</b>	<b>Movement Control Commands</b>	<b>3-55</b>
3.1	The coordinate system	3-55
3.1.1	Machine coordinate system (G53)	3-56
3.1.2	Workpiece coordinate system (G92)	3-57
3.1.3	Resetting the work (G92.1)	3-57
3.1.4	How to select a workpiece coordinate system	3-58
3.1.5	Instantaneous mapping of the ISO functions onto the Siemens frames (until powerline 7.04.2, solution line 1.4)	3-59
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)	3-61

3.2	Determining the coordinate value input modes .....	3-66
3.2.1	Absolute/incremental designation .....	3-66
3.2.2	Diametric and radial commands for X-axis .....	3-69
3.2.3	Inch/metric input designation (G20, G21) .....	3-70
3.2.4	G60: Oriented positioning .....	3-71
3.3	Time-controlling commands .....	3-72
3.3.1	Dwell (G04) .....	3-72
3.4	Tool offset functions .....	3-73
3.4.1	Tool offset data memory .....	3-73
3.4.2	Tool position offset .....	3-73
3.4.3	Tool nose radius compensation function (G40, G41/G42) .....	3-73
3.5	Spindle function (S function) .....	3-80
3.5.1	Spindle command (S5-digit command) .....	3-80
3.5.2	Constant surface speed control (G96, G97) .....	3-81
3.6	Tool function (T function) .....	3-84
3.7	Miscellaneous function (M function) .....	3-84
3.7.1	M codes relating to stop operation (M00, M01, M02, M30) .....	3-84
3.7.2	Internally processed M codes .....	3-85
3.7.3	Macro call via M function .....	3-85
3.7.4	General purpose M codes .....	3-86
<b>4</b>	<b>Enhanced Level Commands .....</b>	<b>4-89</b>
4.1	Program support functions (1) .....	4-89
4.1.1	Canned cycles .....	4-89
4.1.2	Multiple repetitive cycles .....	4-101
4.1.3	Hole-machining canned cycles (G80 to G89) .....	4-118
4.2	Program support functions (2) .....	4-129
4.2.1	Changing of tool offset value Programmable data input (G10) .....	4-129
4.2.2	Subprogram call up function (M98, M99) .....	4-130
4.3	Eight-digit program number .....	4-134
4.4	Automating support functions .....	4-135
4.4.1	Skip function (G31) .....	4-135
4.4.2	Multistage skip (G31, P1-P2) .....	4-138
4.5	Macroprograms .....	4-139
4.5.1	Differences from subprograms .....	4-139
4.5.2	Macroprogram call (G65, G66, G67) .....	4-139
4.6	Advanced functions .....	4-147
4.6.1	High-speed cycle cutting (G05) .....	4-147
4.6.2	Polygonal turning .....	4-148
4.6.3	Compressor in ISO dialect mode .....	4-150
4.6.4	Switchover modes for DryRun and skip levels .....	4-151
4.6.5	Interrupt programm with M96 / M97 (ASUP) .....	4-152



<b>A</b>	<b>Abbreviations</b> .....	<b>A-155</b>
<b>B</b>	<b>Terms</b> .....	<b>B-165</b>
<b>C</b>	<b>G Code Table</b> .....	<b>C-195</b>
	C.1 G code table .....	C-195
<b>D</b>	<b>Machine and Setting Data</b> .....	<b>D-199</b>
	D.1 Machine/Setting Data .....	D-199
	D.2 Channel-specific machine data .....	D-212
	D.3 Axis-specific setting data .....	D-219
	D.4 Channel-specific setting data .....	D-220
<b>E</b>	<b>Data Fields, Lists</b> .....	<b>E-223</b>
	E.1 Machine data .....	E-223
	E.2 Setting data .....	E-226
	E.3 Variables .....	E-226
<b>F</b>	<b>Alarms</b> .....	<b>F-229</b>
<b>G</b>	<b>Commands</b> .....	<b>I-231</b>
	<b>Index</b> .....	<b>I-233</b>

[illegible]

# Programming Basics

# 1

## 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

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

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

### 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-T the maximum number of axis is 8. Axis designation for the first two axes is fixed to X and Z. Further axes can be designated Y, A, B, C, U, V, W.

### 1.1.6 Selection of G code system A, B, or C

ISO Dialect T distinguishes between G code system A, B, and C. G code system B is default setting. The G code system in use is selected by MD \$MN\_MM\_EXTERN\_GCODE\_SYSTEM as follows:

\$MN\_MM\_EXTERN\_GCODE\_SYSTEM = 0: G code system B  
 \$MN\_MM\_EXTERN\_GCODE\_SYSTEM = 1: G code system A  
 \$MN\_MM\_EXTERN\_GCODE\_SYSTEM = 2: G code system C

#### G Code system A

If G code system A is active, G91 is not available. In this case, incremental axes movement for axis X,Y, and Z is programmed by address U, V, and W. U, V, and W are not available as axis designation in this case resulting in a maximum axes number of 6.

Address H is used for programming incremental movement of the C axis in G code system A.

---

#### Notice

- If not otherwise noted, the manual in hand describes G code system B.
  - For the differences between G code system A, B, and C refer to the G code list in the appendix.
- 

### 1.1.7 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, see Chapter 4 "Startup".

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

## 1.1 Introductory explanations

## 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
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

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

Address	Unit	IS-B	IS-C
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 Bit8 = 1 G95 F	mm inch mm inch	0.01 0.0001 0.0001 0.00000 1	0.01 0.0001 0.0001 0.00000 1

### 1.1.8 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.

### 1.1.9 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.

---

#### Notice

- “1” can be omitted for “/1”.
  - 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.

### 1.2.2 Cutting feed (F command)

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 characters F. The axis feed mode to be used is selected by designating the feed function G code (G94 or G95) as indicated in Table 1-3. Select the required feed mode by designating the feed function G code before specifying an F code.

Table 1-3 Cutting feed mode G codes

G code	Function	Group
G94	Designation of feed per minute (mm/min) mode	05
G95	Designation of feed per revolution (mm/rev) mode	05

See 1.2.3 “Switching between feed per minute mode and feed per revolution mode” for details of these G codes. The F code is modal and once designated it remains valid until another F code is designated. If feed mode designation G codes are switched between G94 and G95, however, it is necessary to designate the F code again. If no new F code is designated, alarm 10860 “No feedrate programmed” occurs.

### Feed per revolution mode (G95)

A feedrate of a cutting tool per revolution of the spindle (mm/rev, inch/rev) can be designated by a numeral specified following address character F.

**Note:** The upper limit of feedrates could be restricted by the servo system and the mechanical system. For the actual programmable feedrate range, refer to the manuals published by the machine tool builder.

## 1.2 Basics of feed function

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.

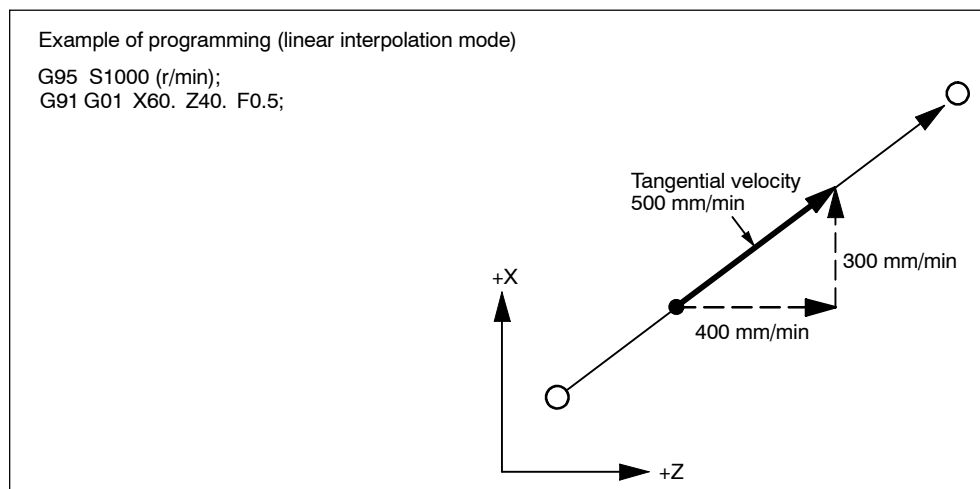


Fig. 1-1 F command in simultaneous 2-axis control linear interpolation (feed per revolution)

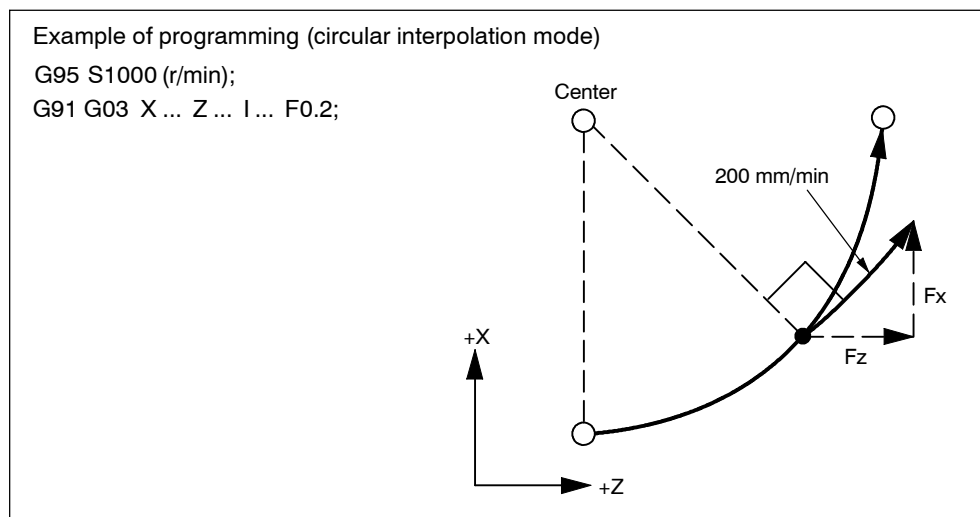


Fig. 1-2 F command in the simultaneous 2-axis control circular interpolation (feed per revolution)

#### Notice

- An F0 command causes an input error.
- A feedrate in the X-axis direction is determined by the radial value.

A feedrate of a cutting tool per minute (mm/min, inch/min) can be designated by a numeral specified following address character F.

**Notice**

The upper limit of feedrates could be restricted by the servo system and the mechanical system. For the actual programmable feedrate range, refer to the manuals published by the machine tool builder.

**Simultaneous 2-axis control**

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.

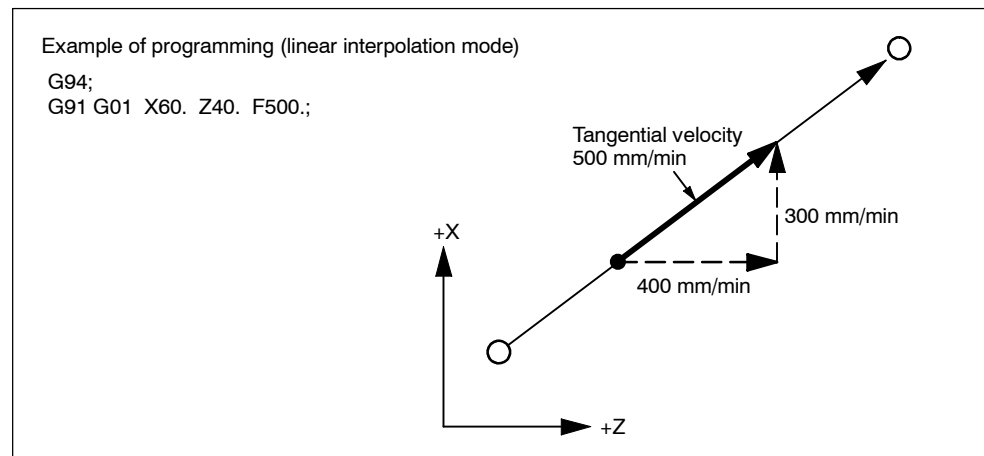


Fig. 1-3 F command in simultaneous 2-axis control linear interpolation (feed per minute)

**Notice**

- An F0 command causes an input error.
- A feedrate in the X-axis direction is determined by the radial value.

## 1.2 Basics of feed function

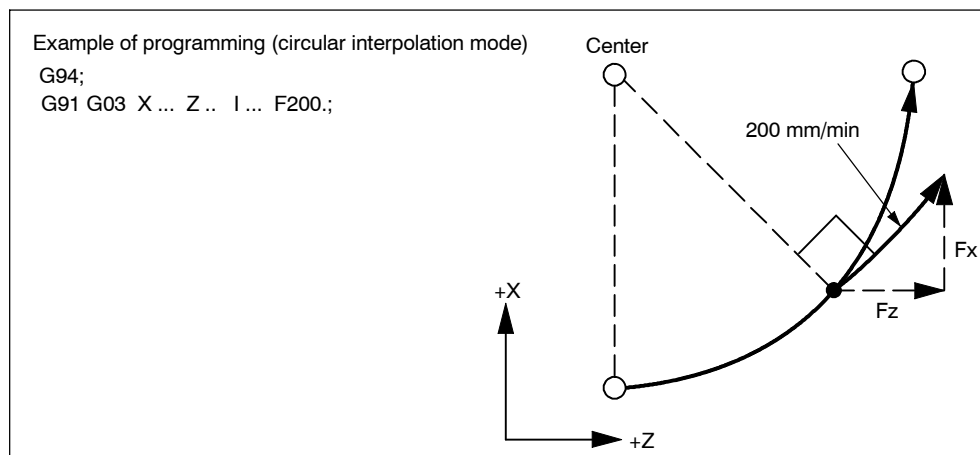


Fig. 1-4 F command in the simultaneous 2-axis control circular interpolation (feed per minute)

### Notice

Do not specify a negative value for an F command.

### Rotary axis and linear axis

An F command specified in the interpolation mode between a rotary axis and a linear axis represents the feedrate in the tangential direction.

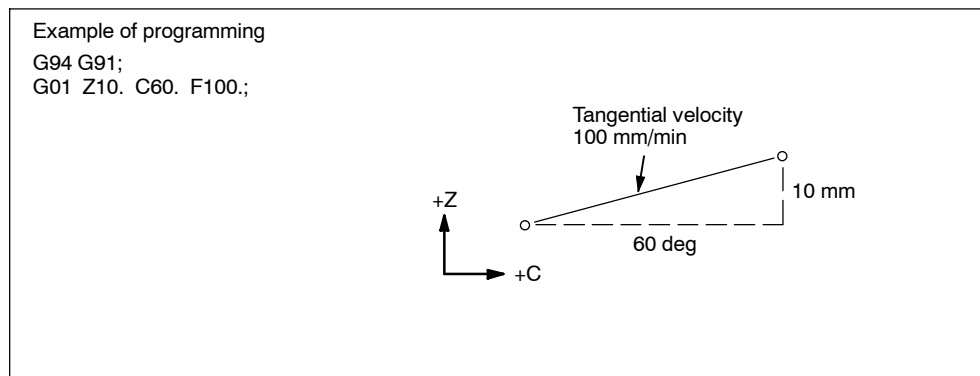


Fig. 1-5 F command in interpolation between rotary axis and linear axis (feed per minute)

### 1.2.3 Switching between feed per minute mode and feed per revolution mode (G94/G95)

Before specifying a feedrate command (F), a G code that determines whether the specified feedrate command is interpreted as feed per minute value or feed per revolution value should be specified. These G codes (G94, G95) are modal and once they are specified they remain valid until the other G code is specified. When the feed mode designation G code is specified, the presently valid F code is canceled. Therefore, an F code must be specified newly after switching the feed mode by designating G94 or G95 command. The initial status that is established when the power is turned on is set by MD 20154, EXTERN\_GCODE\_RESET\_VALUES[4].

Table 1-4 MD EXTERN\_GCODE\_RESET\_VALUES[4] and initial status

MD 20154	Initial G code
MD EXTERN_GCODE_RESET_VALUES[4]=1	G94
MD EXTERN_GCODE_RESET_VALUES[4]=2	G95

#### Feed per minute mode (G94)

By specifying "G94;", the F codes specified thereafter are all executed in the feed per minute mode.

Table 1-5 Meaning of G94 command

G94	Meaning
mm input	mm/rev
inch input	inch/rev

#### Feed per revolution mode (G95)

By specifying "G95;", the F codes specified thereafter are all executed in the feed per revolution mode.

Table 1-6 Meaning of G95 command

G95	Meaning
mm input	mm/rev
inch input	inch/rev

[illegible]

# Commands Calling Axis Movements

## 2.1 Interpolation commands

### 2.1.1 Positioning (G00)

The G00 command moves a tool to the position in the workpiece system specified with an absolute or an incremental command at a rapid traverse rate. In the absolute command, coordinate value of the end point is programmed. In the incremental command the distance the tool moves is programmed.

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

Table 2-1 G code for positioning

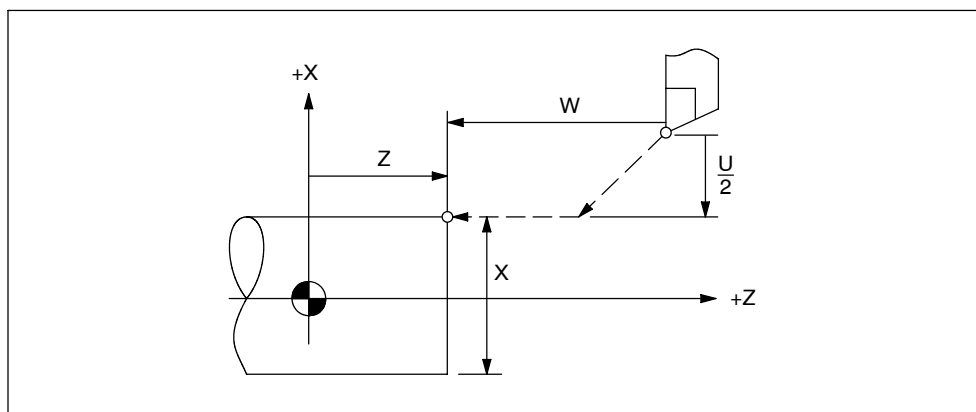
G code	Function	Group
G00	Positioning	01

#### Format

G00 X... Z... ;

When “G00 X(U)... Z(W)... (C(H)... Y(V)...);” 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 2-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.



## 2.1 Interpolation commands

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

**Notice**

- 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.
- The block where a T command is specified must contain the G00 command. Designation of the G00 command is necessary to determine the speed for offset movement which is called by the T command.

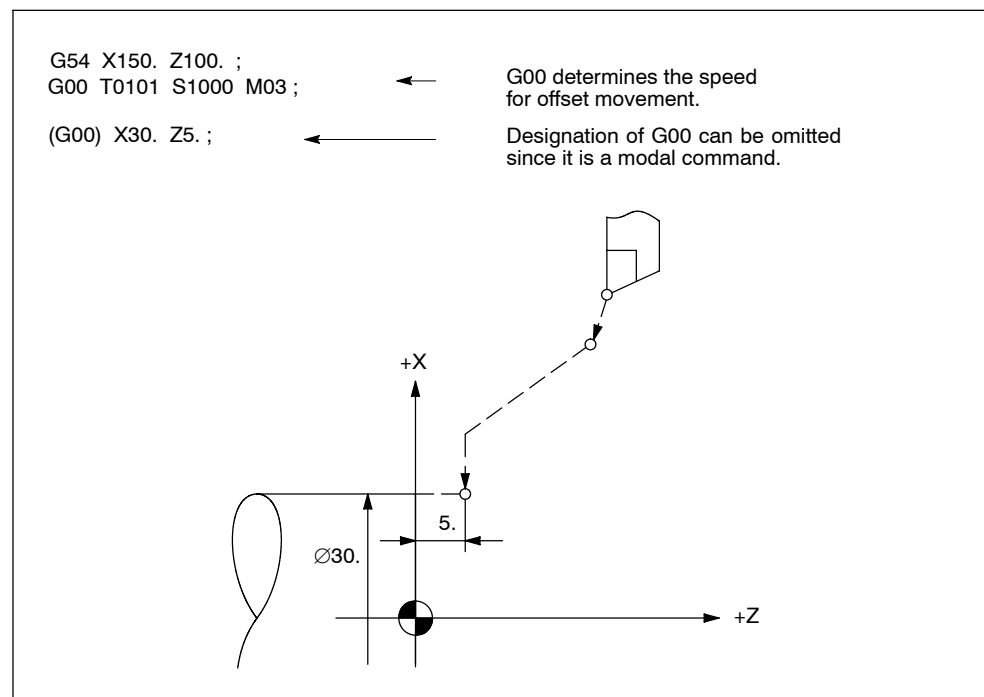


Fig. 2-2 Example of programming

**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... Z... F... ;

With the commands of "G01 X(U)... Z(W)... (C(H)... Y(V)... ) F...;", linear interpolation is executed in the simultaneous 2-axis control mode. The axes not designated in the G01 block do not move. For the execution of the linear interpolation, the following commands must be specified.

### Command format

To execute the linear interpolation, the commands indicated below 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.

(Fx: feedrate in the X-axis direction)

- With an F code, axis feedrate is specified in either feed per spindle revolution (mm/rev or inch/rev) or feed per minute (mm/min or inch/min).

### Notice

For the C-axis, a feedrate cannot be specified in the feed per minute mode.

- End point

The end point can be specified in either incremental or absolute values corresponding to the designation of an address character or G90/G91. For details, see 3.2.1, "Absolute/incremental designation".

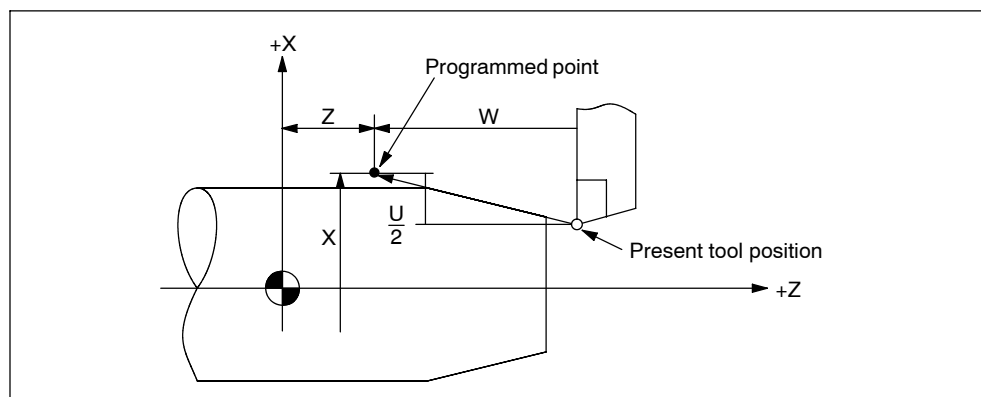


Fig. 2-3 Linear interpolation

## 2.1 Interpolation commands

$$F \text{ (mm/min)} = \sqrt{F_x^2 + F_z^2 + (F_c^2)}$$

Example of programming

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

} Axes are moved in the G01 linear interpolation mode.

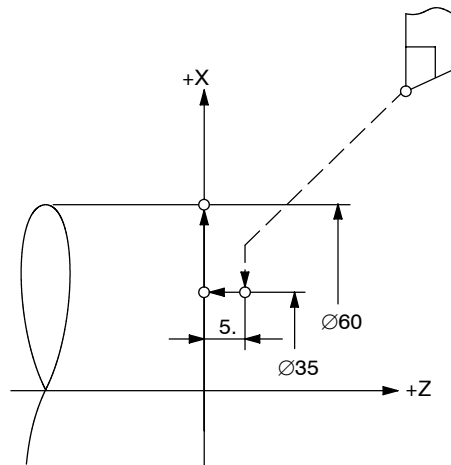


Fig. 2-4 Example of programming

### 2.1.3 Circular interpolation (G02, G03)

#### Format

By specifying the following commands in a program, the cutting tool moves along the specified arc in the ZX plane so that tangential velocity is equal to the feedrate specified by the F code.

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

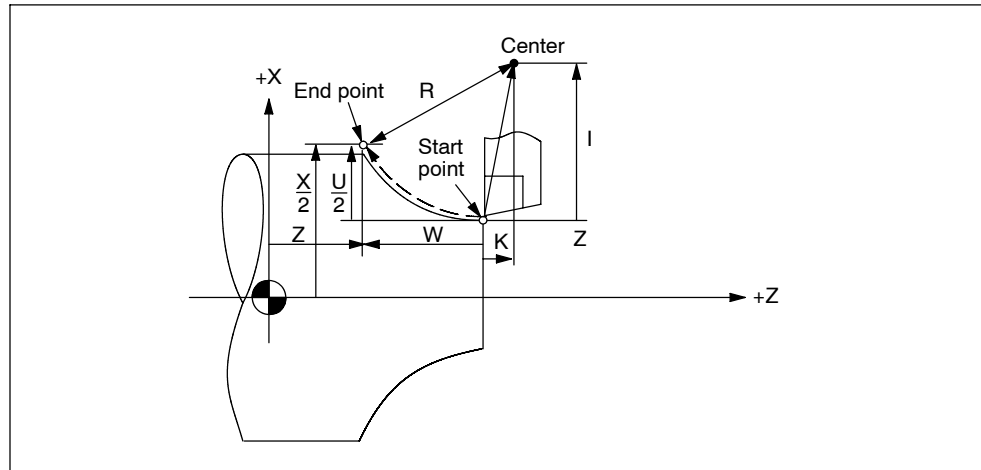


Fig. 2-5 Circular interpolation

#### Command format

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

## 2.1 Interpolation commands

Table 2-2 Commands for circular interpolation

Item	Address	Description
Direction of rotation	G02	Clockwise (CW)
	G03	Counterclockwise (CCW)
End point position	X (U)	X coordinate of arc end point (diametric value)
	Z (W)	Z coordinate of arc end point
	Y (V)	Y coordinate of arc end point
Distance from the start point to the center	I	Distance along the X-axis from the start point to the center of arc (radial value)
	K	Distance along the Z-axis from the start point to the center of arc
	J	Distance along the Y-axis from the start point to the center of arc
Radius of circular arc	R	Distance to the center of arc from the start point

## Rotation direction

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

Table 2-3 Rotation direction

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

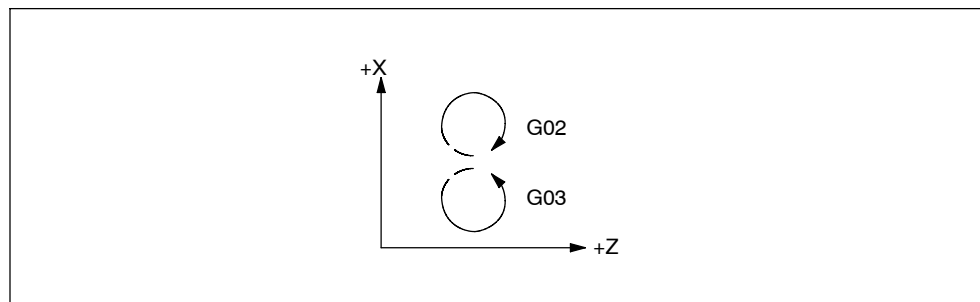


Fig. 2-6 Rotation direction of circular arc

## End point

The end point can be specified in either incremental or absolute values corresponding to the designation of G90 or G91.

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.

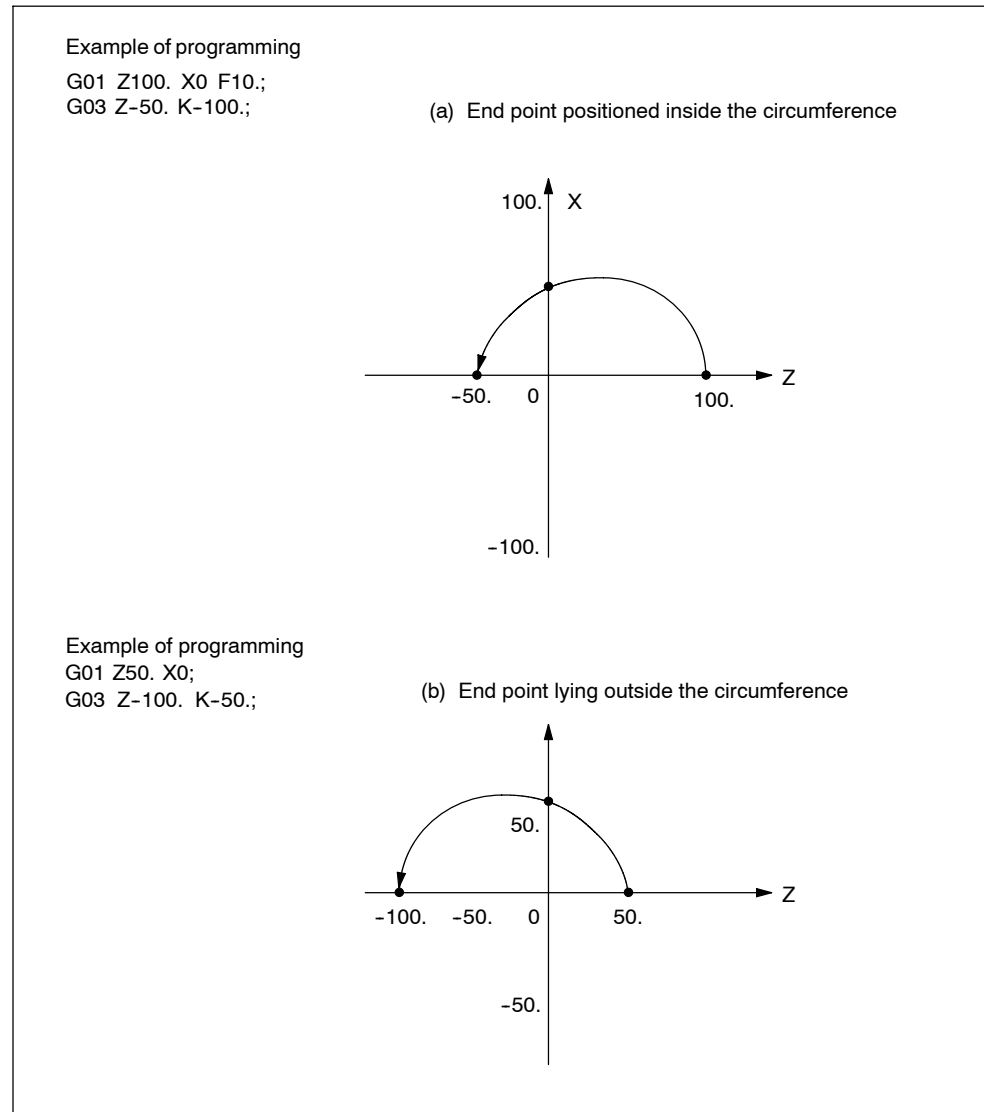


Fig. 2-7 Interpolation with end point off the specified arc

## 2.1 Interpolation commands

## 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.

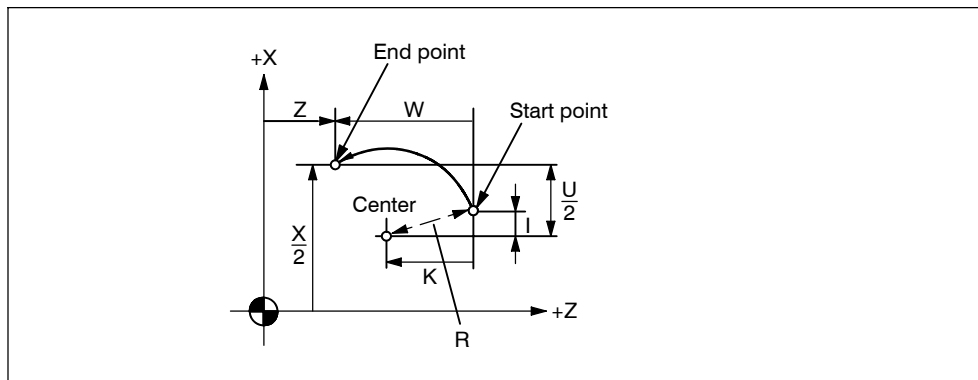


Fig. 2-8

- 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 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”.

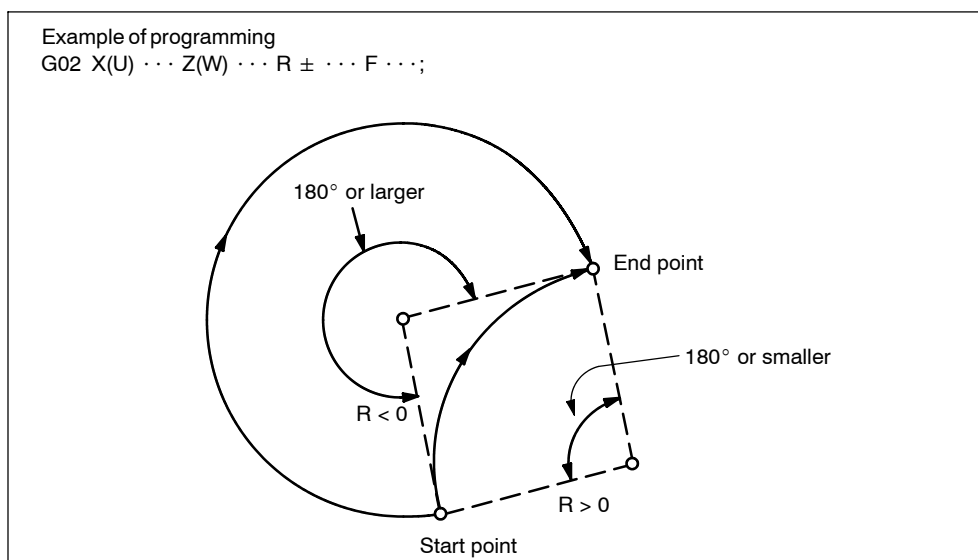


Fig. 2-9 Circular interpolation with radius R designation

## Supplements to circular interpolation

A circular arc extending to multiple quadrants can be defined by the commands in a single block.

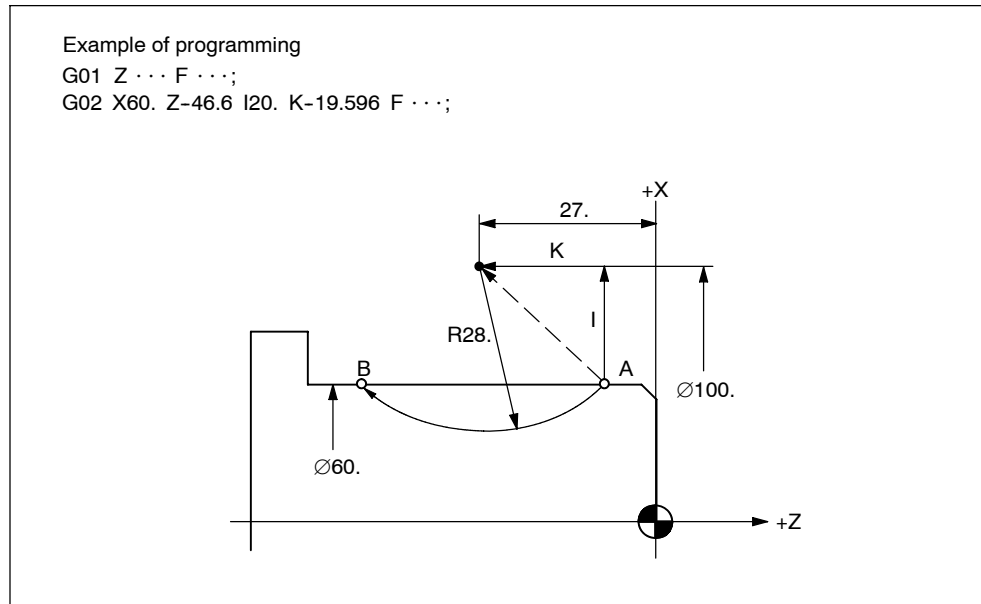


Fig. 2-10 Circular interpolation over multiple quadrants

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

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...; Chamfering
- , 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.

## 2.1 Interpolation commands

**Example**

```

N10 G1 X10. Z100. F1000 G18
N20 A140 C7.5
N30 X80. Z70. A95.824, R10

```

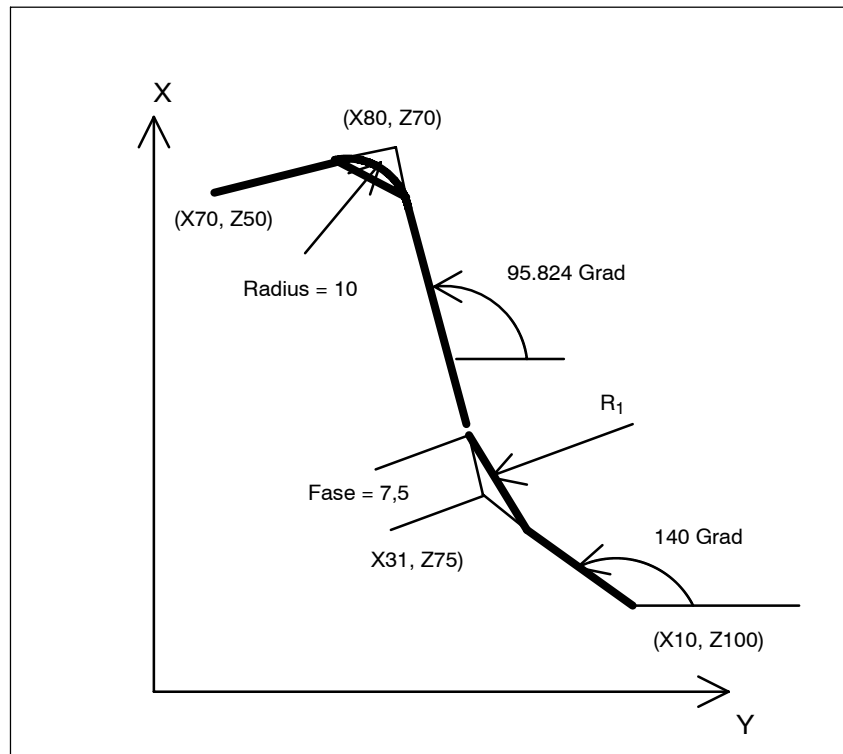


Fig. 2-11 3 straight lines

**Restrictions****ISO dialect mode**

Address C is used in ISO Dialect 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.



### 2.1.4 Cylindrical interpolation (G07.1)

The cylindrical interpolation function allows programming of machining on a cylindrical workpiece (grooving on a cylindrical workpiece) in the manner like writing a program in a plane using the cylinder developed coordinate system. This functions allows programming both in absolute commands (C, Z) and incremental commands (H, W).

The following G code is used for cylindrical interpolation.

Table 2-4 G codes used for cylindrical interpolation

G code	Function	Group
G07.1	Cylindrical interpolation mode	18

#### Format

G07.1 C... r ;

Starts the cylindrical interpolation mode (enables cylindrical interpolation).

G07.1 C0 ;

The cylindrical interpolation mode is cancelled.

C: The rotation axis

r: The radius of the cylinder

Specify G07.1 C... r ; and G07.1 C0 ; in separate blocks.

#### Notice

- G07.1 is based on the Siemens option TRANSMIT. The relevant machine data need to be set accordingly.
- For details refer to the manual "Extended Functions", chapter M1, 2.1 ff.

Specify G07.1 in a block without other commands. G07.1 is a modal G code of group 18. Once G07.1 is specified, the cylindrical interpolation mode ON state remains until G07.1 C0 is commanded. When the power is turned ON or the NC is reset, the cylindrical interpolation mode OFF state is set.

## 2.1 Interpolation commands

## Example

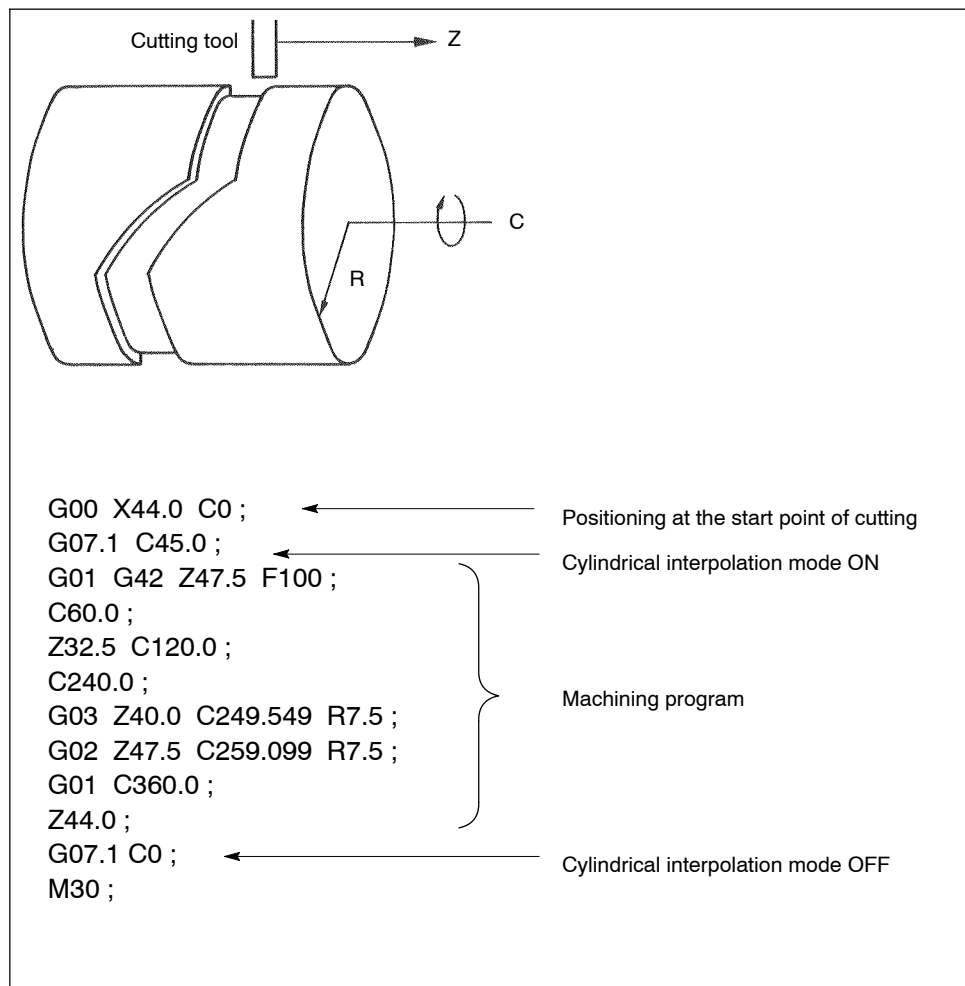


Fig. 2-12 Coordinate system for cylindrical interpolation

In the cylindrical interpolation mode, program restart is not possible. If program restart is attempted from a block in the cylindrical interpolation mode, an alarm occurs. However, program restart is allowed for blocks in which the cylindrical interpolation mode blocks are included.

### 2.1.5 Polar coordinate interpolation (G12.1, G13.1)

The polar coordinate interpolation function allows programming of machining that is executed by the combination of tool movement and workpiece rotation in a virtual rectangular coordinate system.

In the machining accomplished by the combination of a linear axis and a rotary axis, the rotary axis is assumed to be a linear axis that is perpendicular to the linear axis. By assuming a rotary axis as a linear axis, machining an arbitrary shape that is defined by the linear and rotary axis can be programmed easily in the rectangular coordinate system. In this programming, both of absolute commands and incremental commands can be used.

#### Programming format

When G12.1 is specified, the polar coordinate interpolation mode is established and the virtual coordinate system is set in the plane represented by a linear- and a rotary axis with the origin of the absolute coordinate system taken as the origin of this coordinate system. Polar coordinate interpolation is executed in this plane. Note that polar coordinate interpolation starts when G12.1 is specified assuming the present position of the rotary axis to be "0".

---

#### Notice

Return the rotary axis to the origin of the absolute coordinate system before specifying G12.1.

---

#### Features of G12.1 and G13.1

The following G codes are used to turn ON/OFF the polar coordinate interpolation mode.

Table 2-5 G codes used for turning ON/OFF the polar coordinate interpolation

G code	Function	Group
G12.1	Polar coordinate interpolation mode ON	21
G13.1	Polar coordinate interpolation mode OFF	21

Specify G12.1 and G13.1 in a block without other commands.

G12.1 and G13.1 are modal G codes of group 21. Once G12.1 is specified, the polar coordinate interpolation mode ON state remains until G13.1 is specified. When the power is turned ON or the NC is reset, the G13.1 (polar coordinate interpolation mode OFF) state is set.

## 2.1 Interpolation commands

---

**Notice**

- The Polar Coordinate Interpolation 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 ff.
- 

**Restrictions when selecting**

- An intermediate motion block is not inserted (phases/radii).
- A spline block sequence must be terminated.
- Tool radius compensation must be deselected.
- The frame which was active prior to TRACYL is deselected by the control (corresponds to “Reset programmed frame” G500).
- An active working area limitation is deselected by the control for the axes affected by the transformation (corresponds to programmed WALIMOF).
- Continuous path control and rounding are interrupted.
- DRF offsets must have been deleted by the operator.
- In the case of cylinder generated surface curve transformation with groove wall compensation (axis configuration 2, TRAFO\_TYPE\_n = 513), the axis used for the correction (TRAFO\_AXES\_IN\_n[3]) must be set to zero ( $y = 0$ ) so that the groove is machined in the center of the programmed groove center line.

**Restrictions when delecting**

- The same points apply as for selection.

**Restrictions when in polar coordinate interpolation**

- Tool change:  
Tools may only be changed when the tool radius compensation function is deselected.
- Work offset:  
All instructions which refer exclusively to the base coordinate system are permissible (work offset, tool radius compensation). Unlike the procedure for inactive transformation, however, a work offset change with G91 (incremental dimension) is not specially treated. The increment to be traversed is evaluated in the workpiece coordinate system of the new work offset – regardless of which work offset was effective in the previous block.
- Rotary axis:  
The rotary axis cannot be programmed because it is occupied by a geometry axis and cannot thus be programmed directly as a channel axis.

### Example of programming

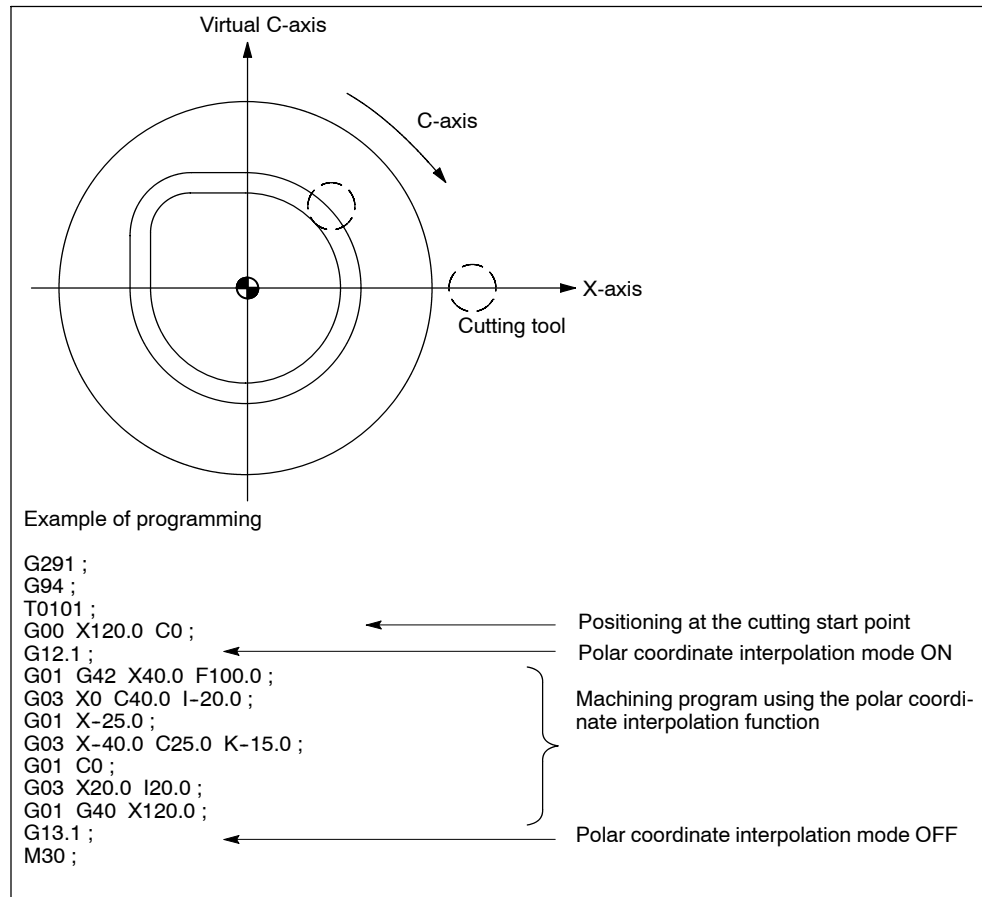


Fig. 2-13 Coordinate system for polar coordinate interpolation

### Notice

- Cylindrical interpolation mode must be deselected before the tool radius compensation and length compensation are deselected.

2.2 Using the thread cutting function

2.2.1 Thread cutting and continuous thread cutting (G33)

Format

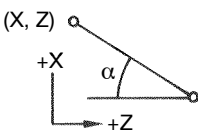
G code system A	G code system B	G code system C
G32	G33	G33

With the commands of “G... X (U)... Z (W)... F... ;”, it is possible to cut straight thread, tapered thread, or scroll thread in the lead specified by an F command to the point specified by absolute coordinate values (X, Z) or incremental coordinate values (U, W).

Direction of thread lead

The direction of thread lead specified by the F commands is indicated in Table 2-6.

Table 2-6 Direction of thread lead

		Direction of thread lead
	$\alpha \leq 45^\circ$	Lead in the Z-axis direction should be specified.
	$\alpha > 45^\circ$	Lead in the X-axis direction should be specified.

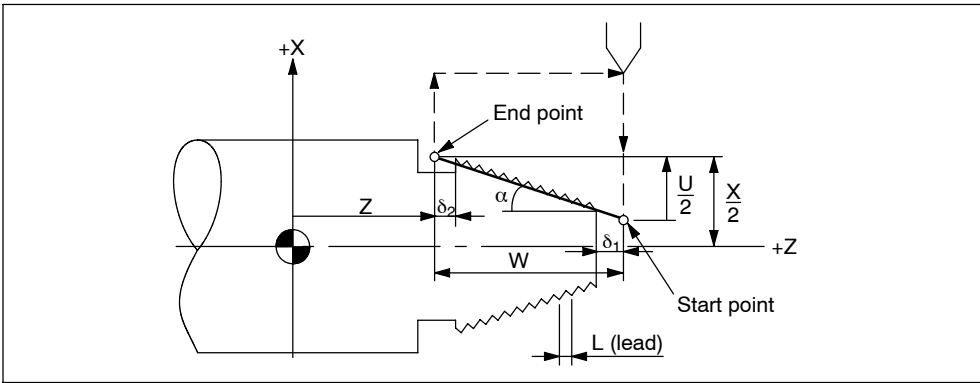


Fig. 2-14 Thread cutting

## Programming formats

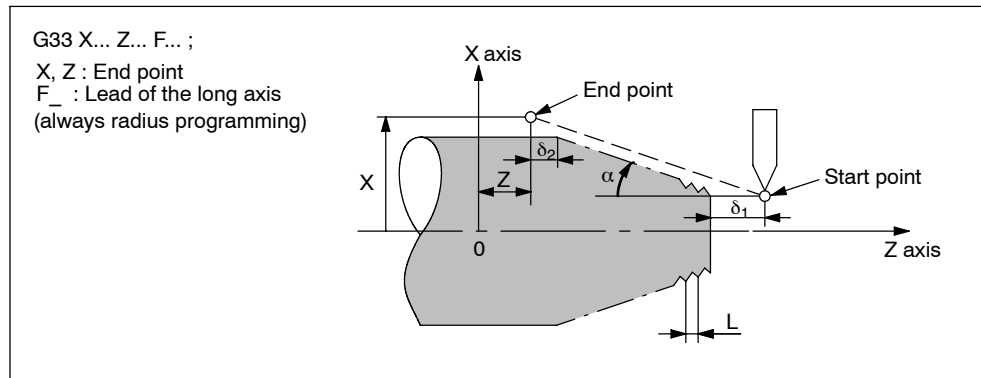


Fig. 2-15

## Example of programming for cutting straight thread (G code system A)

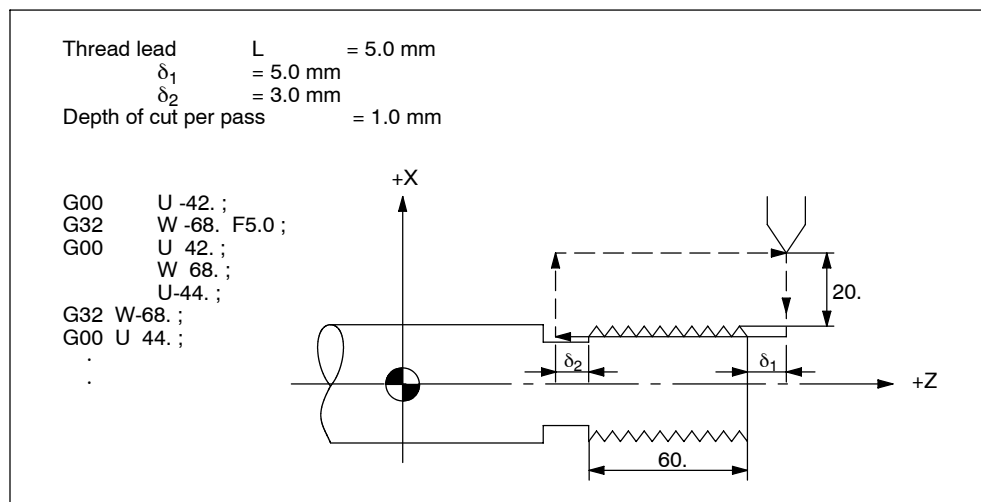


Fig. 2-16 Example of programming for cutting straight thread

## 2.2 Using the thread cutting function

**Example of programming for cutting tapered thread (G code system A)**

Thread lead      L      = 4.0 mm  
                           $\delta_1$       = 3.0 mm  
                           $\delta_2$       = 2.0 mm  
 Depth of cut per pass      = 1.0 mm

```
G00 X13.
G32 X38. W-35. F4.0 ;
G00 X60. ;
      W35. ;
      X11. ;
G32 X36. W-35. ;
G00 X60. ;
      .
      .
```

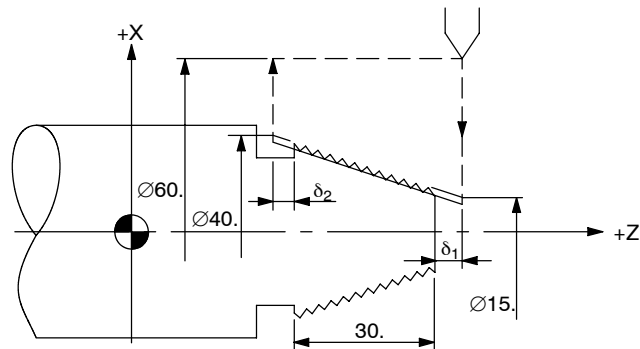


Fig. 2-17 Example of programming for cutting tapered thread



### 2.2.2 Continuous thread cutting

Since the NC has buffer register, designation for continuous thread cutting is possible. In addition, continuous threads can be cut smoothly because the block-to-block pause time is "0" for thread cutting command blocks.

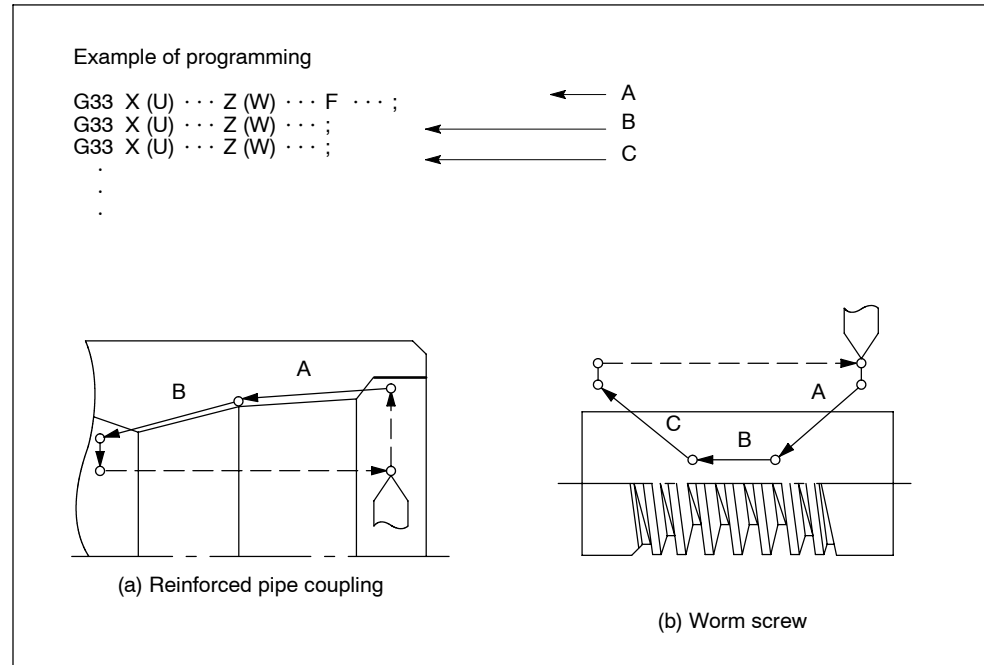


Fig. 2-18 Continuous thread cutting

#### Notice

If designation of thread lead (F) is changed during thread cutting cycle, lead accuracy is lost at joints of blocks. Therefore, thread lead designation must not be changed during thread cutting cycle.

If continuous thread cutting is specified, M codes must not be specified. If an M code is specified, the cycle is suspended at the specified block and continuous thread cannot be cut.

## 2.2 Using the thread cutting function

### Margin for incomplete thread portions ( $\delta_1$ , $\delta_2$ )

At the start and end of thread cutting, lead error is generated. Therefore, margins  $\delta_1$  and  $\delta_2$  should be given at the start and end portions in thread cutting.

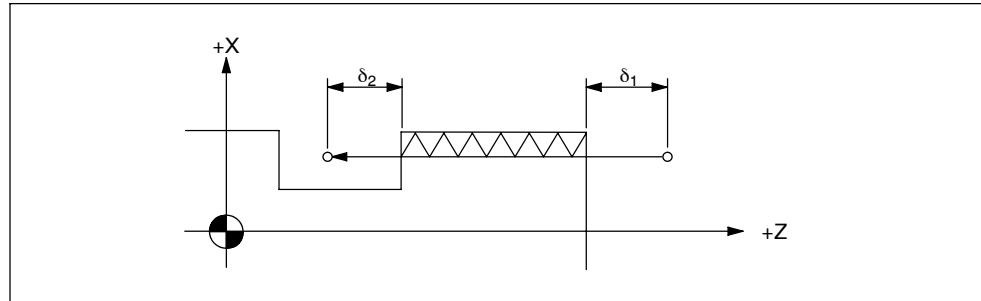


Fig. 2-19 Margins for incomplete threads

#### Notice

Keep the spindle speed at the same value until one thread is cut. If the spindle speed is not maintained constant, accuracy could be lost due to servo lag.

#### Notice

During thread cutting, override operation and feed hold operation are disregarded. If G33 is specified in the G94 (feed per minute) mode, an alarm occurs.

### 2.2.3 Multiple-thread cutting (G33)

G code system A	G code system B	G code system C
G32	G33	G33

Multiple-thread cutting (multiple threads in a lead) is possible without shifting the thread cutting start point. In thread cutting operation, axis feed starts in synchronization with the start-point pulse (1 pulse/turn) output from the spindle pulse generator attached to the spindle. Therefore, the thread cutting start point is always at the same point on the workpiece circumference. In multiple-thread cutting operation, axis feed starts when the spindle rotates by a certain angle after the output of the start-point pulse from the spindle pulse generator.

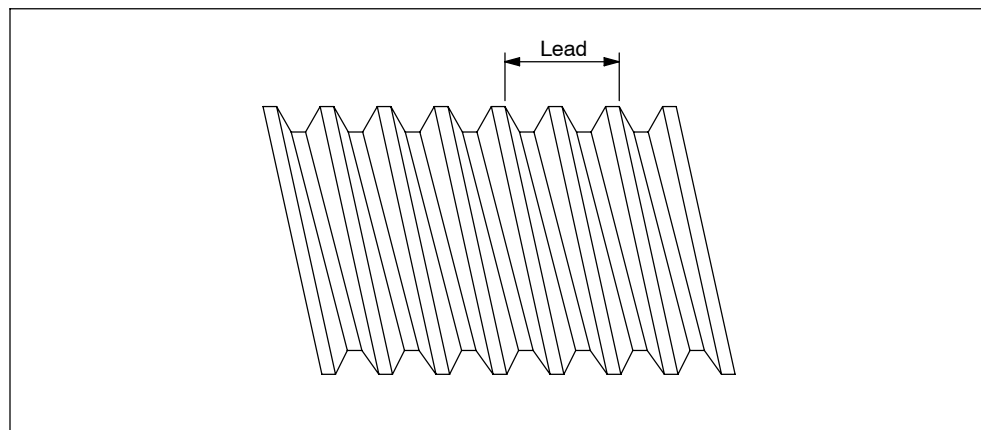


Fig. 2-20 Double-start thread

#### Format

With the commands of “G... X (U)... Z (W)... F... Q... ;”, the spindle rotates by the angle specified by address Q after the output of the start-point pulse of the spindle pulse generator. After that thread cutting starts toward the point specified by X (U) and Z (W) at the lead specified by an F command.

2.2 Using the thread cutting function

Table 2-7      Address Q specified in multi-thread cutting

Least input increment	:	0.001°
Programmable range	:	$0 \leq B < 360.000$

Number of threads and Q command

In general, the thread cutting start points lie on the workpiece circumference; the intervals of these points are calculated by dividing 360° by the number of threads. Examples of multiple threads (double-start, triple-start, and quadra-start threads) are shown in Fig. 2-21.

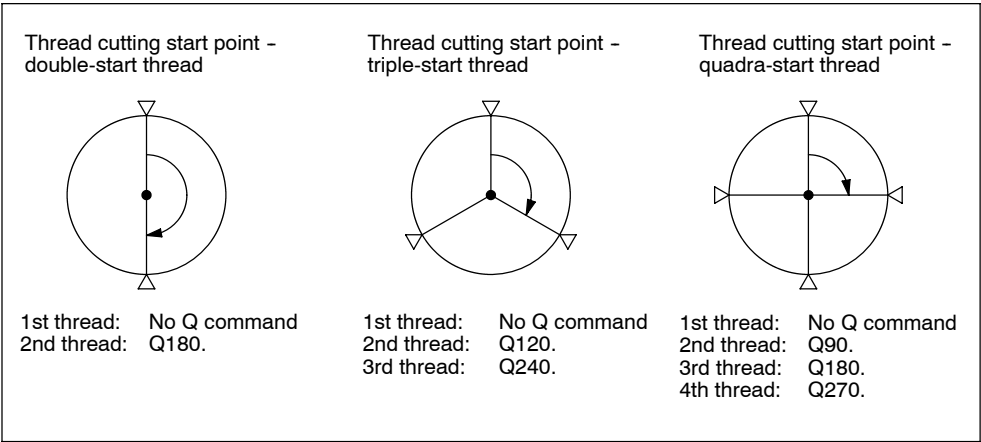


Fig. 2-21      Number of threads and Q commands

### Spindle rotating angle from start-point pulse specified by Q command (G code system A)

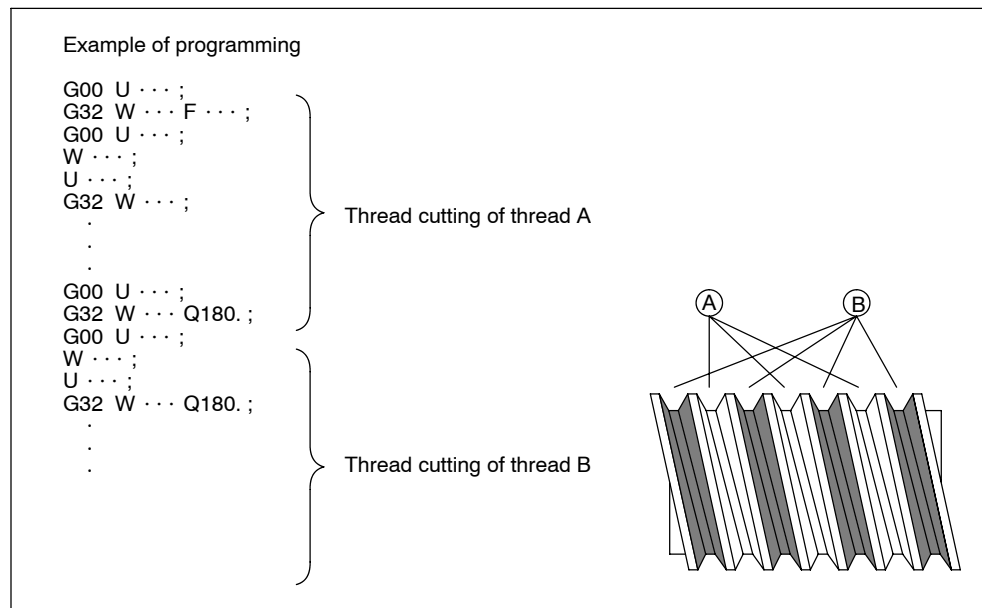


Fig. 2-22 Spindle rotation angle from start-point pulse by Q command

#### Notice

If a Q command is specified for multiple-thread cutting, continuous thread cutting is not possible.

G33W Q90

G33W ← ... Since the operation is suspended at this block to wait for the start-point pulse, continuous thread cannot be cut.

The spindle rotation angle from the start-point pulse is specified using a Q command (0 to 360°) disregarding of the spindle rotating direction.

## 2.2 Using the thread cutting function

## 2.2.4 Variable lead thread cutting (G34)

## Format

G code system A	G code system B	G code system C
G34	G34	G34

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

With the commands of “G34 X (U)... Z (W)... F... K... ;”, variable lead thread can be cut; thread lead variation per one spindle rotation is specified by address K.

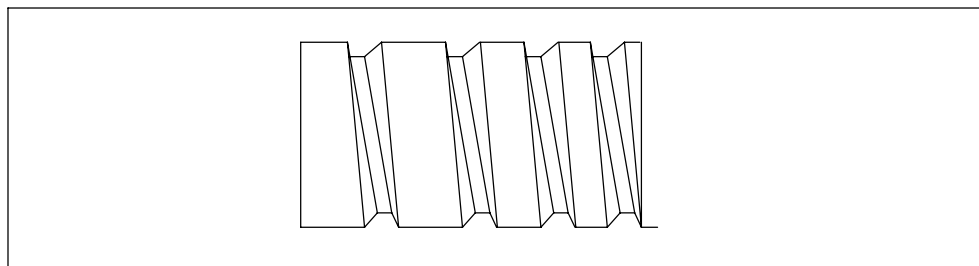


Fig. 2-23 Variable lead thread

Table 2-8 Upper limit of feedrate at end point

	Upper limit
mm output	500 mm/rev
inch output	50 inch/rev

$$S \times (F + \frac{K}{2} + KN) \leq \text{max. cutting feedrate}$$

## Feedrate at end point

Specify the commands so that the feedrate at the end point will not be a negative value.

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

## Notice

In the continuous block thread cutting for variable lead thread cutting, distribution of command pulses is interrupted at joints between blocks.

If a K command is outside the programmable range, an alarm occurs.

If address Q is designated in the G34 block, an alarm occurs.

## 2.3 Reference point return

### 2.3.1 Automatic return to reference point (G28)

#### Format

G28 X... Z... ;

With the commands of "G28 X(U)... Z(W)... (C(H)... Y(V)...);", 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. The axes not designated in the G28 block are not returned to the reference point.

In case incremental encoders are used, manual reference point return needs to be carried out before using G28.

#### Reference position

The reference position is a fixed position on a machine tool to which the tool can easily be moved by the reference position return function. For example, the reference position is used as a position at which tools are automatically changed. Up to four reference positions can be specified by setting coordinates in the machine coordinate system in MD 34000, REFF\_SET\_POS.

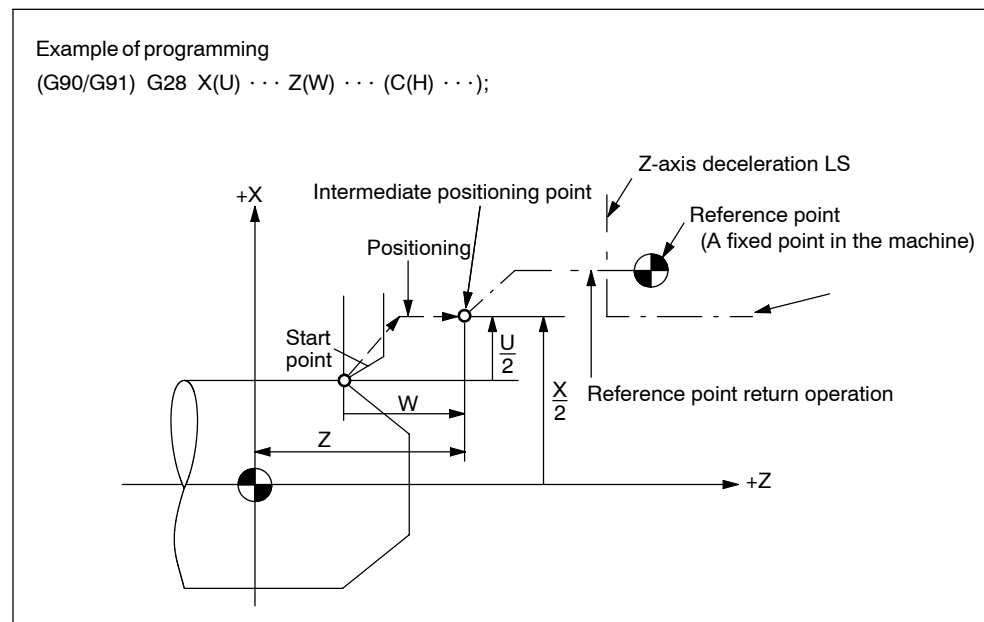


Fig. 2-24 Reference point return

---

### 2.3 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.

The 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 reference point return specification allows the axes to return to the reference point.
- 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.

---

#### Notice

Before specifying the G28 command, the tool position offset mode and nose R offset mode should be canceled. If the G28 command is specified without canceling these modes, they are canceled automatically.

---



### 2.3.2 Reference point return check (G27)

#### Format

G27 X... 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(U)... Z(W)... (C(H)... Y(V)...);".

In the G27 mode, the function checks whether or not the axes positioned by the execution of these commands in the simultaneous 2-axis control mode are located at the reference point. For the axes not specified in this block, 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 (alarm 61816, "axes not reference") 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 position 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 function is not canceled by the G27 command.
- The reference point return check is not executed if G27 is executed in the machine lock ON state.

## 2.3 Reference point return

## 2.3.3 Second to fourth reference point return (G30)

## Format

G30 Pn X... Z... ;

With the commands of "G30 Pn X(U)... Z(W)... (C(H)... Y(V)...)", the axes are moved to P2 (second reference point), P3 (third reference point\*), or P4 (fourth reference point) in the simultaneous 3-axis control mode after the positioning at the specified intermediate positioning point. If "G30 P3 U-40. W30.;" is specified, the X- and Z-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 2-9 Reference points

2nd reference point	REFP_SET_POS[1]
3rd reference point	REFP_SET_POS[2]
4th reference point	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 2.3.1, "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, alarm 61816 "axes not reference" occurs.

#### 2.3.4 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.

### 2.3 Reference point return

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.

## 2.4 Tool retract (G10.6)

To replace the tool damaged during machining or to check the status of machining, the tool can be withdrawn from a workpiece. In fact, a machine specific sequence can be initiated. Therefore, please refer to the machine tool builders documentation for details.

### Format

G10.6 X... Z... ; Activation

G10.6 ; Deactivation

X, Z :

In incremental mode, retraction distance from the position where the retract signal is turned on. In the absolute mode, retraction distance to an absolute position.



---

### Warning

The retraction axis and retraction distance specified in G10.6 need to be changed in an appropriate block according to the figure being machined. Be very careful when specifying the retraction distance;  
An incorrect retraction distance may damage the workpiece, machine, or tool.

---

[illegible]

## Movement Control Commands

### 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... Z...

The above command is called a dimension word.

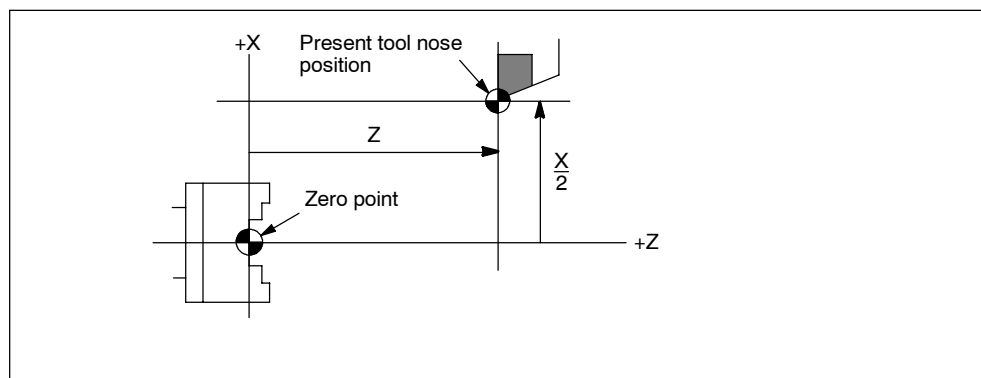


Fig. 3-1 Tool position specified by X... Z...

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

1. Machine coordinate system  
G code system A, B, C: G53
2. Workpiece coordinate system  
G code system A: G50  
G code system B, C: G92
3. Local coordinate system  
G code system A, B, C: G52

### 3.1 The coordinate system

#### 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 point 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

G53 X... Z...

X, 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.

#### Compensation function cancel

When the G53 command is specified, cancel the tool nose radius compensation and tool offset.

#### 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 whenever manual reference position return is applied after power-on, so that the reference position is at the coordinate values set using MD 34100, REFP\_SET\_POS.



### 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 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 (G50 in G code system A) in the program
2. Manually, using the HMI panel

#### Format

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

#### Explanations

The coordinate system for a workpiece is set in such a way that a point on the tool, for example, the tip of the tool, is regarded as positioned to determined coordinates. Assuming "X.. Z..." is an incremental command value, the work coordinate system is defined in such a way that the current tool position is identical with the sum of the specified incremental values and the coordinates of the previous tool position.

### 3.1.3 Resetting the work (G92.1)

With G92.1 X.. (G code system A: G50.3 P0), 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

### 3.1 The coordinate system

#### 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 predefined workpiece coordinate systems.

##### 1. G92 (G50)

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 panel.

A workpiece coordinate system can be selected by determining a G code from G54 to G59, and G54 P{1...100}.

Workpiece coordinate systems are set up subsequently to reference position return after power-on. The default coordinate system after power-on is G54.

#### Examples

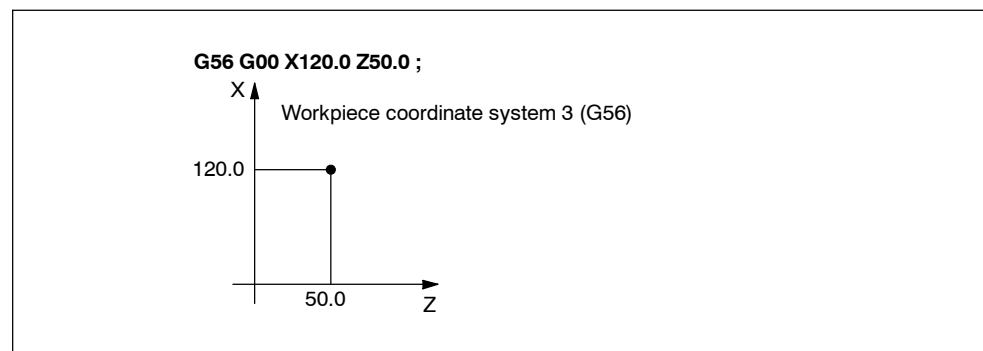


Fig. 3-2 Workpiece coordinate system G56

### 3.1.5 Instantaneous mapping of the ISO functions onto the 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 ... 100} 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

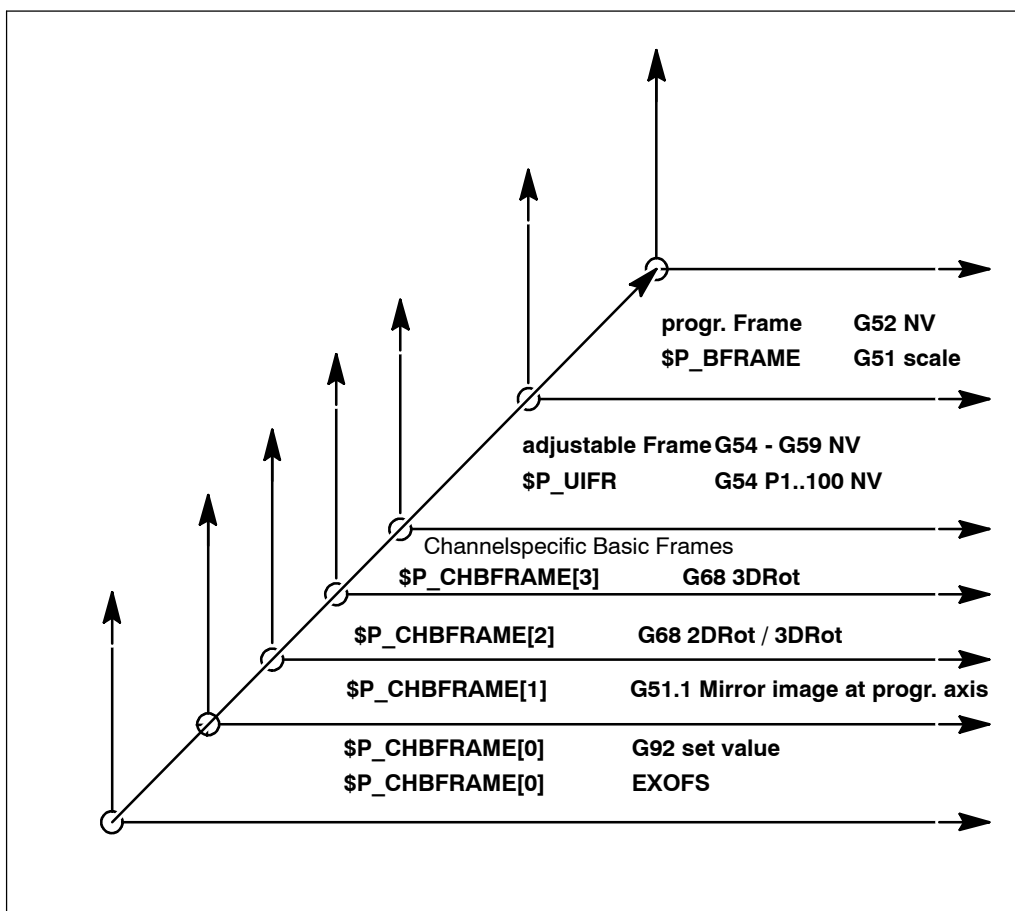


Fig. 3-3 ISO-dialect coordinate systems

### 3.1 The coordinate system

#### Format

Changing by G10:

G10 L2 Pp X... 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, Z: For an absolute command (G90), workpiece zero point offset for each axis.  
For an incremental command (G91), value to be added to the set workpiece zero point offset for each axis (the sum is set as the new offset).

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

- p=1 to 100: Workpiece zero point offset value correspond to additional workpiece coordinate systems G54 P1 ... P100
- IP: For an absolute command (G90), workpiece zero point offset for each axis.  
For an incremental command (G91), value to be added to the set workpiece zero point offset for each axis (the sum is set as the new offset).

Changing by using G92

G92 X... Z... ;

#### Explanations

Changing workpiece coordinate systems 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 workpiece coordinate systems by using G92

A workpiece coordinate system (selected with a code from G54 to G59 and G54 P{1 ...100}) is shifted to set a new workpiece coordinate system by specifying G92 X... Z.... This way, the current tool position is made to match the specified coordinates. If X, 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.

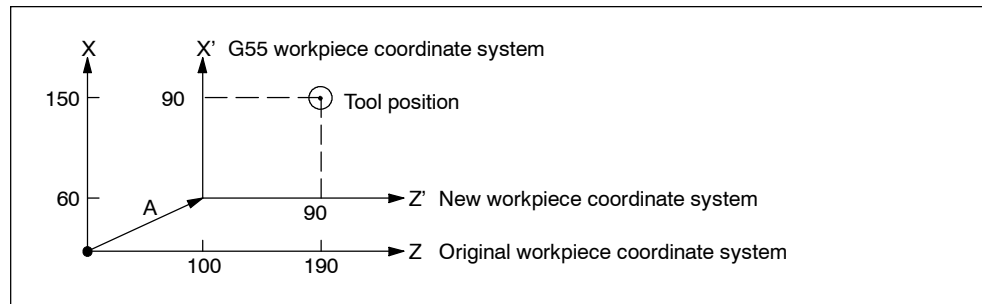
**Example**

Fig. 3-4 Setting of coordinate system with incremental values (G code system A)

**Note**

Siemens frames and ISO dialect workpiece coordinate systems are using a common storage area. In other word, changing a frame in Siemens mode will effect the relevant workpiece coordinate system used in ISO dialect mode.

ISO Dialect mode	Siemens mode
G54	G54
G55	G55
G56	G56
G57	G57
G58	G505
G59	G506
G54 P1 ... 48	G507 ... G554
G54 P49 ... !00	G ...
G92	Basic frame

**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

## 3.1 The coordinate system

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\_SYSTEM\_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. 3-5 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.

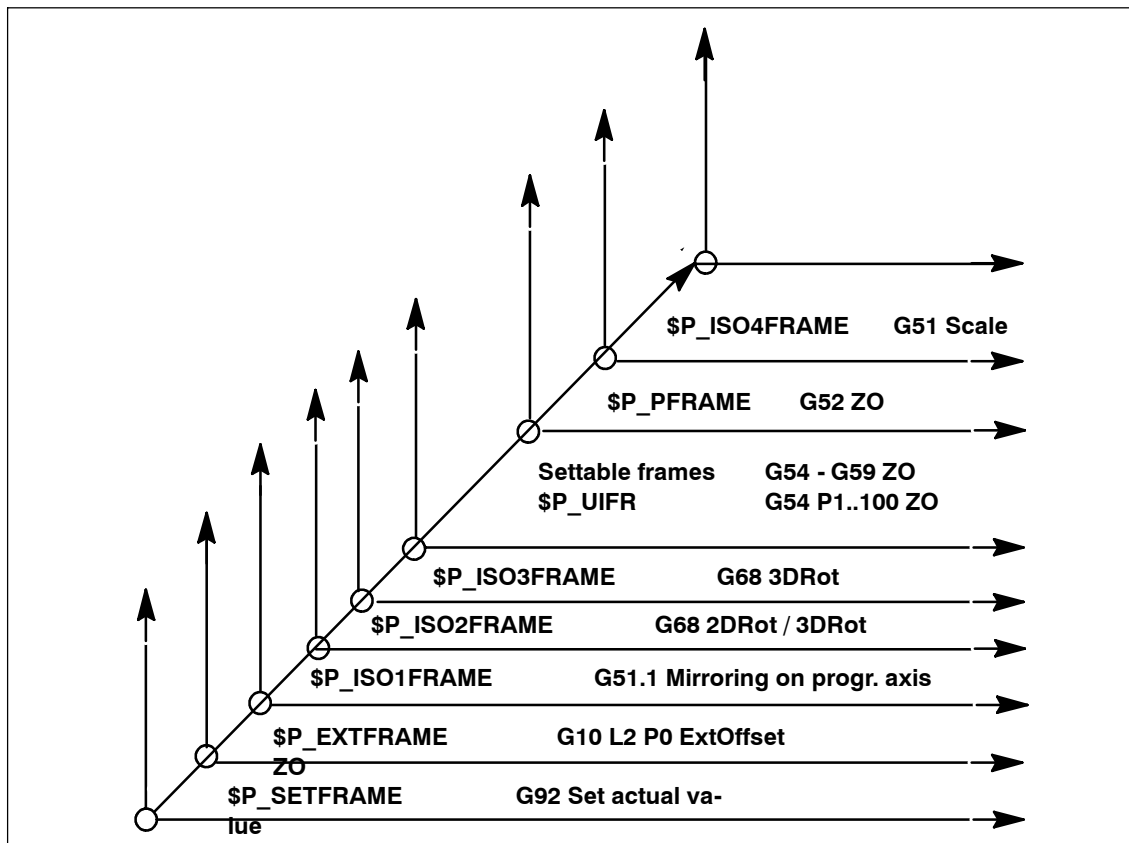


Fig. 3-5 Mapping of the ISO functions to the ISO frames and Siemens frames

**Notice**

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_MASK Bit 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

## 3.1 The coordinate system

**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_MASK Bit 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_MASK Bit 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_MASK Bit 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_MASK Bit 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_MASK Bit 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".

## 3.2 Determining the coordinate value input modes

This section describes the commands used to input coordinate values.

### 3.2.1 Absolute/incremental designation

Axis movement data specified following an axis address determines axis movement distance in either incremental or absolute values.

By using addresses X, Z, C, Y, U, W, H, and V, it is possible to use both incremental and absolute values.

#### Command format

- Absolute commands

To specify axis movement distance in an absolute value, use addresses X, Z, and C.

Example: X... Z... C... ;

- Incremental commands

To specify axis movement distance in an incremental value, use addresses U, W, and H.

Example: U... W... H... ;

- Use of both incremental and absolute commands in the same block

It is allowed to use both incremental and absolute values in the same block.

Example: X... W... ;  
U... Z... ;

If addresses that represent the same axis are specified in the same block like "X... U... ;", the address specified later becomes valid.

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

Table 3-1 Absolute and incremental commands and meaning

Address	Command value		Meaning (description)
X	Absolute	Diametric value	Position in the X-axis direction
Z		-	Position in the Z-axis direction
C		-	Position in the C-axis direction
Y		-	Position in the Y-axis direction

## 3.2 Determining the coordinate value input modes

Table 3-1 Absolute and incremental commands and meaning, continued

Address	Command value		Meaning (description)
U	Incremental value	Diametric value	Movement distance in the X-axis direction
W		-	Movement distance in the Z-axis direction
H		-	Movement distance in the C-axis direction
V		-	Movement distance in the Y-axis direction
I	Incremental value	Radial value	X-axis direction component of the distance to the center of arc viewed from the start point of arc
K		-	Z-axis direction component of the distance to the center of arc viewed from the start point of arc
J		-	Y-axis direction component of the distance to the center of arc viewed from the start point of arc
R	Incremental value	-	Direct designation of arc radius

Since a diametric value is specified for addresses X and U, actual axis movement distance is a half the specified value.

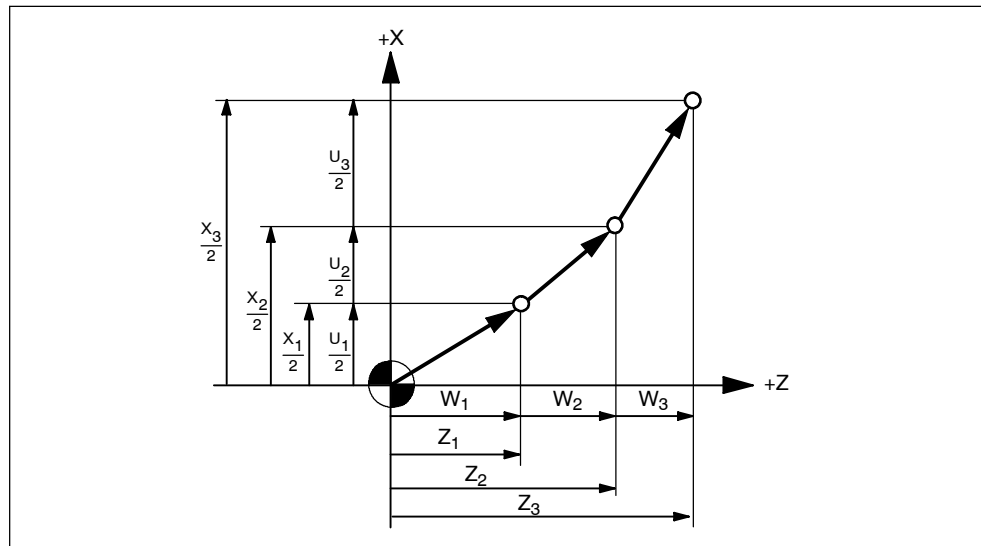


Fig. 3-6 Absolute and incremental coordinate values

### 3.2 Determining the coordinate value input modes

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

Table 3-2 Function of G90 and G91 commands

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

Table 3-3 Valid address for G90/G91 designation

Address		G90 command	G91 command
	X, Z, C, Y	Absolute	Incremental
	U, W, H, V	Incremental	Incremental

Example: With the commands of "G91 G00 X40. Z50.;" axis movement commands are executed as incremental commands.

#### Auxiliary data for circular interpolation

The auxiliary circular interpolation data I, J, K, and R are always interpreted as incremental commands.

---

##### Notice

It is not allowed to specify G90 and G91 in the same block. If both of these G codes are specified in the same block, the one specified later becomes valid. For example, if the commands of "G01 G90 X80. G91 Z60.;" are specified in a block, G91 specified later becomes valid and all axis movement commands (X80. and Z60.) are interpreted as incremental commands.

---

## 3.2 Determining the coordinate value input modes

## 3.2.2 Diametric and radial commands for X-axis

To specify X-axis commands, address X or U is used and dimensions are usually specified in diametric values.

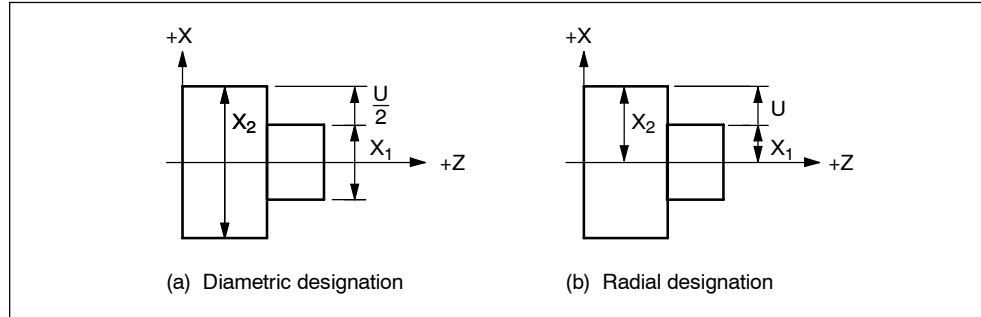


Fig. 3-7 Coordinate values

Table 3-4 Use of diametric and radial designation

Item	Diametric Designation	Radial Designation
Address X command	Diametric value	Radial value
Address U command	Diametric incremental value	Radial incremental value
X-axis position display	Diametric value	
Tool position offset amount	Diametric value	
Tool coordinate data for tool coordinate system	Diametric value	
Nose R amount	Radial value	
Feedrate F in the X-axis direction	Radial value/rev, Radial value/mm	
Radius designation for circular interpolation (I, K, J, R)	Radial value	
G90 to G94, G70 to G76 Chamfering, rounding, multiple chamfering parameters	Radial value	

### 3.2 Determining the coordinate value input modes

#### 3.2.3 Inch/metric input designation (G20, G21)

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

Table 3-5 Dimension unit selection G codes

G code	Function	Group
G20 (G70, G code syst. C)	Input in “inch” system	06
G21 (G71, G code syst. C)	Input in “mm” system	06

#### Command format

G20 (G70) and G21 (G71) 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.

#### Supplements to the dimension unit designation commands

A parameter is used to select “inch/mm”. Therefore, the state when the power is turned ON is determined by the setting for this parameter.

If the dimension unit system should be switched over during the execution of a program, the tool position offset and nose R offset function must be canceled before the switching over of the dimension unit system.

After switching over the dimension unit system between G20 and G21, the following processing must be accomplished.

- Set the coordinate system before specifying axis move commands.
- If position data are displayed in a workpiece coordinate system, or when an external position data display unit is used, reset the present position data to “0”.

The tool offset amounts stored in memory are treated in a different manner between the G20 and G21 modes.

Table 3-6 Tool offset amounts in G20 (G70) and G21 (G71) modes

Stored offset amount	in the G20 (G70) (inch system) mode	in the G21 (G70) (mm system) mode
150000	1.5000 inch	15.000 mm

### 3.2.4 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).

#### Format

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

X: Dwell time (decimal point representation)

P: Dwell time (integer representation)

There are two different methods how to execute the programmed dwell time:

MD \$MC\_EXTERN\_FUNCTION\_MASK

Bit2 = 0: Dwell always in seconds [s]

Bit2 = 1: Dwell in seconds (G94 mode) or spindle rotations (G95 mode)

The execution of programmed commands is suspended for the length of time in the feed per minute mode (G94) and a number of spindle rotations in the feed per revolution mode (G95) determined by the address X or P by specifying G04 X...; or G04 P...;

The block used to determine dwell is not allowed to contain commands other than G04 commands.

#### Example

G94 G04 X1000 ;

Standard notation:  $1000 * 0.001 = 1\text{s dwell}$

Pocket calculator notation: 1000s dwell

G95 G04 X1000 ;

Standard notation:  $1000 * 0.001 = 1\text{ rev dwell}$

Pocket calculator notation: 1000 rev dwell

The use of standard notation or pocket calculator notation is decided by MD EXTERN\_FLOATINGPOINT\_PROG.



## 3.4 Tool offset functions

The following three kinds of tool offset functions are provided: tool position offset function, nose R offset function, and tool radius offset function.

### 3.4.1 Tool offset data memory

The memory area where the data of the offset functions and coordinate system setting is called the tool offset data memory.

### 3.4.2 Tool position offset

The tool position offset function adds the offset amount to the coordinate value specified in a program when a tool offset number is specified and moves the nose R to the position obtained by the addition.

### 3.4.3 Tool nose radius compensation function (G40, G41/G42)

Since the nose of a cutting tool is rounded, overcuts or undercuts occur in taper cutting or arc cutting since offset simply by the tool position offset function is not satisfactory. How such problems occur is shown in Fig. 3-8. The tool nose radius compensation function called by G41 and G42 compensates for an error to finish the workpiece to the programmed shape.

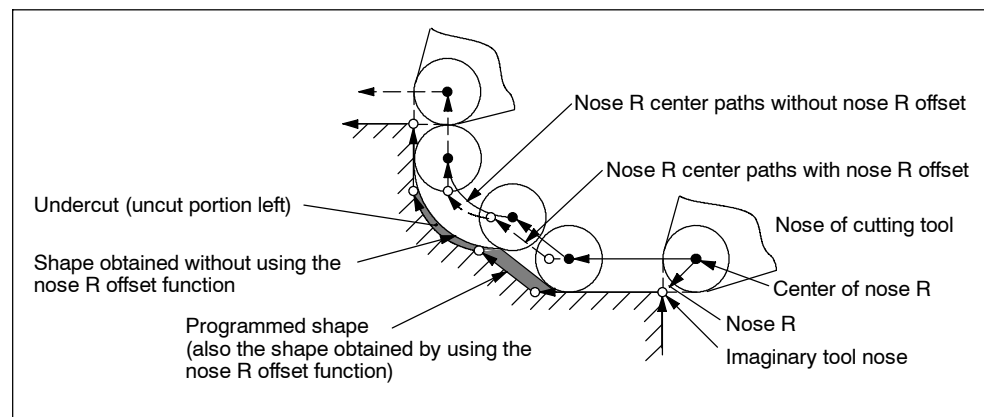


Fig. 3-8 Tool nose radius compensation function

### 3.4 Tool offset functions

#### Nose R offset amount

The term “Nose R offset amount” means the distance from the tool nose to the center of nose R.

- Setting the nose R offset amount

For the nose R offset amount, set the radius of the circle of the tool nose without a sign.

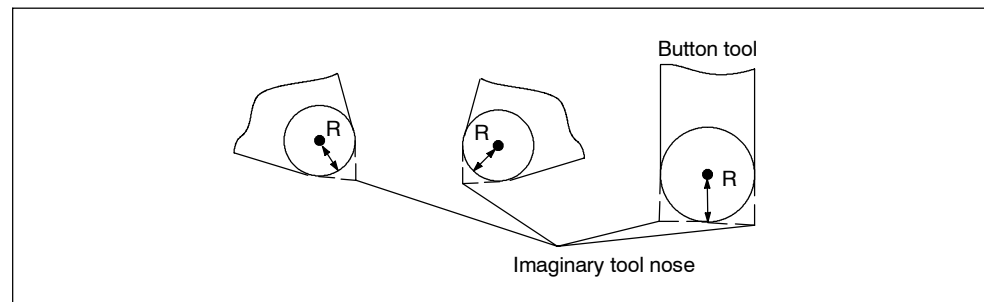


Fig. 3-9 Setting the nose R offset amount and imaginary tool nose

#### Designation of imaginary tool nose position (control point)

- Control point memory

The position of the imaginary tool nose viewed from the center of the nose R is expressed using a 1-digit number, 0 to 9. This is called the control point. The control point should be written to the NC memory in advance as with the tool data.

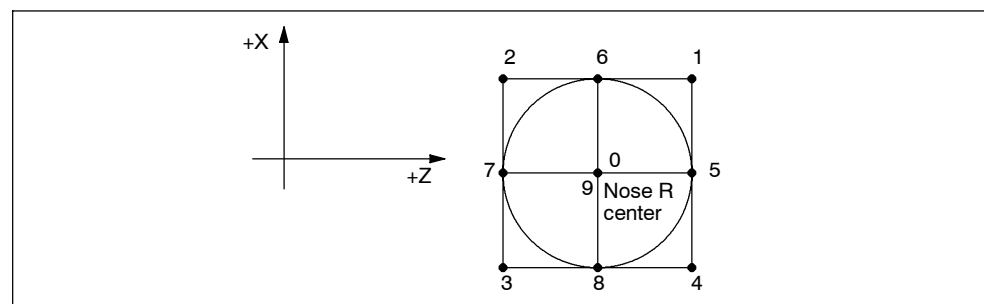


Fig. 3-10 Control point

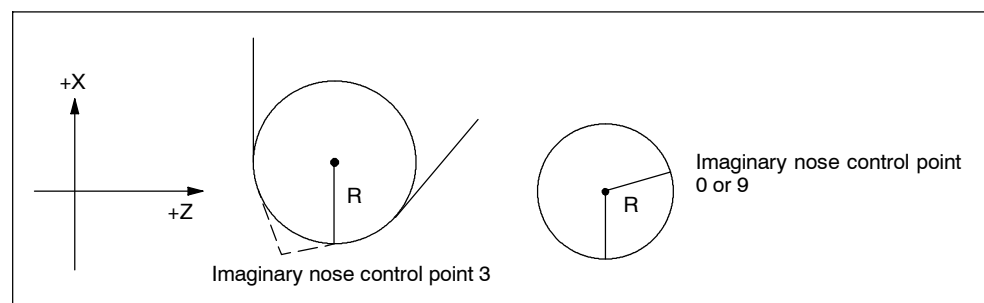


Fig. 3-11 Example of control point setting

## Control points and programs

When control points 1 to 8 are used, the imaginary tool nose position should be used as the reference to write a program. Write the program after setting a coordinate system.

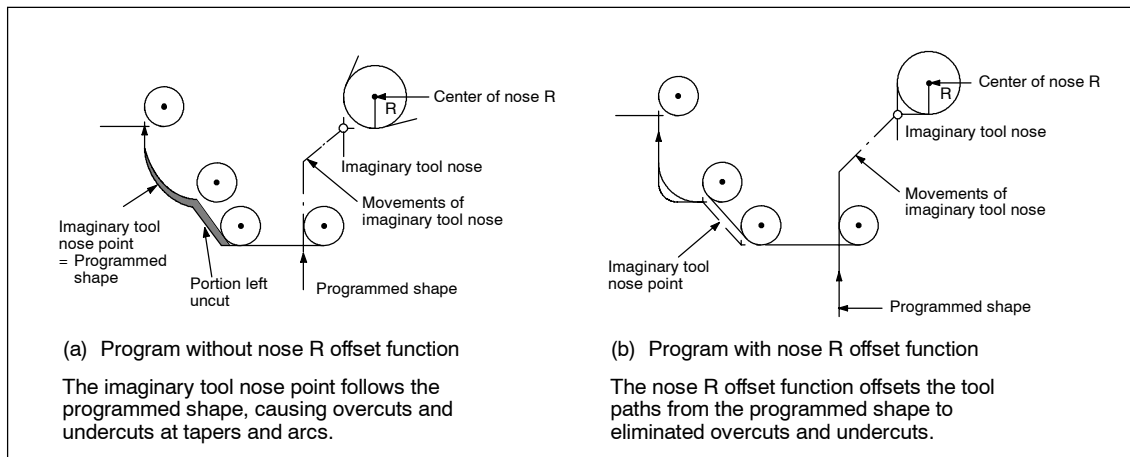


Fig. 3-12 Program and tool movements for control points 1 to 8

When control points 0 or 9 is used, the center of nose R should be used as the reference to write a program. Write the program after setting a coordinate system. If the nose R offset function is not used, the program shape must not be different from the shape to be machined.

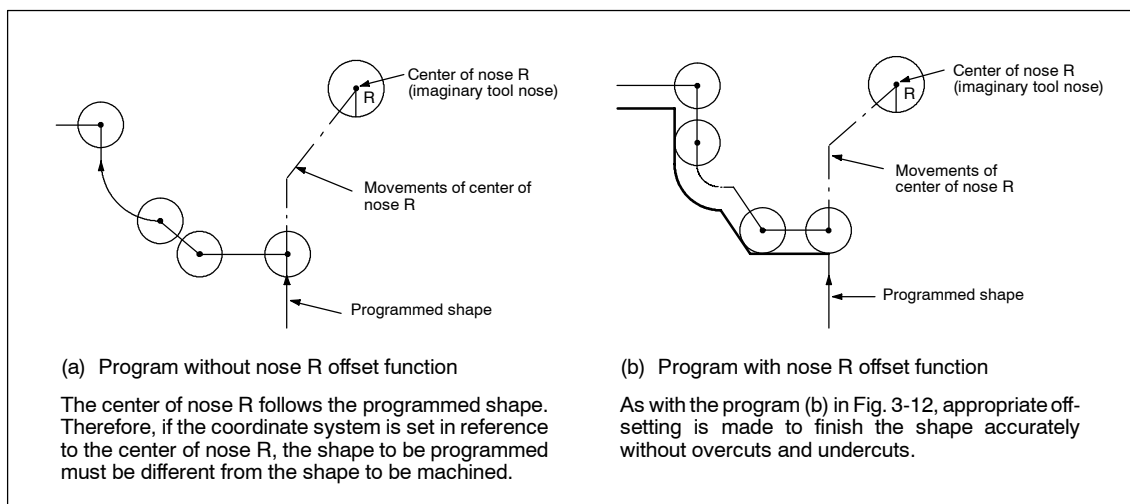


Fig. 3-13 Program and tool movements for control point 0 or 9

## 3.4 Tool offset functions

**Nose R offset commands**

- Designation of tool offset amount  
The tool offset amount is called by T command.
- Designation of tool nose radius compensation function ON  
To designate ON/OFF of the tool nose radius compensation function use the following G codes.

Table 3-7 G codes used for turning ON/OFF tool nose radius compensation function

G code	Function	Group
G40	Tool nose radius compensation cancel	07
G41	Tool nose radius compensation, left (nose R center is at the left side)	07
G42	Tool nose radius compensation, right (nose R center is at the right side)	07

G40 and G41/G42 are modal G codes in group 07, and once designated the specified G code mode remains valid until another G code is specified. When the power is turned ON or the CNC is reset, the G40 mode is set.

To enter the tool nose radius compensation mode, specify either G41 or G42 with a T code.

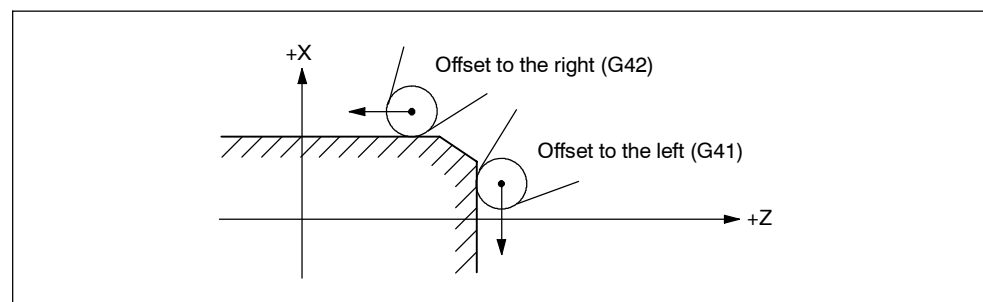


Fig. 3-14 Designation of tool nose radius compensation direction

The tool nose radius compensation direction can be changed over between “to the right” and “to the left” by specifying G41 or G42 during the execution of a program. It is not necessary to cancel the nose R offset mode by specifying G40 or deselecting the tool before changing over direction of offset. To cancel the tool nose radius compensation mode, specify G40.

### Outline of tool nose radius compensation movements

Fig. 3-15 shows how the tool nose radius compensation function is executed.

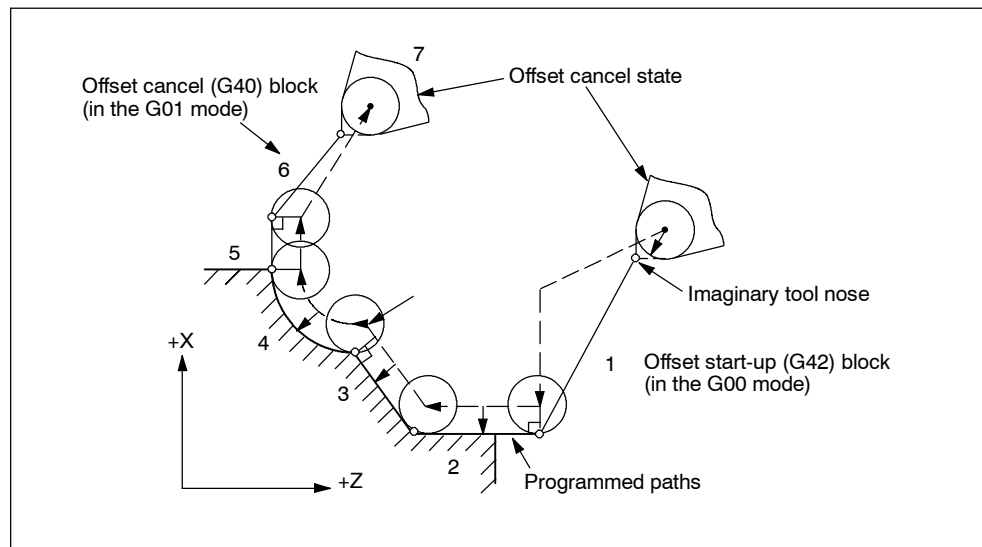


Fig. 3-15 Outline of tool nose radius compensation movements (G42, control point 3)

- In the offset cancel state, the imaginary tool nose position 7 agrees with the point specified in the program 1.
- In the offset mode, the center of nose R is offset by the nose R amount from the programmed paths and it follows the offset paths. Therefore, the imaginary tool nose position does not agree with the programmed point. Note that the present position display shows the position of the imaginary tool nose.
- At the offset start-up block 1 and cancel block 6, the movements to link the compensation mode and compensation cancel mode are inserted. Therefore, special attention must be paid for specifying the offset start-up and cancel blocks.

#### Notice

1. The nose R offset function can be used for circular interpolation specified by radius designation.
2. It is allowed to specify a subprogram (M98, M99) in the offset mode. The nose R offset function is applied to the programmed shape which is offset by the tool position offset function.

### 3.4 Tool offset functions

#### Entering the offset mode

The compensation mode is set when both of a tool offset (by a T code) and G41/G42 are specified. More precisely, the compensation mode starts at the time when the AND condition of a T code and a G code is satisfied. There are no differences whichever of these codes is specified first (see Fig. 3.24). The initial movement when the offset mode starts in the offset cancel state is called the start-up motion.

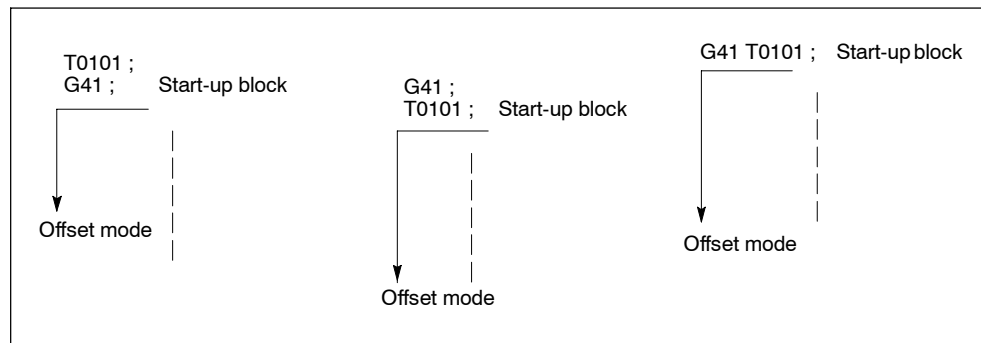


Fig. 3-16 Compensation mode entry methods

## Example of programming

```

N1 G92 X140. Z20. ;
N2 G00 S1700 M03 T0202 ;
N3 (G00) G42 X0 Z5. ;      _____ Nose R offset start-up block
N4 G01 Z0 F0.2 ;
N5 X20. ;
N6 Z-20. ;
N7 X30. W-15. S1100 ;
N8 G1 W-20. R3. ;          _____ (Rounding)
N9 G1 X50. K-3. S700 ;     _____ (Chamfering)
N10 G01 Z-70. ;
N11 G02 X90. Z-90. R20. S360 ; _____ (Arc designated by R command)
N12 G01 X110. S300 ;
N13 G04 U0 ;               _____ (Dwell for making sharp edge)
N14 (G01) Z-110. ;
N15 X120. ;
N16 G00 X140. Z30. ;      _____ Nose R offset cancel block
N17 T0 G40 ;
...

```

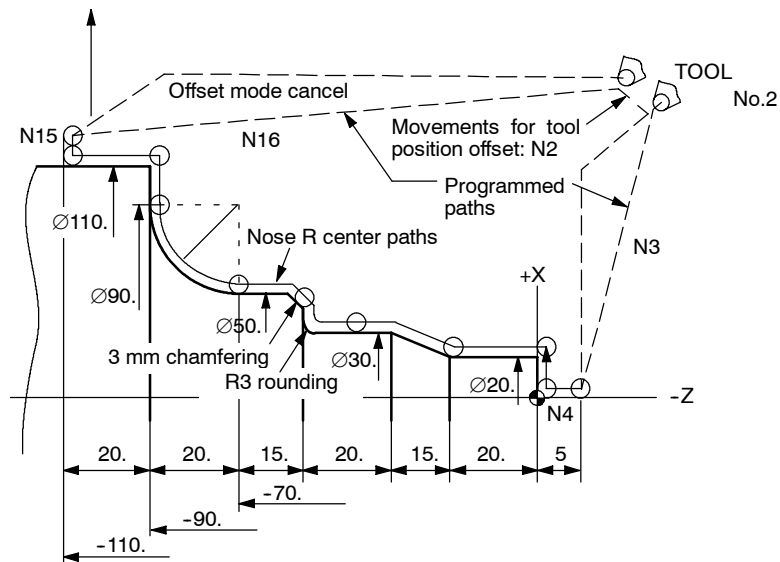


Fig. 3-17 Example of programming

## 3.5 Spindle function (S function)

### 3.5.1 Spindle command (S5-digit command)

A spindle speed can be directly specified by entering a 5-digit number following address 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.

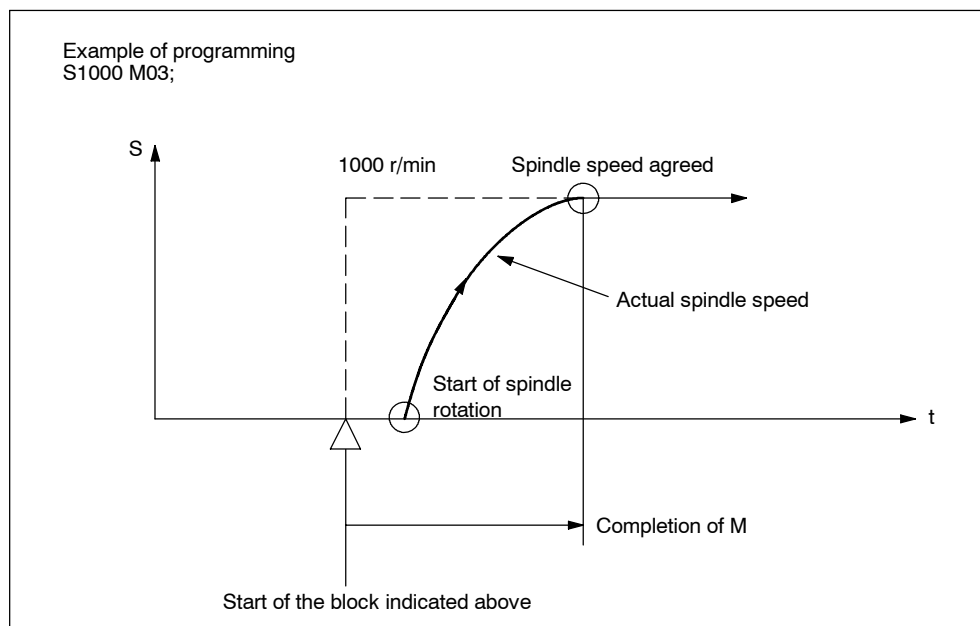


Fig. 3-18 Spindle speed command

- For the output of S5-digit commands, it is possible to add the control function implemented by the PLC can be added by the NC. In this case, it is possible to set the spindle speed in manual operation to the speed that corresponds to the specified S command by using the rotary switch on the machine operation panel. For details, refer to the manuals published by the machine tool builder.
- 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.
- 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.



- Spindle speed override is possible for the specified S code.
- For the machine that has the gearbox with which gear range can be changed by specifying an M code, specify the M code to select an appropriate gear range before specifying an S code. For the number of gear ranges and the available spindle speed range in the individual gear ranges, refer to the manuals published by the machine tool builder.

### 3.5.2 Constant surface speed control (G96, G97)

The G codes indicated in table 3-7 are used for the constant surface speed control function. G96 and G97 are modal G code of 02 group.

Table 3-8 G codes for constant surface speed control

G code	Function	Group
G96	Constant surface speed control ON	02
G97	Constant surface speed control cancel	02

#### Constant surface speed control ON (G96)

With the commands of "G96 S... (M03) ;", the workpiece surface speed is designated by a maximum 5-digit number following address S. The unit used for specifying the surface speed is indicated in Table 3-9.

Table 3-9 Units of surface speed designation

	Unit
mm	m/min
inch	ft/min

In the constant surface speed control mode, the NC assumes the present value of the X-axis as the workpiece diameter and calculates the spindle speed every 32 msec so that the specified surface speed is maintained. The specified surface speed can be changed by specifying a required S code in the following blocks.

## 3.5 Spindle function (S function)

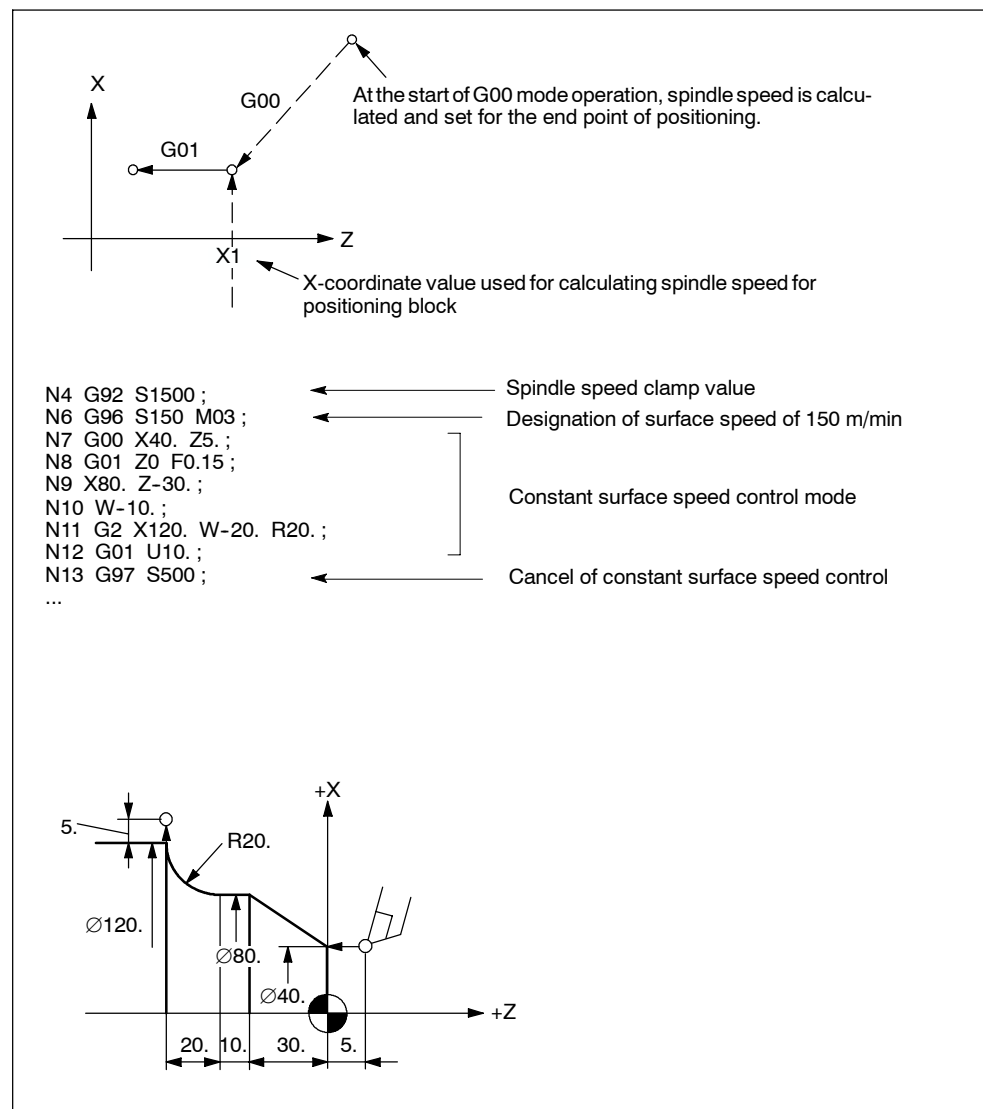


Fig. 3-19 Constant surface speed

**Canceling the constant surface speed control (G97)**

Specify a spindle speed (r/min) by a maximum of 5-digit number following address S with the commands "G97 S... (M03) ;". The constant surface speed control mode is canceled, and the spindle rotates at the specified spindle speed.

### Spindle gear range selection

For the machine that has the gearbox with which gear range can be changed by specifying an M code, specify the M code to select an appropriate gear range before specifying G96. For details, refer to the manuals published by the machine tool builder.

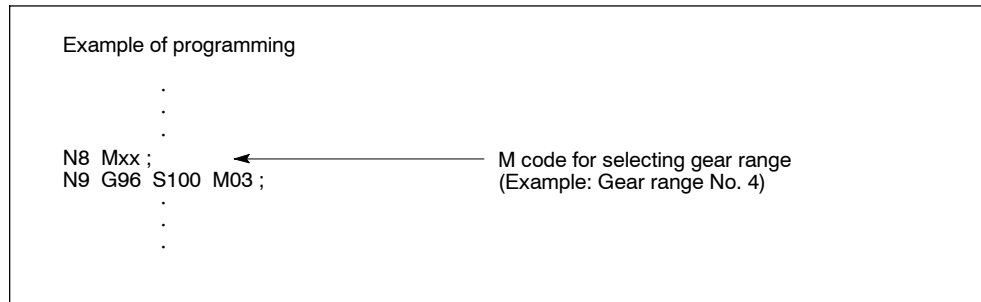


Fig. 3-20

### Supplements to the constant surface speed control commands

- To execute the constant surface speed control, set the G92 coordinate system or a workpiece coordinate system so that the X-coordinate value of the center-line of the spindle will be "0" and program the operation on this coordinate system. With this, X-coordinate values in a program represent the diameter of workpiece accurately.

## 3.6 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.

## 3.7 Miscellaneous function (M function)

The miscellaneous function is specified by a maximum of a three-digit number following address M. With the exception of specific M codes, the functions of M00 to M97 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.

### 3.7.1 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.

**Notice**

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

**Notice**

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.

**3.7.2 Internally processed M codes**

M codes in the range of M98 and M99 are processed by the NC.

Table 3-10 Internally processed M codes

M code	Function
M98	Subprogram call
M99	End of subprogram

**3.7.3 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  
 10841: \$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.

### 3.7 Miscellaneous function (M function)

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

### 3.7.4 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 3-11 Other general M codes

M code	Function
M03	Spindle start, forward direction
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.

### Further information

/FBFA/ SINUMERIK 840D/840Di/810D  
Description of Functions  
**ISO Dialects for SINUMERIK** (03.07 Edition)

## Notes

[illegible]



## Enhanced Level Commands

### 4.1 Program support functions (1)

#### 4.1.1 Canned cycles

The canned cycle function defines the four block operations of basic cutting operation, in-feed, cutting (or thread cutting), retraction, and return, in one block (to be called as one cycle).

## 4.1 Program support functions (1)

Table 4-1 Table of canned cycles

G code	Straight cycle	Taper cycle
G90 Cutting cycle (OD cutting)	<p>G90 X (U) ... Z (W) ... F ... ;</p>	<p>G90 X (U) ... Z (W) ... R ... F ... ;</p>
G92 Thread cutting cycle	<p>G92 X (U) ... Z (W) ... F ... ;</p>	<p>G92 X (U) ... Z (W) ... R ... F ... ;</p>
G94 Face cutting cycle	<p>G94 X (U) ... Z (W) ... F ... ;</p>	<p>G94 X (U) ... Z (W) ... R ... F ... ;</p>

## Cutting cycle commands

The cutting cycle is used for outside diameter (OD) cutting and has two kinds of cycles – straight cutting cycle and taper cutting cycle.

### Straight cutting cycle

#### Format

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

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

With the commands of “G... X(U)... Z(W)... F... ;”, straight cutting cycle is executed as indicated by sequence 1 to 4 shown in Fig. 4-1.

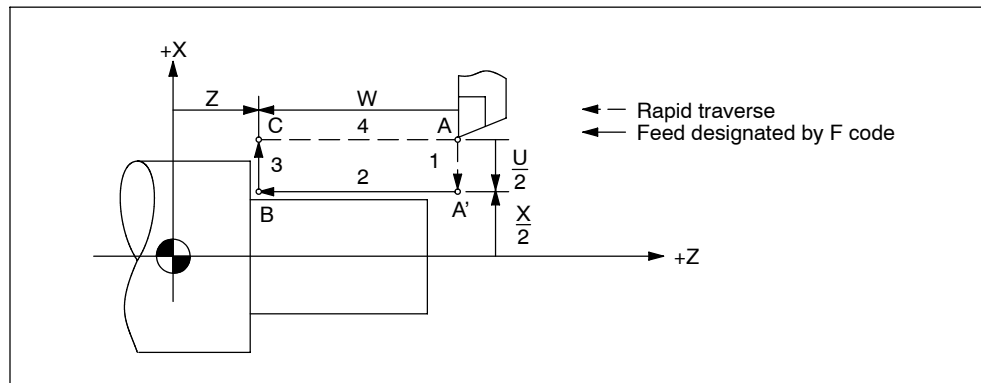


Fig. 4-1 Straight cutting cycle

Since G77 (G90, G20) is a modal G code, cycle operation is executed by simply specifying in-feed movement in the X-axis direction in the succeeding blocks.





## 4.1 Program support functions (1)

**Format**

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

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

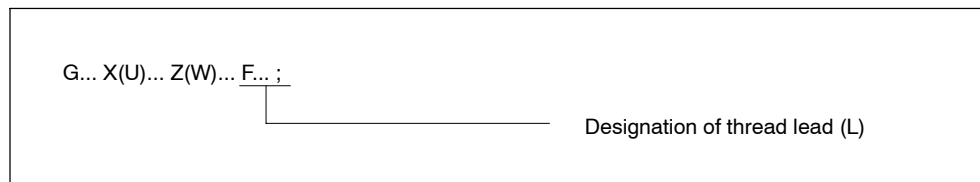
**Straight thread cutting cycle**

Fig. 4-5

With the commands indicated above, straight thread cutting cycle 1 to 4, shown in Fig. 4-6, is executed.

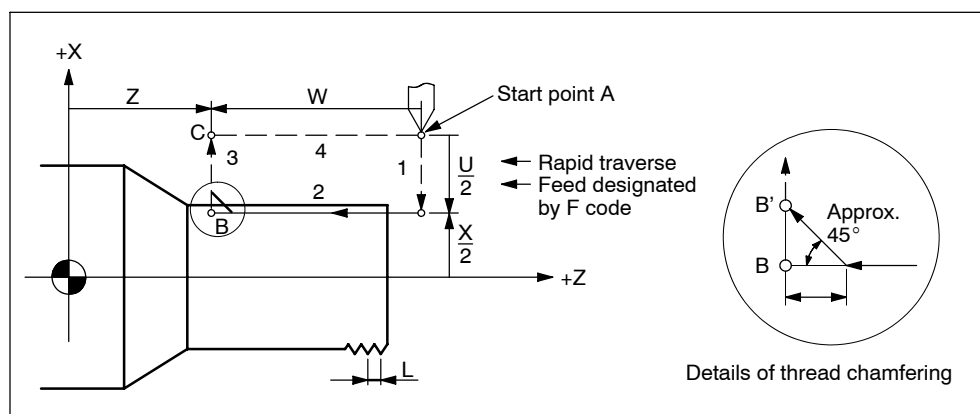


Fig. 4-6 Straight thread cutting cycle

Since G78 (G92, G21) is a modal G code, thread cutting cycle is executed by simply specifying depth of cut in the X-axis direction in the succeeding blocks. It is not necessary to specify G78 (G92, G21) repeatedly in these blocks.

## 4.1 Program support functions (1)

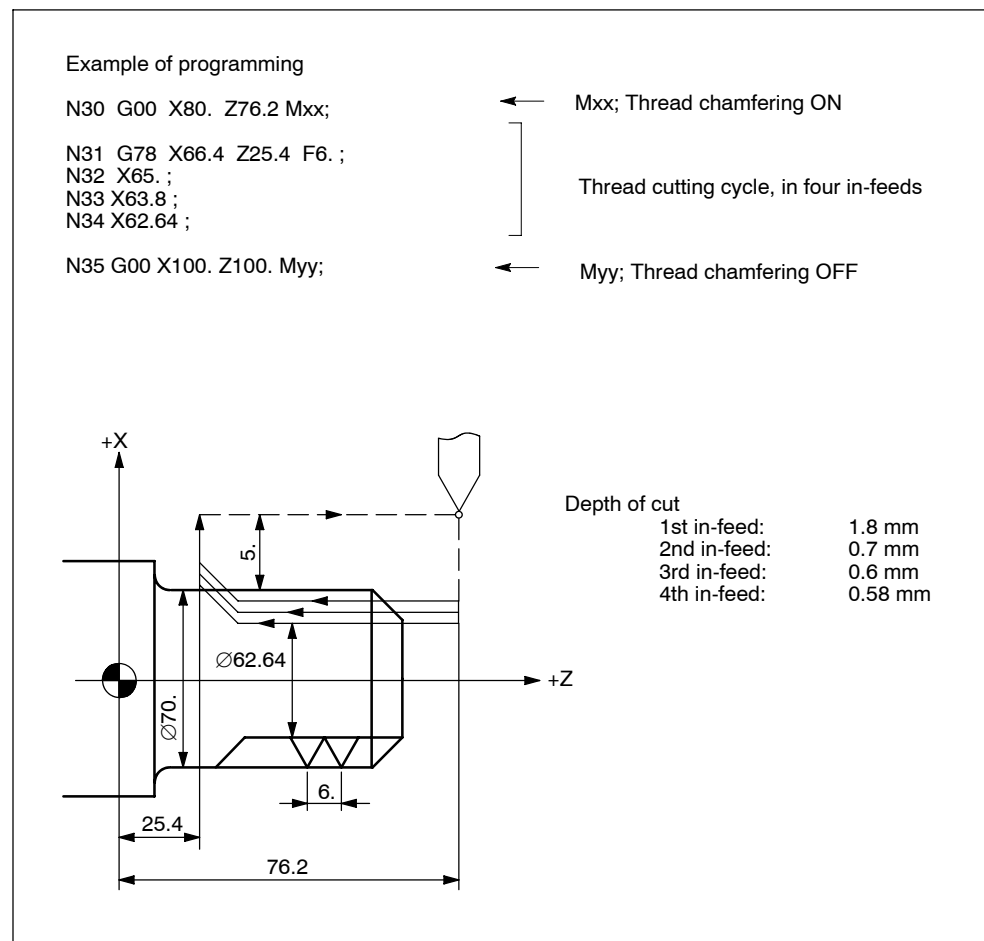


Fig. 4-7 Straight thread cutting cycle (G code system B)

- When the G78 (G92, G21) cycle is executed with the single block function ON, the cycle is not suspended halfway, but it stops after the completion of the cycle consisting of sequence 1 to 4.
- Thread chamfering can be performed in this thread cutting cycle. A signal from the machine tool initiates thread chamfering. Thread chamfering size  $\gamma$  can be set for GUD7\_ZSFI[26] in increments of 0.1L. Here, "L" represents the specified thread lead.

It is recommended to program the sequence that turns ON and OFF the "thread chamfering input" by using appropriate M codes.

## 4.1 Program support functions (1)

## Tapered thread cutting cycle

## Format

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

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

With the commands of “G... X(U)... Z(W)... R... F... ;” tapered thread cutting cycle of 1 to 4 as shown in Fig. 4-8 is executed.

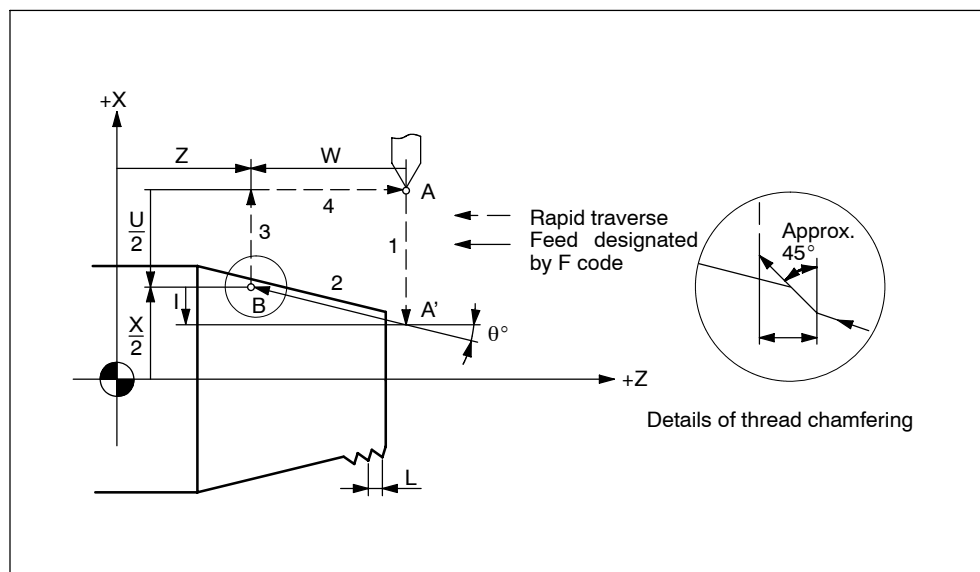


Fig. 4-8 Tapered thread cutting cycle

The sign of address R is determined by the direction viewing point A' from point B. Since G78 (G92, G21) is a modal G code, thread cutting cycle is executed by simply specifying depth of cut in the X-axis direction in the succeeding blocks. It is not necessary to specify G78 (G92, G21) repeatedly in these blocks.



## 4.1 Program support functions (1)

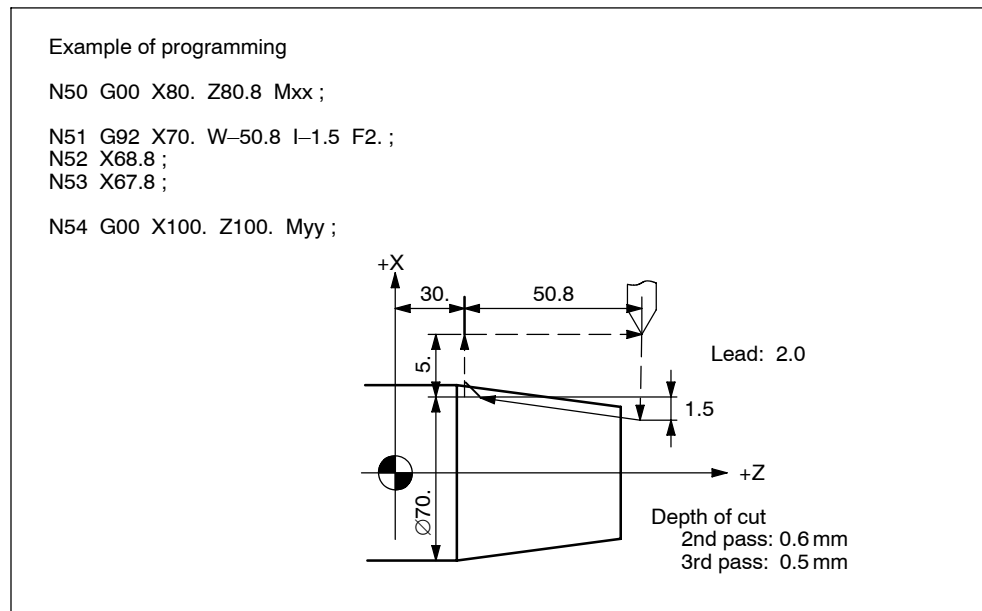


Fig. 4-9 Tapered thread cutting cycle (G code system A)

When the G78 (G92, G21) cycle is executed with the single block function ON, the cycle is not suspended halfway, but it stops after the completion of the cycle consisting of sequence 1 to 4.

The S, T, and M functions that are used as the cutting conditions for the execution of the G78 (G92, G21) cycle should be specified in blocks preceding the G78 (G92, G21) block. However, if these functions are specified in a block independently without axis movement commands, such designation is valid if the block is specified in the G78 (G92, G21) mode range.

When the CYCLE START button is pressed while the cutting tool is at start point A or chamfering completion point B, the suspended cycle is executed again from the beginning.

If the thread cutting feed hold option is not selected, the thread cutting cycle is continued even if the FEED HOLD button is pressed during the execution of thread cutting cycle. In this case, the operation is suspended upon completion of retraction operation after finishing the thread cutting cycle.

## 4.1 Program support functions (1)

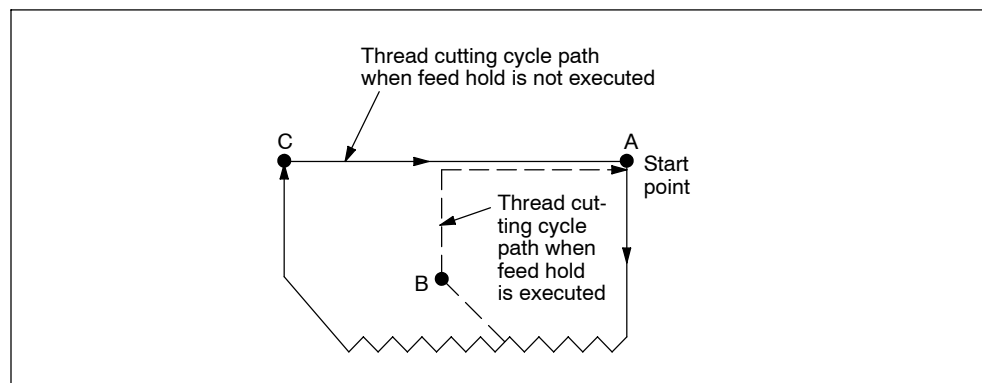


Fig. 4-10 Feed hold during thread cutting cycle

If chamfer size is "0" when the G78 (G92, G21) cycle is executed with chamfering ON, an alarm occurs.

## Straight facing cycle

## Format

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

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

With the commands of "G... X(U)... Z(W)... F... ;", straight facing cycle of 1 to 4 as shown in Fig. 4-11 is executed.

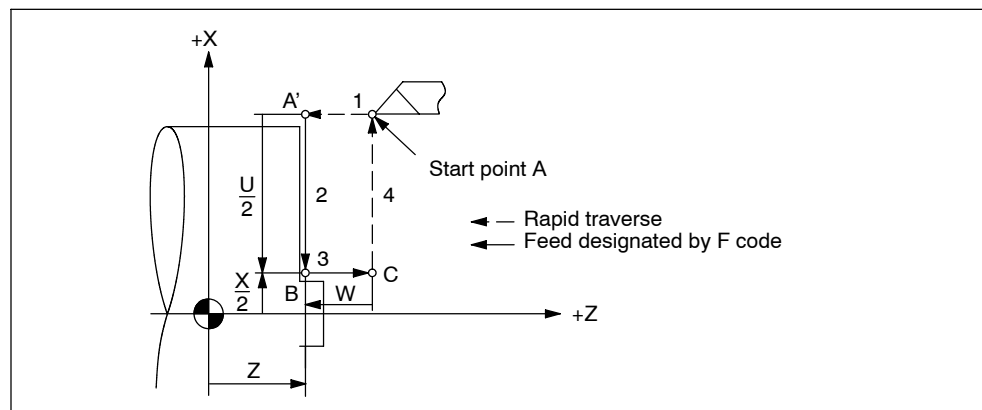


Fig. 4-11 Straight facing cycle

Since G79 (G94, G24) is a modal G code, thread cutting cycle is executed by simply specifying depth of cut in the Z-axis direction in the succeeding blocks. It is not necessary to specify G79 (G94, G24) repeatedly in these blocks.

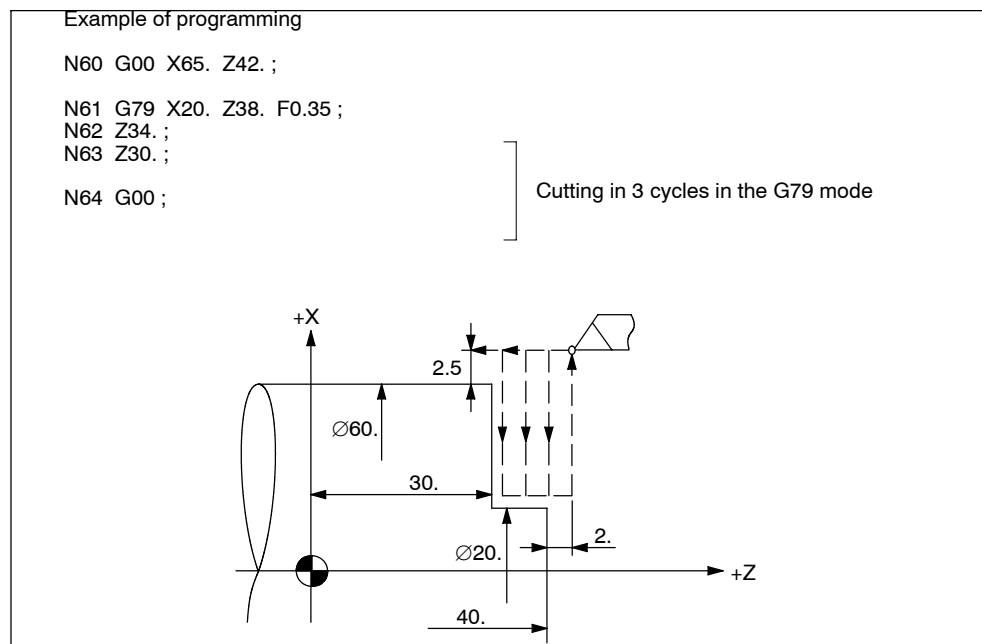


Fig. 4-12 Straight facing cycle (G code system B)

## Taper facing cycle

### Format

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

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

With the commands of “G... X(U)... Z(W)... R... F... ;”, taper facing cycle of 1 to 4 as shown in Fig. 4-13 is executed.

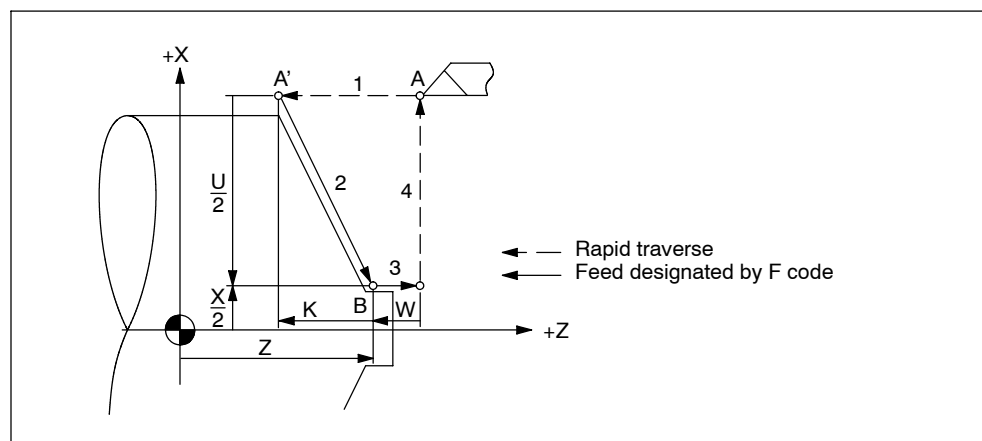


Fig. 4-13 Taper facing cycle

## 4.1 Program support functions (1)

The sign of address R is determined by the direction viewing point A' from point B.

Example of programming

N70N G00 X74. Z32. ;

N71 G79 X20. Z30. R-5.29 F0.3 ;

N72 Z25. ;

N73 Z20. ;

N74 G00 ;

Taper cutting in 3 cycles in the G79 mode

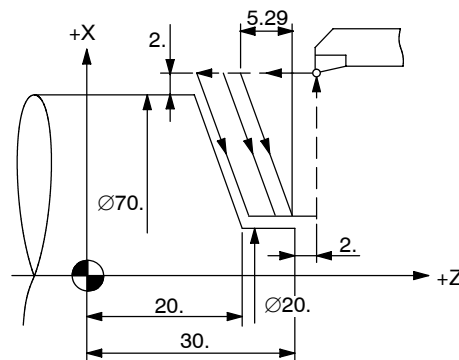


Fig. 4-14 Taper facing cycle (G code system B)

The S, T, and M functions that are used as the cutting conditions for the execution of the G79 (G94, G24) cycle should be specified in blocks preceding the G79 (G94, G24) block. However, if these functions are specified in a block independently without axis movement commands, such designation is valid if the block is specified in the G79 (G94, G24) mode range.

If the G79 (G94, G24) cycle is executed with the single block function ON, the cycle is not interrupted halfway but it stops after the completion of the cycle consisting of sequence 1 to 4.

### 4.1.2 Multiple repetitive cycles

By using the multiple repetitive cycles, programming steps can be considerably reduced due to the features that both rough and finish cutting cycles can be executed by simply defining the finishing shape, and the like.

For the multiple repetitive cycles, seven kinds of cycles (G70 to G76) are provided in G code systems A and B as indicated in Table 4-2. Note that these G codes are all non-modal G code.

Table 4-2 Cycles called by G70 to G76 (G code system A and B)

G code	Cycle name	Remark	
G70	Finishing cycle		
G71	Stock removal cycle, longitudinal axis	G70 cycle can be used for finishing	Nose R offset possible
G72	Stock removal cycle transverse axis		
G73	Contour repetition		
G74	Deep hole drilling and recessing in longitudinal axis		
G75	Deep hole drilling and recessing in transverse axis		
G76	Multiple thread cutting cycle		

The same cycles are provided in G code system C. However, different G codes are used as indicated below.

Table 4-3 Cycles called by G72 to G78 (G code system C)

G code	Cycle name	Remark	
G72	Finishing cycle	G72 cycle can be used for finishing	Nose R offset possible
G73	Stock removal cycle, longitudinal axis		
G74	Stock removal cycle transverse axis		
G75	Contour repetition		
G76	Deep hole drilling and recessing in longitudinal axis		
G77	Deep hole drilling and recessing in transverse axis		
G78	Multiple thread cutting cycle		

#### Note

The following cycle description of the a.m. cycles refers to G code system A and B.

## 4.1 Program support functions (1)

**Stock removal cycle, longitudinal axis (G71)**

By using the multiple repetitive cycles, programming steps can be considerably reduced due to the features that both rough and finish cutting cycles can be executed by simply defining the finishing shape, and the like.

There are two types of stock removal cycles.

**Type I**

The specified area is removed by  $\Delta d$  (infeed depth for stock removal) with finishing allowances  $\Delta u/2$  and  $\Delta w$  left over, whenever a contour of A to A' to B is described by an NC program.

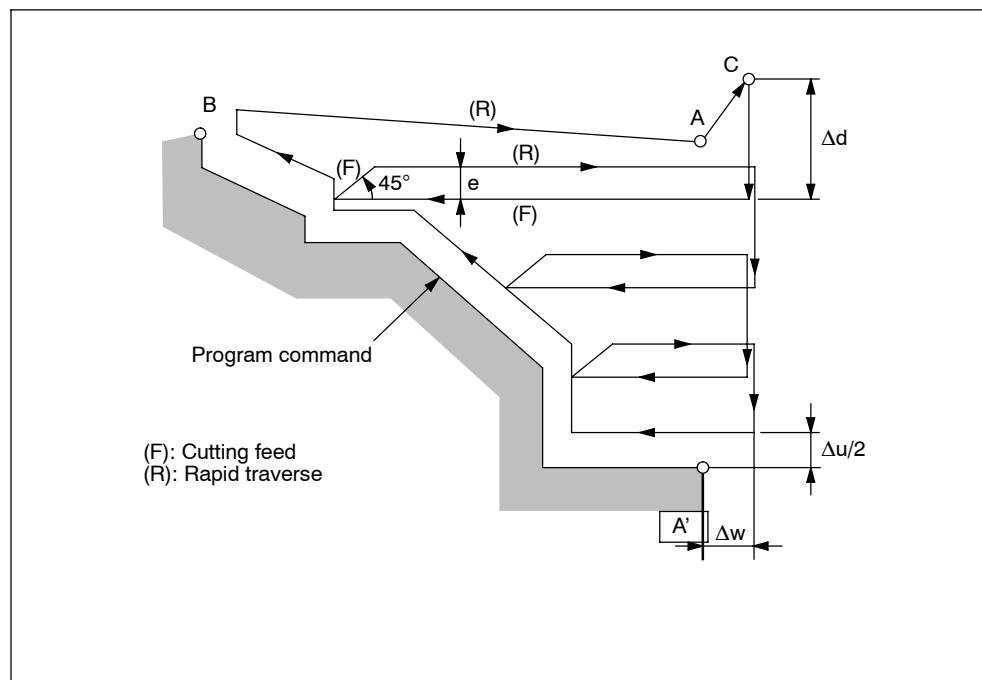


Fig. 4-15 Cutting path in stock removal in turning (type I)

**Format**

**G71 U... R... ;**

**U:** Infeed depth for stock removal ( $\Delta d$ ), radius designation

This value is modal and can also be preset using GUD7, \_ZSFI[30]. The value set here can be overwritten by the NC program command.

**R:** Retraction amount ( $e$ )

This value is modal and can also be preset using GUD7, \_ZSFI[31]. The value set here can be overwritten by the NC program command.

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

**P:** Starting block of contour definition  
**Q:** Ending block of contour definition  
**U:** Finishing allowance in X direction ( $\Delta u$ ) (diameter/radius designation)  
**W:** Finishing allowance in Z direction ( $\Delta w$ )  
**F:** Machining feed  
**S:** Spindle speed  
**T:** Tool selection

F, S, or T functions issued within the NC program block range specified by address P and Q will be ignored. The relevant F, S, or T functions specified in the G71 block are effective.

### Note

- Both  $\Delta d$  and  $\Delta u$  are specified by means of the address U. If the addresses P and Q are present, then  $\Delta u$  is the case.
- Four cutting sectors are possible. The relevant signs of  $\Delta u$  and  $\Delta w$  vary as shown in the figure below:

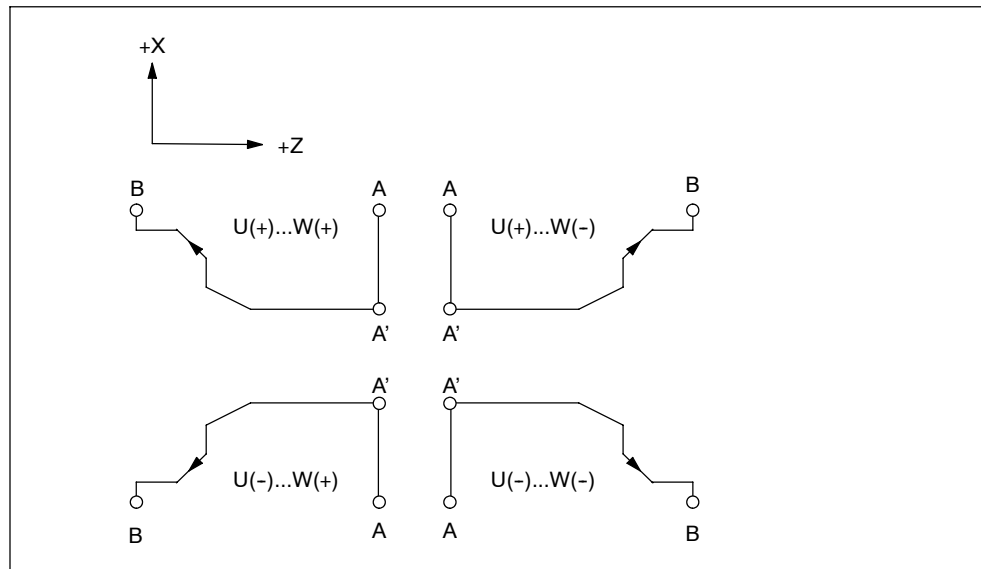


Fig. 4-16

Within the block specified by address P, the contour between points A and A' is determined (G00 or G01). A move command in the Z axis cannot be specified in this block.

The contour defined between A' and B must represent a steadily increasing or decreasing pattern in both X and Z axis.

- Within the range of NC blocks specified by address P and Q, subprograms cannot be called.

## 4.1 Program support functions (1)

**Type II**

In contrast to type I, type II does not necessarily have to show a steady increase or decrease along the X axis. In other words, it can also contain concaves (pockets).

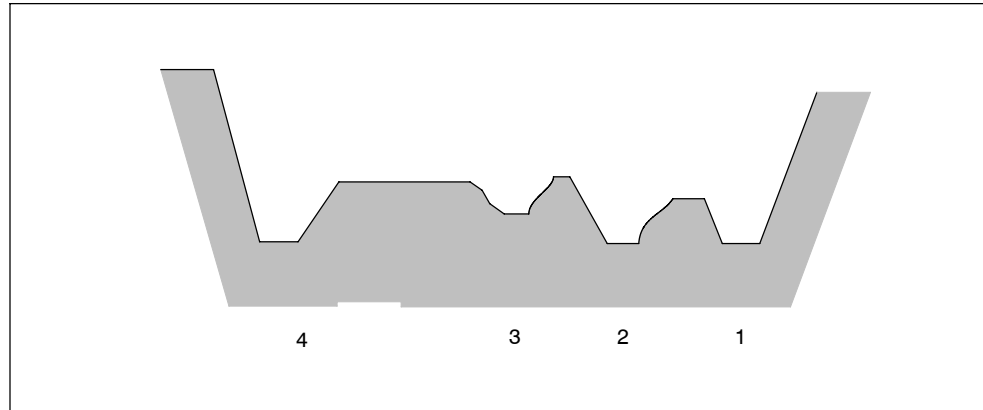


Fig. 4-17 Pockets in stock removal cycle (type II)

However, the Z axis profile must represent a monotone decrease or increase. For example, the following profile cannot be machined:

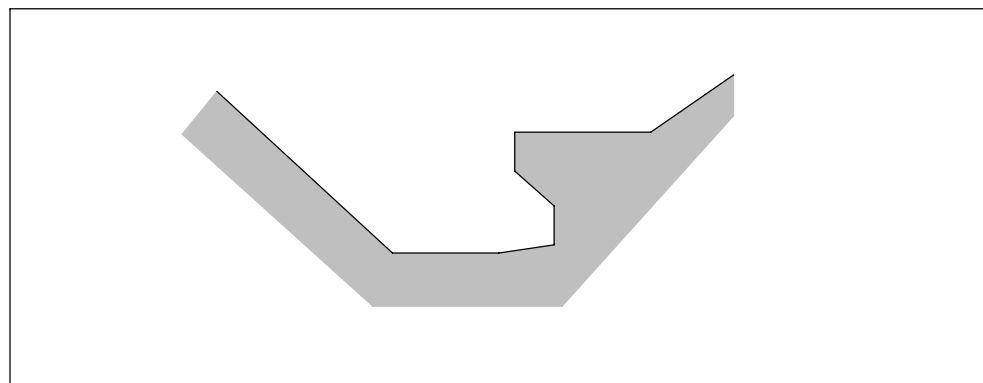


Fig. 4-18 Contour which cannot be machined in G71 cycle

**How to distinguish between type I and type II**

Type I: Only one axis is specified in the first block of the contour description

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

Whenever the first block does not contain a Z axis movement command and type II should be used, W0 has to be specified.



**Example**

Type I	Type II
G71 V10.0 R4.0 ;	G71 V10.0 R4.0 ;
G71 P50 Q100 .... ;	G71 P50 Q100 ..... ;
N50 X(U)... ;	N50 X(U)... Z(W)... ;
::	::
::	::
N100..... ;	N100..... ;

**Stock removal cycle, transverse axis (G72)**

With the G72 command, stock removal cycle and rough finishing cycle in which finishing allowance is left on face can be specified. In comparison to the cycle called by G71, which carries out cutting by the movement in parallel to the Z-axis, the G72 cycle carries out cutting by the movements parallel to the X-axis. Therefore, the cycle called by G72 executes the same operation as with the cycle called by G71 in a different direction.

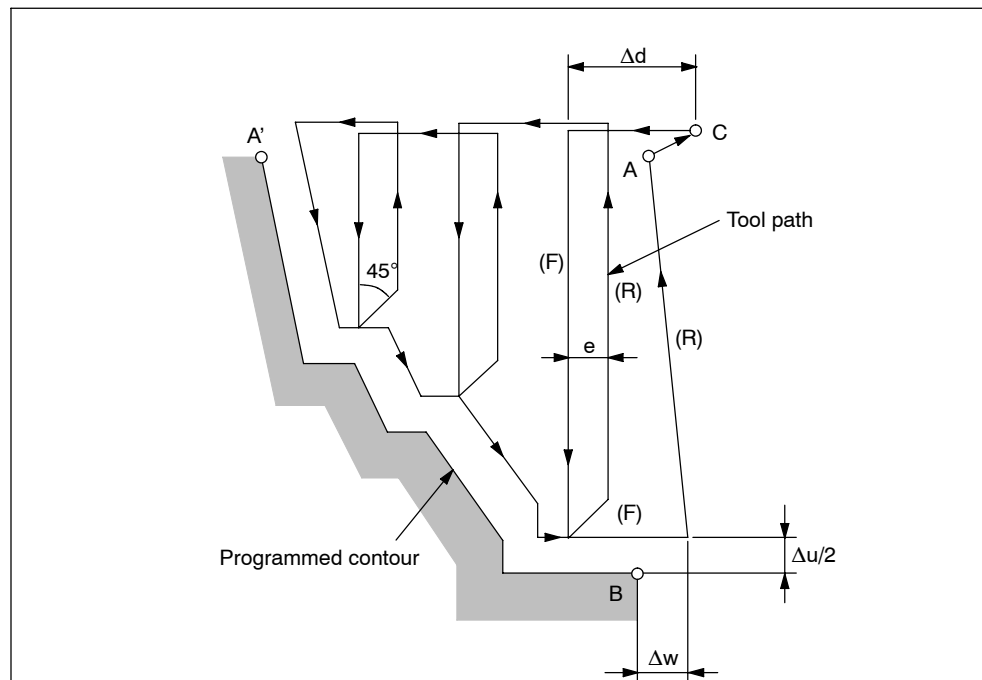


Fig. 4-19 Cutting path of a stock removal cycle, transverse axis



**G73 P... Q... U... W F... S... T... ;****P:** Starting block of contour definition**Q:** Ending block of contour definition**U:** Finishing allowance in X axis direction ( $\Delta u$ ) (diameter/radius designation)**W:** Finishing allowance in Z axis direction ( $\Delta w$ )**F:** Machining feed**S:** Spindle speed**T:** Tool selection

F, S, or T functions issued within the NC program block range specified by address P and Q will be ignored. The relevant F, S, or T functions specified in the G73 block are effective.

**Note**

1. The values  $\Delta i$  and  $\Delta k$ , or  $\Delta u$  and  $\Delta w$  are determined by address U and W respectively. However, their meanings are specified by the appearance of addresses P and Q present in the G73 block. Addresses U and W refer to  $\Delta i$  and  $\Delta k$  respectively whenever P and Q are not specified in the same block. Addresses U and W refer to  $\Delta u$  and  $\Delta w$  respectively whenever P and Q are specified in the same block.
2. Four cutting sectors are possible. The relevant signs of  $\Delta u$  and  $\Delta w$  vary as shown in the figure below:

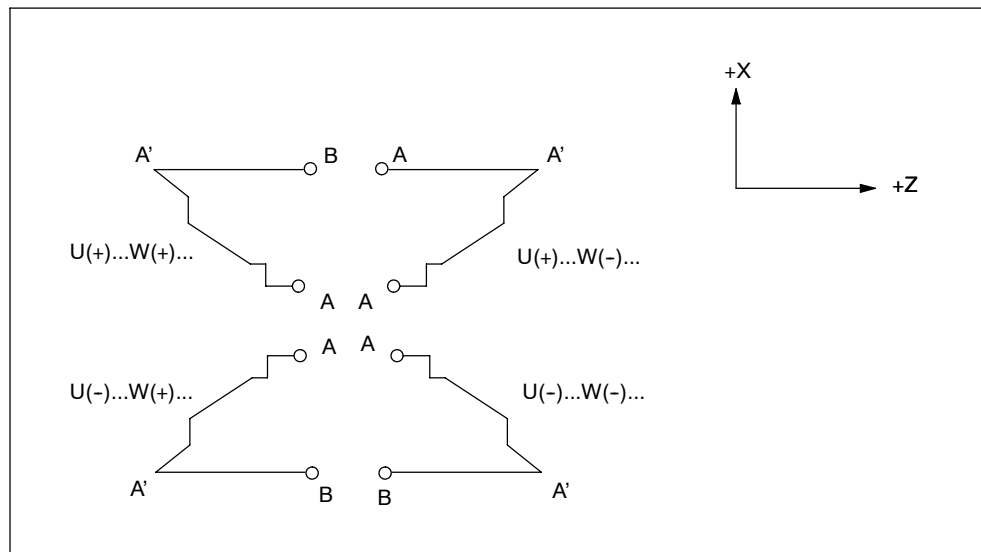


Fig. 4-21 Signs of numbers specified with U and W in stock removal in facing

The contour between A and A' is determined in the block specified by address P (G00 or G01). A move command in the X axis cannot be specified in this block. The contour between A' and B has to show a steadily increasing and decreasing pattern in both X and Z axes.

---

4.1 Program support functions (1)

3. Through the G73 command with P and Q specification, the cycle machining is performed. Four cutting sectors are considered here. Note the sign of  $\Delta u$ ,  $\Delta w$ ,  $\Delta k$ , and  $\Delta i$ . The tool returns to point A once the machining cycle has been completed.

**Finishing cycle (G70)**

While rough cutting is performed by G71, G72 or G73, the finishing is implemented through the following command.

**Format**

**G70 P... Q... ;**

**P:** Starting block of contour definition.

**Q:** Ending block of contour definition.

**Note**

1. The functions specified between the blocks determined by addresses P and Q are effective in G70 while those of F, S, and T are specified in the block G71, G72, G73 are not effective.
2. The tool is returned to the start point and the next block is read once the cycle machining through G70 has been completed.
3. Subprograms cannot be called within the blocks determined by the addresses P and Q.

## Examples

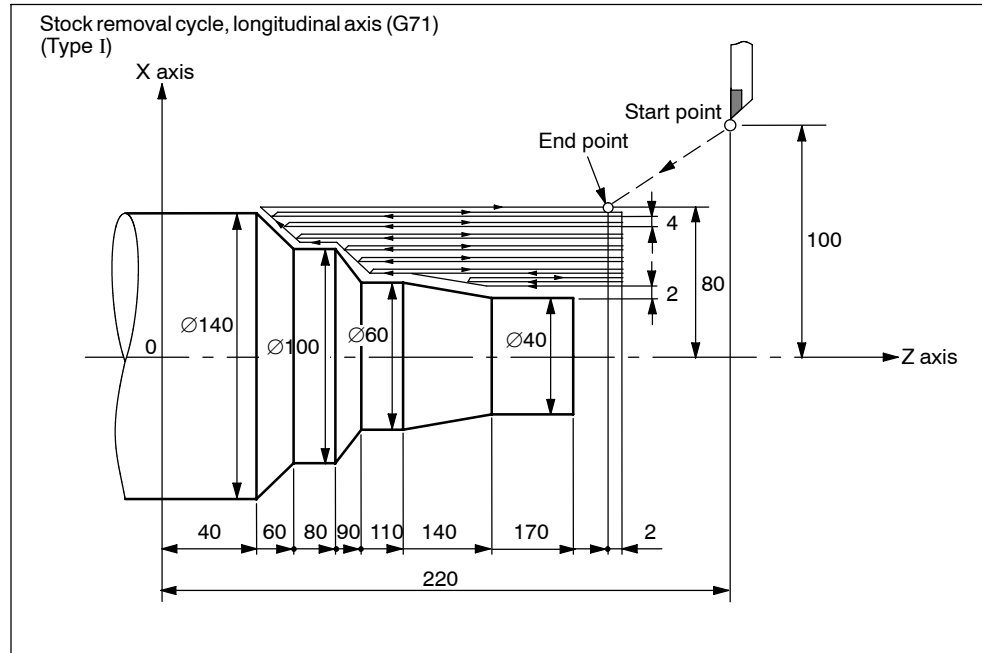


Fig. 4-22 Stock removal cycle, longitudinal axis

( Diameter designation, metric input )

```

N010 G00 G90 X200.0 Z220.0 ;
N011 X142.0 Z171.0 ;
N012 G71 U4.0 R1.0 ;
N013 G71 P014 Q020 U4.0 W2.0 F0.3 S550 ;
N014 G00 X40.0 F0.15 S700 ;
N015 G01 Z140.0 ;
N016 X60.0 Z110.0 ;
N017 Z90.0 ;
N018 X100.0 Z80.0 ;
N019 Z60.0 ;
N020 X140.0 Z40.0 ;
N021 G70 P014 Q020 ;
N022 G00 X200 Z220 ;

```

## 4.1 Program support functions (1)

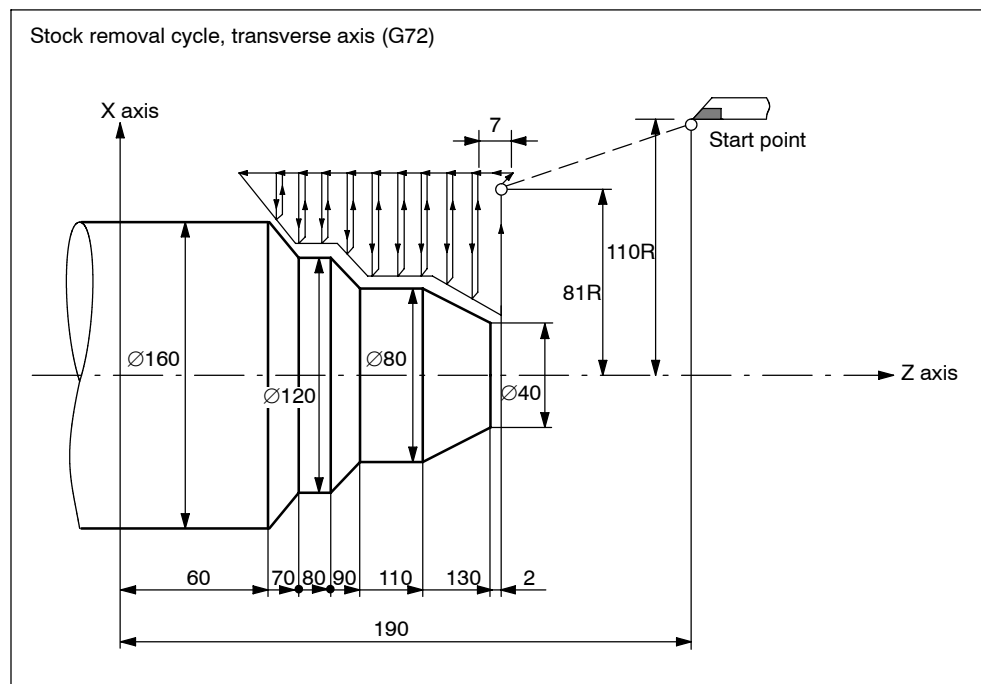


Fig. 4-23 Stock removal cycle, transverse

( Diameter designation, metric input )

```

N010 G00 G90 X220.0 Z190.0 ;
N011 G00 X162.0 Z132.0 ;
N012 G72 W7.0 R1.0 ;
N013 G72 P014 Q019 U4.0 W2.0 F0.3 ;
N014 G00 Z59.5 F0.15 S200;
N015 G01 X120.0 Z70.0 ;
N016 Z80.0 ;
N017 X80.0 Z90.0 ;
N018 Z110.0 ;
N019 X36.0 Z132.0 ;
N020 G70 P014 Q019 ;
N021 X220.0 Z190.0 ;

```

## 4.1 Program support functions (1)

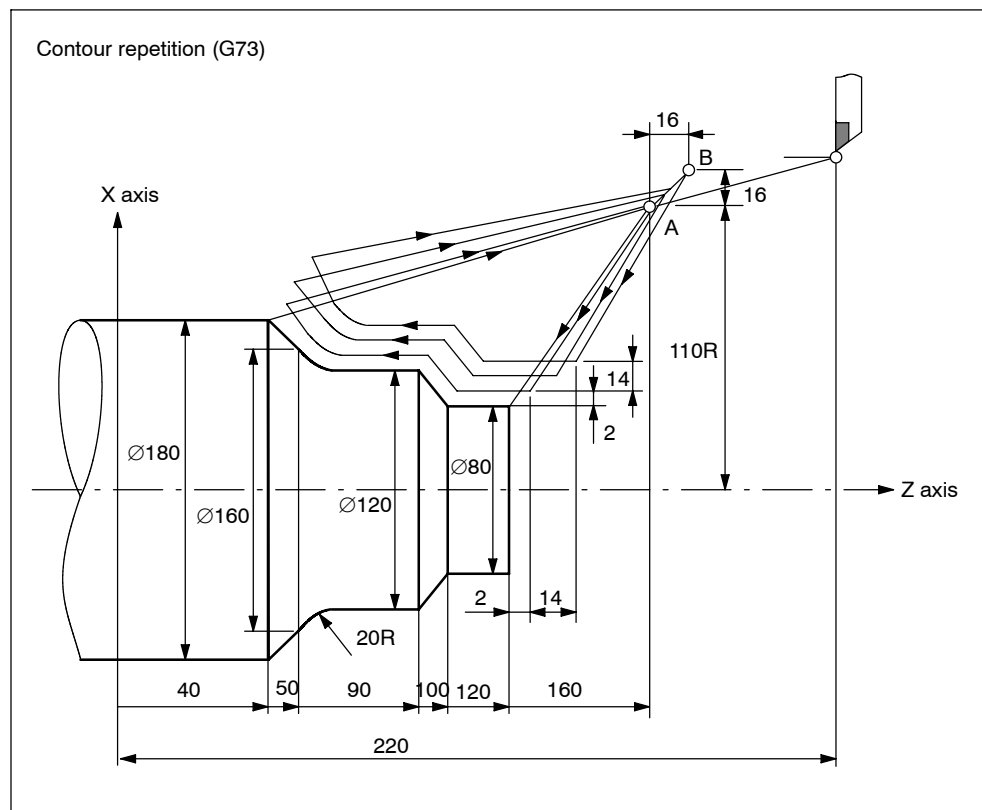


Fig. 4-24 Contour repetition

( Diameter designation, metric input )

```

N010 G00 G90 X260.0 Z220.0 ;
N011 G00 X220.0 Z160.0 ;
N012 G73 U14.0 W14.0 R3 ;
N013 G73 P014 Q020 U4.0 W2.0 F0.3 S0180 ;
N014 G00 X80.0 Z120.0 ;
N015 G01 Z100.0 F0.15 ;
N017 X120 Z90.0 ;
N018 Z70 ;
N019 G02 X160.0 Z50.0 R20.0 ;
N020 G01 X180.0 Z40.0 F0.25 ;
N021 G70 P014 Q020 ;
N022 G00 X260.0 Z220.0 ;

```

## 4.1 Program support functions (1)

**Deep hole drilling and recessing in longitudinal axis (G74)**

In the cycle called by G74, peck feed operation parallel to the Z-axis is repeated to carry out a face cut-off cycle.

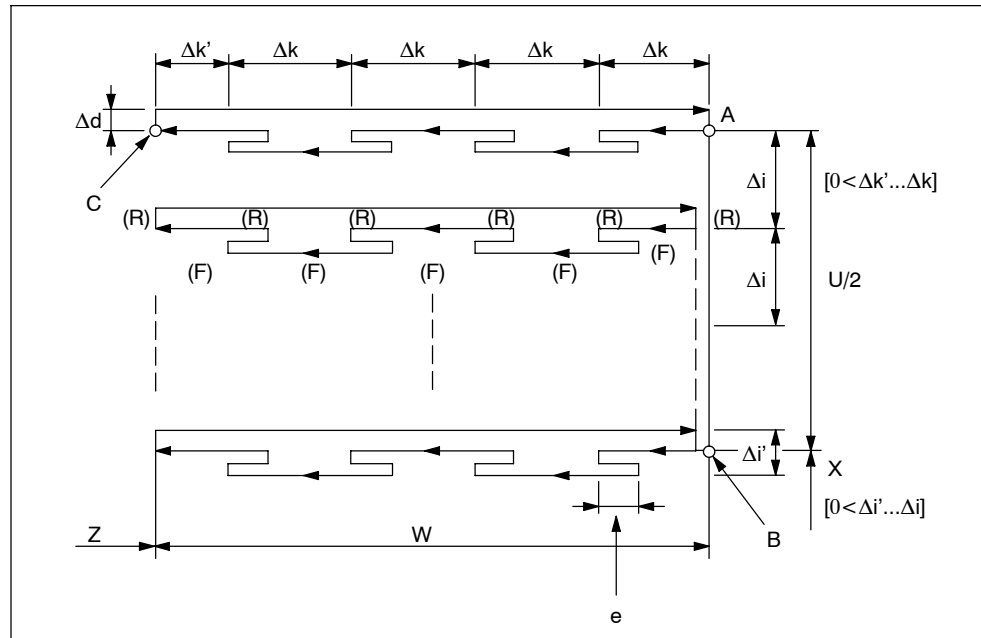


Fig. 4-25 Cutting path in deep hole drilling cycle

**Format**

**G74 R... ;**

**R:** Retraction amount (**e**)

This value is modal and can be preset using GUD7, ZSFI[29]. The value set here can be overwritten by the NC program command.

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

**X:** Starting point X (absolute position)

**U:** Starting point X (incremental)

**Z:** Starting point Z (absolute position)

**W:** Starting point Z (incremental)

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

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

**R:** Retraction amount (**Δd**) at recess base

**F:** Feed rate

**Note**

1. While both **e** and **Δd** are determined by address **R** their meanings are specified by the appearance of address **X (U)**. **Δd** is used when **X(U)** is specified.



- Through the G74 command with an X (U) specification, cycle machining is performed.

### Deep hole drilling and recessing in transverse axis (G75)

The G75 cycle executes an OD cut-off cycle while carrying out peck feed operation parallel to the X-axis. In comparison to the G74 cycle in which the OD cut-off cycle is executed in parallel to the X-axis, the G75 cycle executes virtually the same operation excluding that the cycle is executed in parallel to the X-axis.

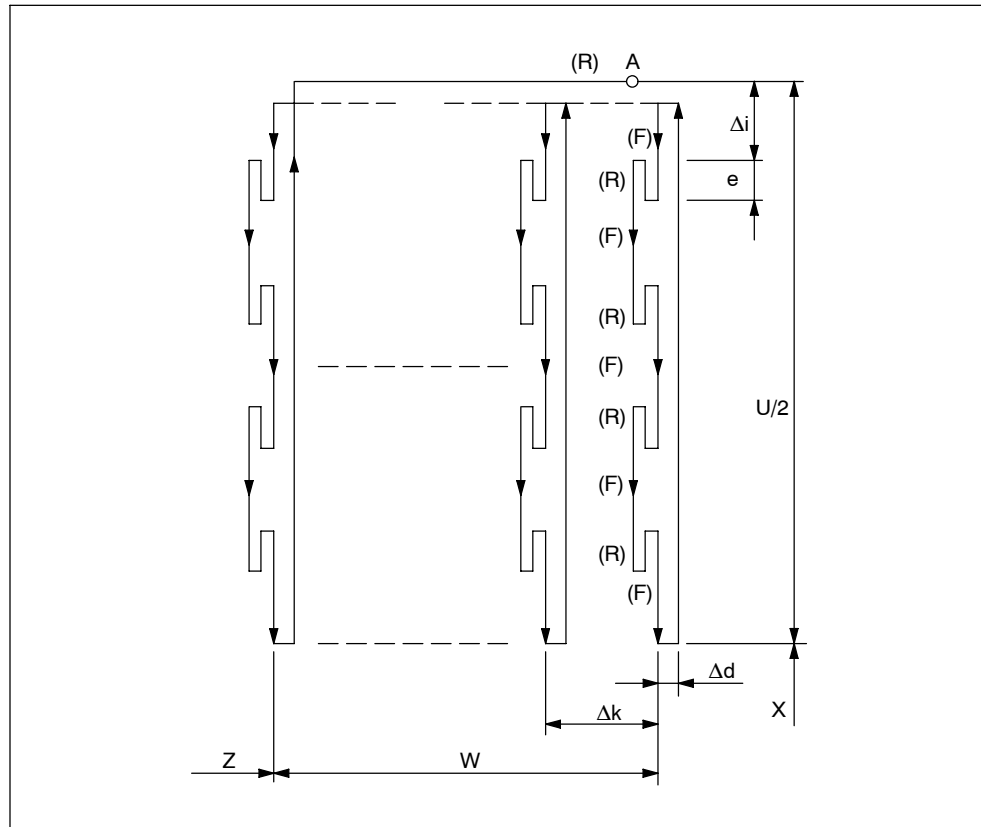


Fig. 4-26 Fig. 4-27 Cutting path in deep hole drilling and recessing in transverse axis (G75)

### Format

**G75 R... ;**

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

The meaning of the addresses are the same as those of G74 cycle.

Four cutting sectors are possible.

### Multiple thread cutting cycle (G76)

G76 calls an automatic thread cutting cycle for cutting straight or taper thread in which in-feed is carried out along a thread angle.

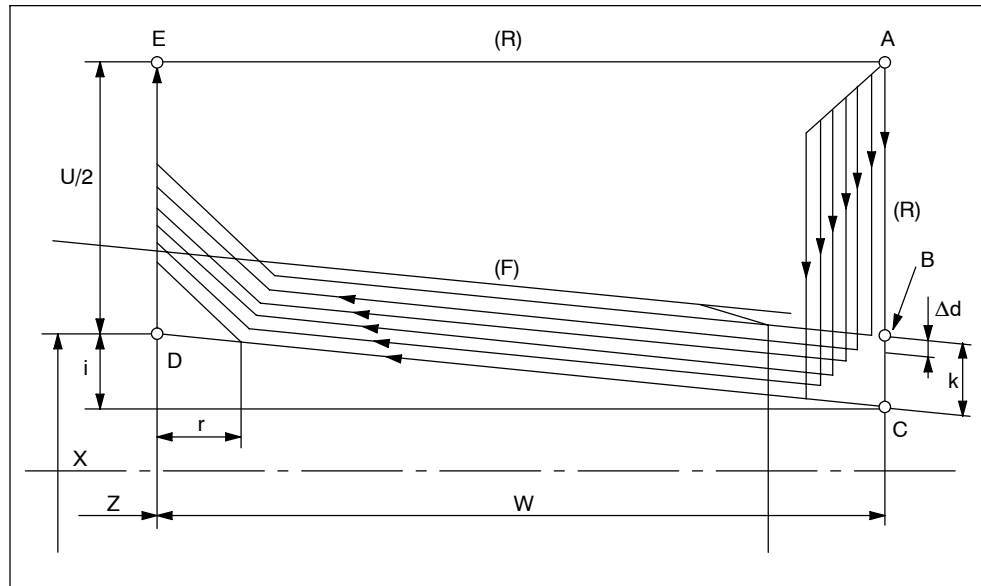


Fig. 4-28 Cutting path of a multiple thread cutting cycle

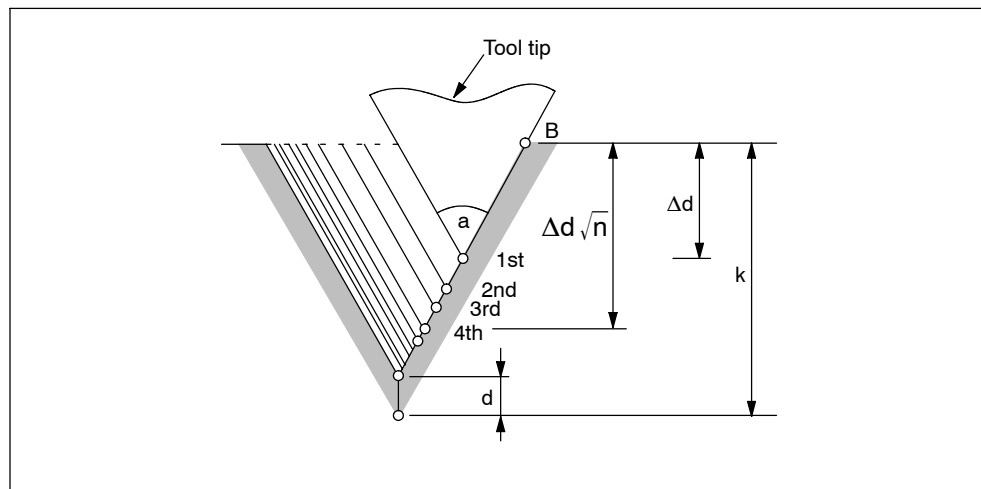


Fig. 4-29 In-feed in thread cutting

## Format

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

**P:**

**m:** Number of finishing cuts

This value is modal and can also be preset using GUD7, ZSFI[24]. The value set here can be overwritten by the NC program command.

**r:** Size of chamfer at the end of the thread ( $1/10 \times$  thread pitch)

This value is modal and can also be preset using GUD7, ZSFI[26]. The value set here can be overwritten by the NC program command.

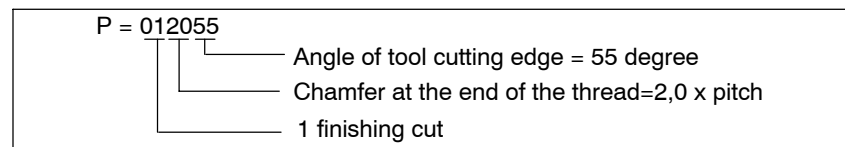
**a:** Angle of tool cutting edge

This value is modal and can also be preset using GUD7, ZSFI[25]. The value set here can be overwritten by the NC program command.

All the above parameters are specified by address P at the same time.

Example for address P:

G76 P012055 Q4 R0.5



**Q:** Minimum infeed depth ( $\Delta d_{min}$ ), radius value

The cutting depth is clamped at the value specified at address Q whenever the cutting depth of one cycle operation ( $\Delta d - \Delta d - 1$ ) becomes less than this limit. This value is modal and can also be preset using GUD7, ZSFI[27]. The value set here can be overwritten by the NC program command.

**R:** Finishing allowance (**d**)

This value is modal and can also be preset using GUD7, ZSFI[28]. The value set here can be overwritten by the NC program command.

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

**X, U:** Endpoint of thread in X axis direction (absolute position (X), incremental (U))

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

**R:** Radius difference for tapered thread (**i**).  $i = 0$  for ordinary straight thread

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

**Q:** Infeed amount for the 1st cut ( $\Delta d$ ), radius value

**F:** Lead (**L**)

## 4.1 Program support functions (1)

**Note**

1. The appearance of X (U) and X (W) determine the meaning of the data specified by address P, Q, and R.
2. Through the G76 command with X (U) and Z (W) specification cycle machining is performed. One edge cutting is performed and the load on the tool tip is reduced when this cycle is applied.

The amount of cutting per cycle is kept constant by assigning the cutting depth  $\Delta d$  to the first path, and  $\Delta d_n$  to the nth path. Corresponding to the sign of each address, four symmetrical sectors are considered here.

3. The notes on thread cutting are equivalent to those on G32 for thread cutting and G92 for the thread cutting cycle.

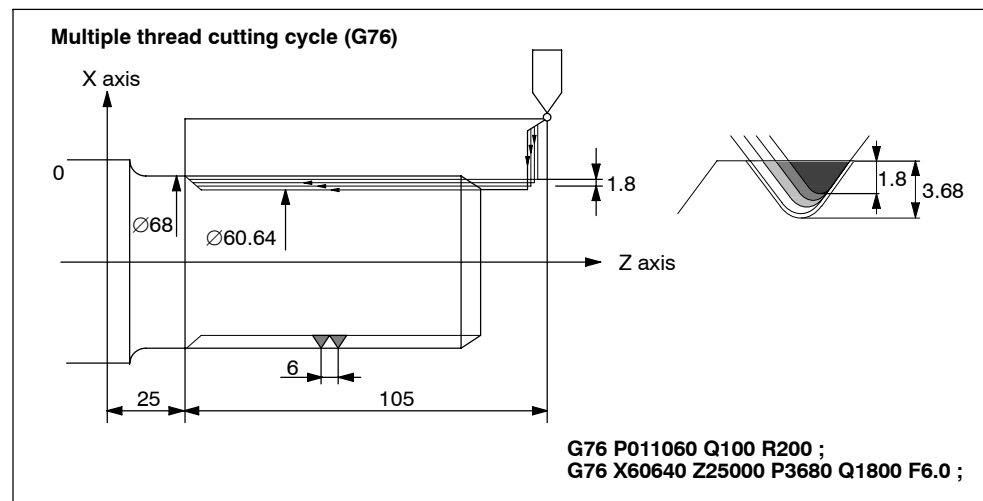
**Examples**

Fig. 4-30 Multiple thread cutting cycle (G76)

**Notes on multiple repetitive cycle (G70–G76)**

1. G70, G71, G72, or G73 cannot be commanded in MDA mode. If it is commanded, alarm 14011 is generated. However, G74, G75, and G76 can be commanded in MDA mode.
2. M98 (subprogram call) and M99 (subprogram end) cannot be commanded in the blocks containing G70, G71, G72, or G73 and between the sequence numbers specified by addresses P and Q.
3. The following commands cannot be specified in the blocks between the sequence numbers specified by addresses P and Q:
  - One shot G codes with the exception of G04 (dwell)
  - 01 group G codes with the exception of G00, G01, G02, and G03
  - 06 group G codes
  - M98 / M99
4. Do not program in such a way that the final movement command of the contour definition for G70, G71, G72, and G73 finishes off with chamfering or corner rounding. An alarm is issued whenever the above is specified.
5. In the G74, G75, and G76 cycles, addresses P and Q use the least input increments to specify the amount of travel and depth of cut.
6. No tool nose radius compensation can be carried out within G71, G72, G73, G74, G75, G76, or G78 cycles.

## 4.1 Program support functions (1)

## 4.1.3 Hole-machining canned cycles (G80 to G89)

Hole-machining canned cycles (G80 to G89) can define specific movements for machining holes that usually require several blocks of commands by single-block commands. G80 cancels the called out canned cycle program.

G codes that call out canned cycles G80 to G89 are common for all G code systems.

**G codes calling canned cycles and axis movement patterns of canned cycles**

G codes that call out a canned cycle and the axis movement pattern of the called canned cycle are indicated in Table 4-4.

Table 4-4 Hole-machining canned cycles

G code	Hole machining operation (direction)	Processing at bottom hole	Retraction (+ direction)	Applications
G80	-	-	-	Cancel
G83	Cutting feed/intermittent	Dwell	Rapid traverse	Front drilling cycle
G84	Cutting feed	Dwell -> spindle CCW	Cutting feed	Front tapping cycle
G85	Cutting feed	Dwell	Cutting feed	Front boring cycle
G87	Cutting feed/intermittent	Dwell	Rapid traverse	Side drilling cycle
G88	Cutting feed	Dwell -> spindle CCW	Cutting feed	Side tapping cycle
G89	Cutting feed	Dwell	Cutting feed	Side boring cycle
G88	Cutting feed	Spindle forward rotation after dwell	Manual return → Spindle forward rotation	Boring
G89	Cutting feed	Dwell	Cutting feed	Boring

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

- Operation 1 - Positioning of X (Z) and C axis
- Operation 2 - Rapid traverse movement to level R
- Operation 3 - Hole machining
- Operation 4 - Operation at hole bottom
- Operation 5 - Retraction to R level
- Operation 6 - Rapid retraction to the initial point

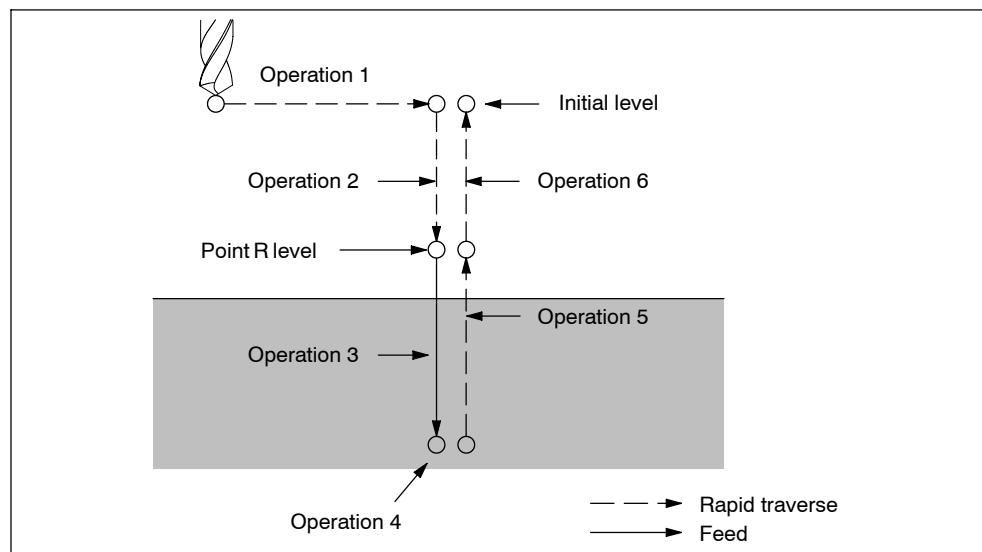


Fig. 4-31 Sequence of drilling cycle operation

## Explanations

### Positioning axis and drilling axis

As shown below, a drilling G code determines the positioning axes as well as the drilling axis. The C-axis and X or Z-axis correspond to the positioning axes. The drilling axis is represented by the X or Z-axis: These axes are not used as positioning axes.

Table 4-5 Positioning plane and its respective drilling axis

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

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

### Drilling mode

The G codes (G83–G85 / G87–89) are modal, and remain active until they are canceled. The current state is the drilling mode whenever they are active. The data is retained until modified or canceled once drilling data is specified in the drilling mode.

All necessary drilling data have to be specified at the beginning of the canned cycles. Only data modifications are allowed to be specified while canned cycles are being carried out.

## 4.1 Program support functions (1)

**Return point level (G98/G99)**

When the G code system A is active, the tool traverse away from the bottom of a hole and goes back to the initial level. When specifying G98 while the G code system B or C is active, the tool, coming from the bottom of a hole, returns to the initial level. When specifying G99, the tool returns to the R level from the bottom of a hole.

The figure below describes the movement of the tool when G98 or G99 is specified. G99 is generally applied for the first drilling operation, while G98 is applied for the last drilling operation. Even when drilling is performed in the G99 mode, the initial level does not change.

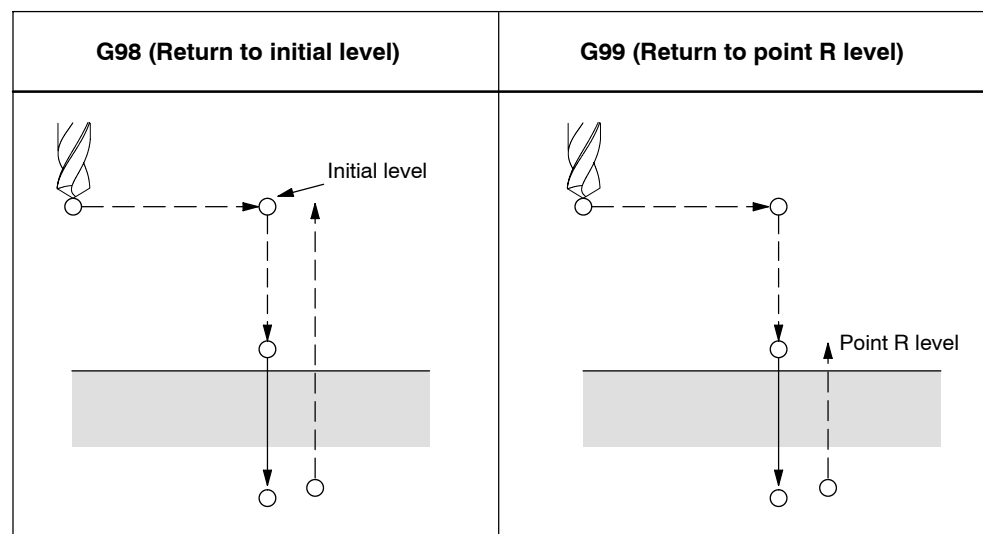


Fig. 4-32 Return point level (G98/G99)

**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).

Drilling data is stored, but drilling is not performed whenever K0 is specified.

**Cancel**

Use G80 or a group 01 G code (G00, G01, G02, G03) to cancel a canned cycle.



## Symbols and figures

The individual canned cycle are explained in the following sections. The following symbols are used in the figures below:




	Positioning (rapid traverse G00)
	Cutting feed (linear interpolation G01)
	Manual feed
P1	Dwell time
M $\alpha$	M code for C-axis clamp
M ( $\alpha+1$ )	M code for C-axis unclamp

Fig. 4-33



### Caution

In each canned cycle, the address R (distance between initial level and point R) is always treated as a radius.

However, Z or X (distance between point R and hole bottom) is treated either as a diameter or radius, depending on the specification.

## Face deep hole drilling cycle (G83) / side deep hole drilling cycle (G87)

The setting of GUD7, \_ZSFI[20] decides whether The deep hole drilling cycle or high-speed deep hole drilling cycle is applied. The normal drilling cycle is applied whenever depth of cut for each drilling is not specified.

## 4.1 Program support functions (1)

**High-speed deep hole drilling cycle (G83, G87)  
(GUD7, \_ZSFI[20]=0)**

When using high-speed deep hole drilling cycle, the drill repeats the cycle of drilling at the cutting feedrate. It intermittently retracts by a specified distance until the tool reaches the bottom of the hole.

**Format**

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

or

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

**X, C or Z, C:** Hole position

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

**R\_:** Distance from the initial level to R level

**Q\_:** Infeed

**P\_:** Dwell time at bottom of hole

**F\_:** Cutting feedrate

**K\_:** Number of repetitions (if required)

**M\_:** M code for clamping C-axis (if required)

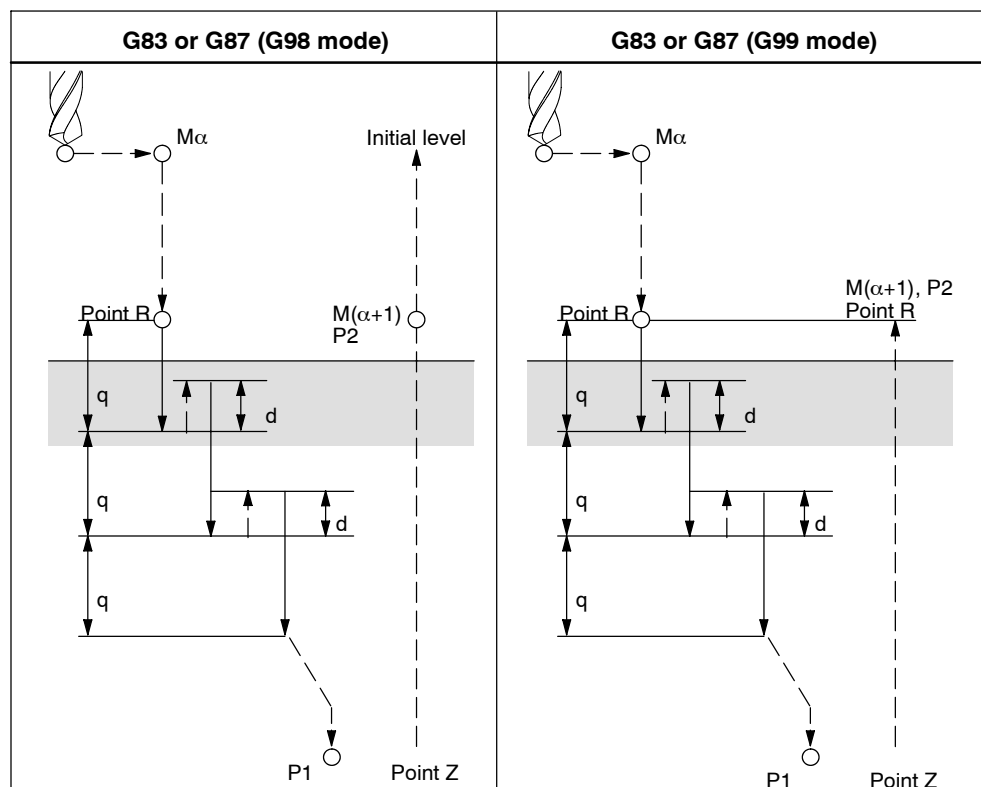


Fig. 4-34 High-speed deep hole drilling cycle

<b>M<math>\alpha</math>:</b>	M code for clamping C-axis
<b>M(<math>\alpha+1</math>):</b>	M code for unclamping C-axis
<b>P1:</b>	Dwell time (program)
<b>P2:</b>	Dwell time specified in GUD7, _ZSFR[22]
<b>d:</b>	Retraction amount specified in GUD7, _ZSFR[21]

### Deep hole drilling cycle (G83, G87) (GUD7, \_ZSFI[20]=1)

#### Format

G83 X(U)... C(H)... Z(W)... R... Q... P... F... M... K... ;  
or  
G87 Z(W)... C(H)... X(U)... R... Q... P... F... M... K... ;

**X, C or Z, C:** Hole position

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

**R\_:** Distance from the initial level to R level

**Q\_:** Infeed

**P\_:** Dwell time at bottom of hole

**F\_:** Cutting feedrate

**K\_:** Number of repetitions (if required)

**M\_:** M code for clamping C-axis (if required)

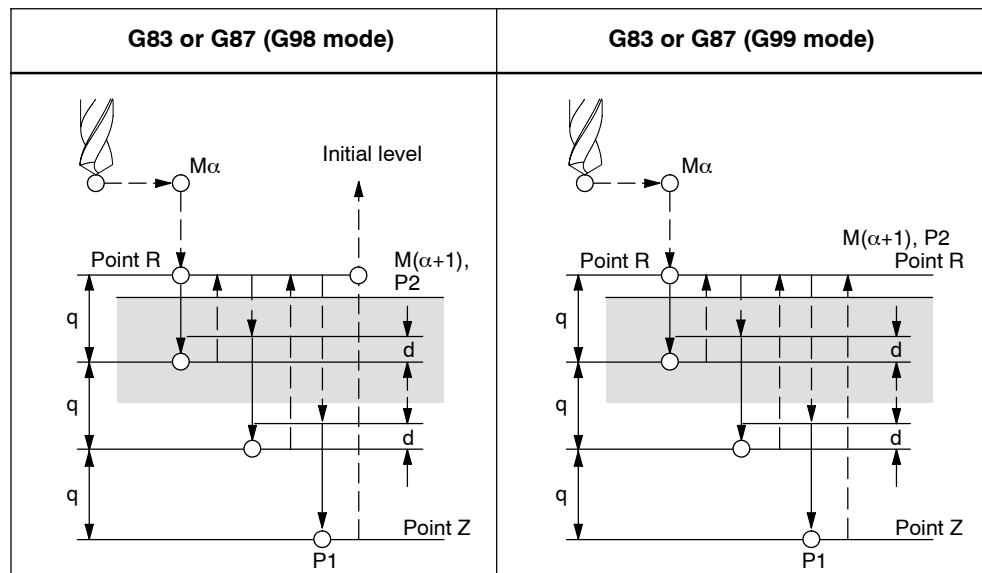


Fig. 4-35 Deep hole drilling cycle

## 4.1 Program support functions (1)

<b>M<math>\alpha</math>:</b>	M code for clamping C-axis
<b>M(<math>\alpha+1</math>):</b>	M code for unclamping C-axis
<b>P1:</b>	Dwell time (program)
<b>P2:</b>	Dwell time specified in GUD7, _ZSFR[22]
<b>d:</b>	Retraction amount specified in GUD7, _ZSFR[21]

**Example**

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

**Drilling cycle (G83 or G87)**

The normal drilling cycle is applied whenever the depth of cut for each drilling is not specified. In this case, the tool is retracted from the bottom of the hole in rapid traverse.

**Format**

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

or

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

**X, C or Z, C:** Hole position

**Z or X:** The distance from point R to the bottom of the hole

**R\_:** Distance from the initial level to R level

**P\_:** Dwell time at bottom of hole

**F\_:** Cutting feedrate

**K\_:** Number of repetitions (if required)

**M\_:** M code for clamping C-axis (if required)

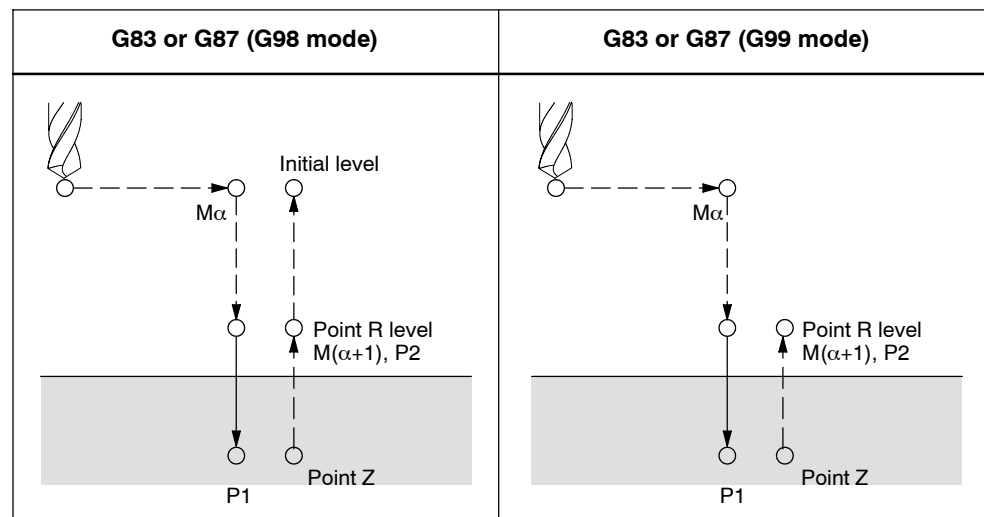


Fig. 4-36

**M $\alpha$ :** M code for clamping C-axis  
**M( $\alpha$ +1):** M code for unclamping C-axis  
**P1:** Dwell time (program)  
**P2:** Dwell time specified in GUD7, \_ZSFR[22]

### Example

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

### Notice

If \_ZSFR[10]

- > 0 = value is used for anticipation distance "d" (distance minimal 0.001)
- = 0 The anticipation distance d is calculated internally in the cycles as follows:
  - If the drilling depth 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 Program support functions (1)

**Face tapping cycle (G84), Side tapping cycle (G88)**

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

**Format**

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

or

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

**X, C or Z, C:** Hole position

**Z or X:** The distance from point R to the bottom of the hole

**R\_:** Distance from the initial level to R level

**P\_:** Dwell time at bottom of hole

**F\_:** Cutting feedrate

**K\_:** Number of repetitions (if required)

**M\_:** M code for clamping C-axis (if required)

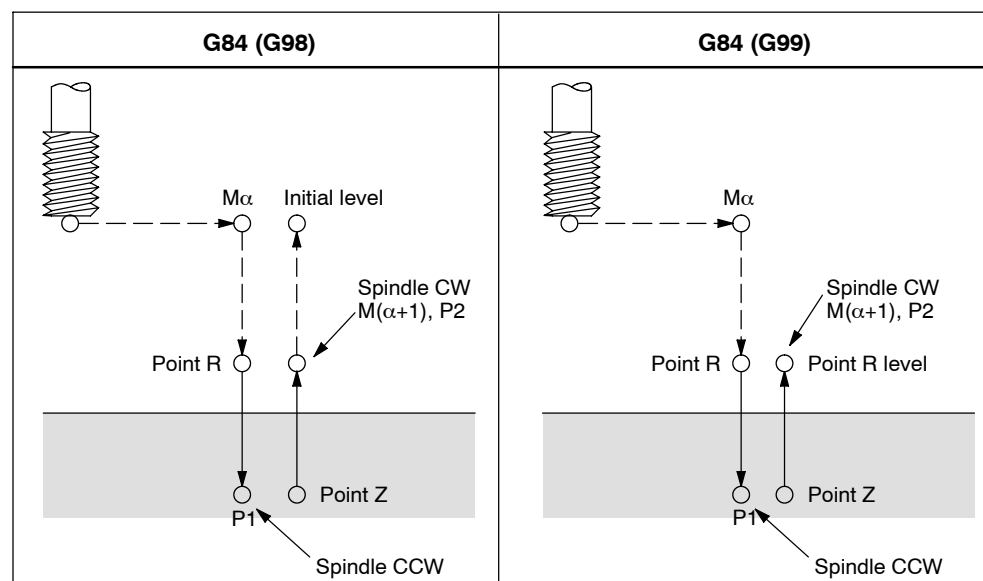


Fig. 4-37

P2: Dwell specified in GUD7, \_ZSFR[22]

**Explanations**

In tapping operation, the spindle is rotated clockwise towards the bottom of the hole and reversed for retraction. The cycle is not stopped until the return operation is completed.

**Example**

M3 S2500 ;	Rotate the tapping tool
G00 X50.0 C0.0 ;	Positioning X and C axis
G84 Z-35.0 R-5.0 P500 F5.0 ;	Tapping hole 1
C90.0 ;	Tapping hole 2
C180.0 ;	Tapping hole 3
C270.0 ;	Tapping hole 4
G80 M05 ;	Cycle cancel and tapping tool stop

**Face drilling cycle (G85)****Side drilling cycle (G89)****Format**

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

or

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

**X, C or Z, C:** Hole position

**Z or X:** The distance from point R to the bottom of the hole

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

**P:** Dwell time at bottom of hole

**F:** Cutting feedrate

**K:** Number of repetitions (if required)

**M:** M code for clamping C-axis (if required)

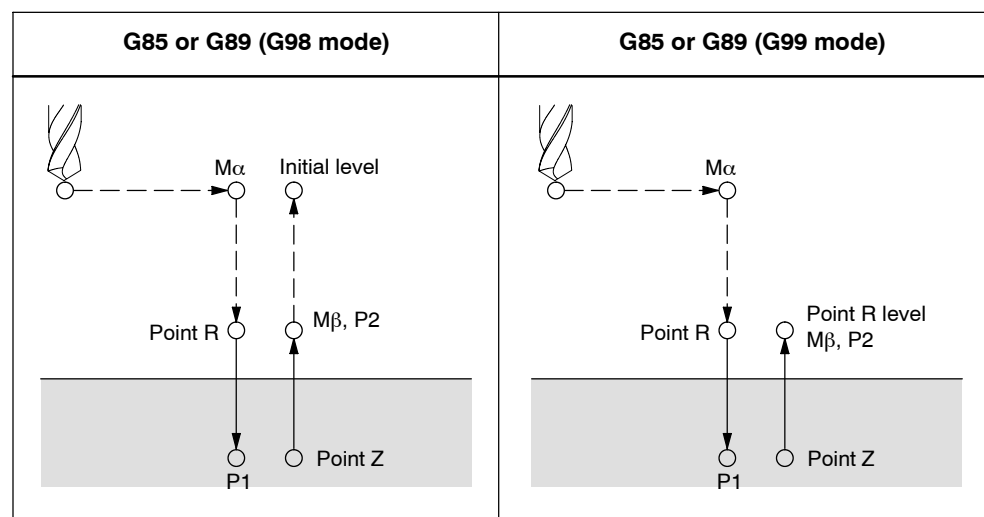


Fig. 4-38

P2: Dwell specified in GUD7, \_ZSFR[22]

---

4.1 Program support functions (1)**Explanations**

Rapid traverse is performed to point R after positioning at the hole position. Drilling is then carried out from point R to point Z and subsequently returned to point R.

**Example**

M3 S2500 ;	Rotate the drilling tool
G00 X50.0 C0.0 ;	Positioning X and C-axis
G85 Z-40.0 R-5.0 P500 F5.0 M31 ;	Machining hole 1
C90.0 M31 ;	Machining hole 2
C180.0 M31 ;	Machining hole 3
C270.0 M31 ;	Machining hole 4
G80 M05 ;	Cycle canceling and drilling tool stop

**Canned cycle for drilling cancel (G80)**

G80 cancels canned cycle.

**Format**

G80 ;

**Explanations**

Canned cycle for drilling is canceled and normal operation is continued.



## 4.2 Program support functions (2)

### 4.2.1 Changing of tool offset value Programmable data input (G10)

By using the commands of “G10 P... X(U)... Y(V)... Z(W)... R(C)... Q ;”, it is possible to write and update the tool offset amount using a part program. If an address is omitted in the designation of data input block, the offset amounts for the omitted addresses remains unchanged.

Table 4-6 Description of addresses

Address	Description
P	Offset number (see explanation below)
X	Offset value on X axis (absolute, incremental)
Y	Offset value on Y axis (absolute, incremental)
Z	Offset value on Z axis (absolute, incremental)
U	Offset value on X axis (incremental)
V	Offset value on Y axis (incremental)
W	Offset value on Z axis (incremental)
R	Tool nose radius offset value (absolute)
C	Tool nose radius offset value (incremental)
Q	Imaginary tool nose number

#### Address P

Address P specifies the tool offset number and, at the same time, whether tool geometry offset or tool wear offset is to be changed. The value to be specified with address P depends on the setting of MD \$MC\_EXTERN\_FUNCTION\_MASK, Bit1 as follows:

\$MC\_EXTERN\_FUNCTION\_MASK, Bit1 = 0

P1 to P99: Writing tool wear offset

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

\$MC\_EXTERN\_FUNCTION\_MASK, Bit1 = 1

P1 to P9999: Writing tool wear offset

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

## 4.2 Program support functions (2)

**Note**

Use of this command in a program allows the tool to advance little by little. This command can also be used to input offset values one at a time from a program by specifying this command successively instead of inputting these values one at a time from the MDI unit.

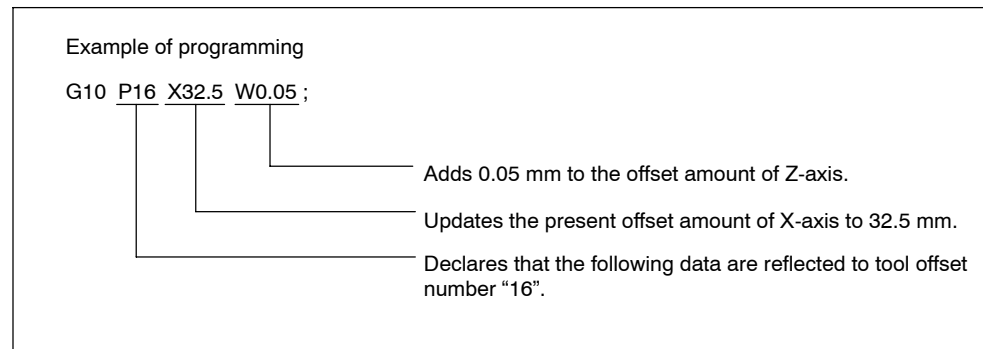


Fig. 4-39

**Setting the workpiece coordinate system shift data**

With the commands of "G10 P00 X (U) ··· Z (W) ··· C (H) ··· ;", 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             |

**4.2.2 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 4-7 are used.

Table 4-7 Subprogram call M code

M code	Function
M98	Subprogram call up
M99	End of subprogram

### Subprogram call (M98)

- M98 P xxxx yyyy  
y: Program number (max. 4 digits)  
x: Number of repetitions (max. 4 digits)
- The program syntax M98 Pxxxxyyyy is used to call a subprogram with the number yyyy and repeat it xxxx times. If the xxxx is not programmed, the sub-program is executed only once. The subprogram name always consists of 4 digits or is extended to 4 digits by adding 0's.  
For example, if M98 P21 is programmed, the part program memory is searched for program name 0021.spf and the subprogram is executed once. To execute the subprogram 3 times, program M98 P30021.
- As an alternative, the number of subprogram executions can also be programmed at address 'L'. The number of the subprogram is still programmed as Pxxxx. If the number of executions is programmed at both addresses, the number of executions programmed at address 'L' is valid. A valid range for address 'L' is 1 to 9999.
- Nesting of subprograms is possible - the allowable nesting level is four. If the nesting level exceeds this limit, an alarm occurs.

### Example:

N20 M98 P20123 ;	Subprogram 1023.spf will be executed twice
N40 M98 P55 L4 ;	Subprogram 0055.spf will be executed four times
N60 M98 P30077 L2 ;	Subprogram 0077.spf will be executed twice
	The number of executions programmed at address 'P' = 3 is ignored

## 4.2 Program support functions (2)

### End of subprogram code (M99)

M99 terminates the subprogram.

If M99 Pxxxx is programmed, execution resumes at block number Nxxxx on the return jump to the main program. The block number must always begin with 'N'. The system initially searches forwards 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 scanned backwards (towards the start of the program). If M99 appears without a block number (Pxxxx) in a subprogram, the subprogram is terminated and the processor jumps back to the main program to the block following the subprogram call.

If M99 appears without a block number (Pxxxx) in a main program, it is jumped back back to the head of the main program and the program is executed again. These M commands are not output to the PLC.

### Subprogram return jump with 'RET'

In the Siemens shell cycles for stock removal (as in ISO Dialect), it is necessary after roughing to resume program execution in the main program after the contour definition. To achieve this, the shell cycle must contain a subprogram return jump to the block after the end of the contour definition. The RET command has been extended with two optional parameters for skipping the blocks with the contour definition in the stock removal cycles after the subprogram call (with G71–G73).

The command RET (STRING: <sequence no./label>) is used to resume program execution in the calling program (main program) at the block with <sequence no./label>.

If program execution is to be resumed at the next block after <sequence no./label>, the 2nd parameter in the RET command must be > 0; RET (<sequence no./label>, 1). If a value > 1 is programmed for the 2nd parameter, the subprogram also jumps back to the block after the block with <sequence no./label>.

In G70–G73 cycles, the contour to be machined is stored in the main program. The extended RET command is required in order to resume execution after the contour definition in the main program at the end of G70 (finish cut via contour with stock removal cycle). To jump to the next NC block after the contour definition at the end of the shell cycle for G70, the shell cycle must be terminated with the following return syntax:

RET ('N' << \$C\_Q, 1)

Search direction:

The search direction for <sequence no./label> is always forwards first (towards the end of the program) and then backwards (towards the head of program).

**Example**

```
N10 X10. Y20.
N20 G71 P30 Q60 U1 W1 F1000 S1500
      N10 ... ;Shell cycle for stock removal cycle
      N20 DEF STRING[6]BACK
      N30 ...
      N90
      N100 RET ('N'<<$C_Q, 1) ;Return jump to block after
                                ;Contour def. -> N70

N30 X50. Z20.
N40 X60.
N50 Z55.
N60 X100. Z70.
N70 G70 P30 Q60
N80 G0 X150. Z200.
N90 M30
```

---

**Notice**

M30 in Siemens mode: is interpreted as a return jump in a subprogram.  
M30 in ISO Dialect mode: is also interpreted as the end of the part program in a subprogram.

---

## 4.3 Eight-digit program number

## 4.3 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 98**

\$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 Pxxxxxxx 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.

**4.4 Automating support functions****4.4.1 Skip function (G31)**

By specifying "G31 X(U)... Z(W)... 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.

The skip function is used when the end of machining is not programmed but specified with a signal from the machine. It is used also for measuring the dimensions of a workpiece. For details of how to use this function, refer to the manual supplied by the machine tool builder.

**Format**

G31 X... Z... F\_;

G31: One-shot G code (It is effective only in the block in which it is specified)

**If skip signal is turned ON**

When the skip signal is input, the coordinate values of the point where the skip signal is input are automatically saved to the parameters. Therefore, the coordinate values of the skip point can be used as the coordinate data in macro programs.

\$AA_IM[X]	Saving the X-axis coordinate value
\$AA_IM[Z]	Saving the Z-axis coordinate value

#### 4.4 Automating support functions

##### If skip signal is not turned ON

If the skip signal is not turned ON during the execution of the commands specified in the G31 block, the operation stops upon completion of these commands and an alarm occurs. Note that G31 is a non-modal G code.

If G31 is issued while the skip signal input is ON, alarm 21700 is issued.

##### Operation after skip signal ON

How the axes move after the turning ON of the skip signal varies depending on the commands specified in the block to be executed next.

##### When axis move commands in the next block are incremental commands

The position where the skip signal is turned ON is taken as the reference point to execute the incremental commands in the next block.

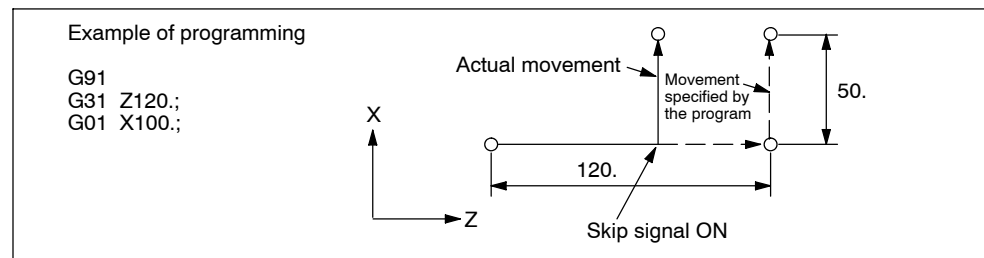


Fig. 4-40 Example of programming



### When axis move command in the next block is absolute command (one axis)

The axis specified in the next block moves to the specified position and the other axis remains at the position where the skip signal has turned ON.

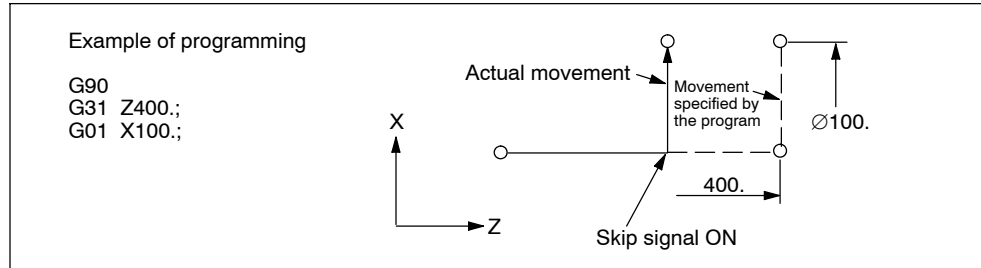


Fig. 4-41 Example of programming

### When axis move commands in the next block are absolute command (two axes)

The axes move to the specified position when the skip signal is turned ON.

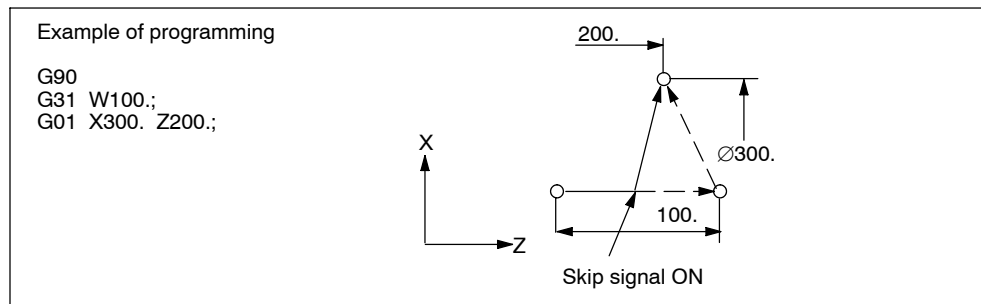


Fig. 4-42 Example of programming

#### Notice

Before specifying G31, cancel the nose R offset mode by specifying G40. If G31 is specified without canceling the nose R offset mode, an alarm occurs.

---

4.4 Automating support functions**4.4.2 Multistage skip (G31, P1-P2)**

The multistage skip function stores coordinates in a macro variable within a block specifying P1 to P2 after G31 whenever a skip signal is turned on. In order to match multiple Pn (n=1,2) 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... Z... F... P ... ;

X, Z: End point

F: Feedrate

P: P1-P2

**Explanation**

Multistage skip is activated by specifying P1 or P2 in a G31 block. The digital inputs are assigned to addresses P1 and P2 through machine data as follows:

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

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

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

## 4.5 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.

### 4.5.1 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.

### 4.5.2 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 4-1.

Table 4-8 Macroprogram calling format

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

## 4.5 Macroprograms

### 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 4-9 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.

---

**Notice**

The transfer parameters can only be read in the subroutine.

---

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.

Example:

```
N5 I10 J10 K30 J22 K55 I44 K33
      set1      set2      set3
$C_I[0]=10
$C_I[1]=44
$C_I_ORDER[0]=1
$C_I_ORDER[1]=3

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

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

#### 4.5 Macroprograms

### 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 4-10 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 4-11 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
H	\$C_H	W	\$C_W
I	\$C_I[0]	X	\$C_X
J	\$C_J[0]	Y	\$C_Y

## 4.5 Macroprograms

Table 4-11 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
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. Suffixes 1 to 10 specified in the table below indicate the order they are used in a set, and the suffix is not written in actual commands.

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 4-12 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]
I5	\$C_I[4]	K10	\$C_K[9]



Table 4-12 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
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.

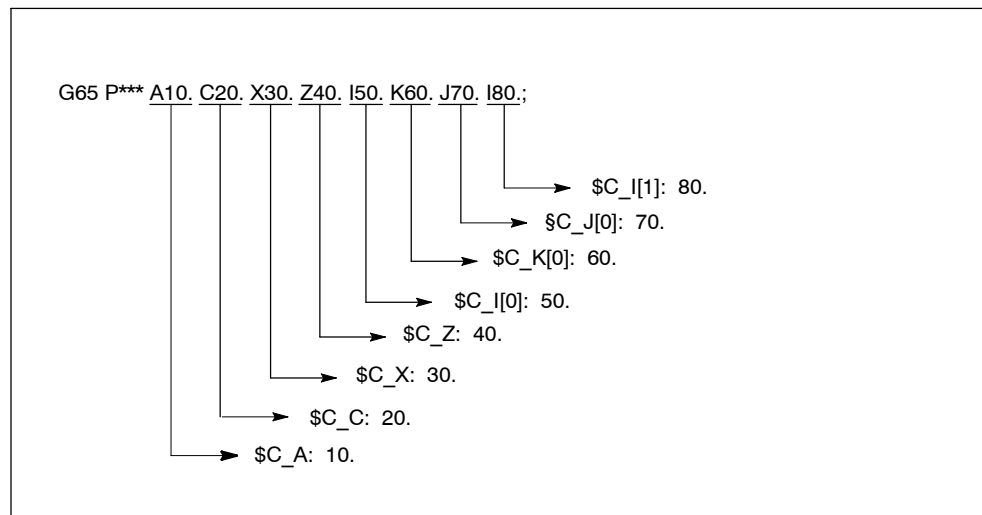


Fig. 4-43 Example of argument specification

## 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.

## 4.5 Macroprograms

### 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:
N10 G290 ; Switching into Siemens mode,
           ; if transfer parameters need to be read
N20 G01 F=$C_F G95 S=$C_S
N30 G1 X=$C_X Y=$C_Y
N40 G291 ; Switching into ISO mode
N50 M3 G54 T1
N60
...
N80 M99
```

## 4.6 Advanced functions

### 4.6.1 High-speed cycle cutting (G05)

The G05 command is used to call any subprogram similar to a M98 P\_ subprogram call. The subprogram to be called can be a pre-compiled partprogram deriving from Siemens code.

#### Format

G05 Pxxxxx Lxxx ;

**Pxxxxx** program number to be called

**Lxxx** number of repetitions

(L1 applies when this parameter is omitted)

#### Example

G05 P10123 L3 ;

This block calls program 10123.mpf and executes it 3 times.

#### Limitations

- Only Siemens code part programs can be pre-compiled.
- When calling a subprogram by G05, it is not switched into Siemens mode. The G05 command behaves like a M98 P\_ subprogram call.
- A block containing a G05 command without address P is ignored without alarm.
- A block containing a G05.1 command with or without address P as well as G05 P0 or G05 P01 is ignored without alarm.

## 4.6 Advanced functions

## 4.6.2 Polygonal turning

When rotating the workpiece and a tool at a certain ratio, a polygonal figure can be machined.

For example, by changing conditions such as rotation ratio of workpiece and tool as well as the number of cutters, a square or hexagon can be machined. Under certain circumstances, the machining time can be reduced compared to machining using C and X axis in polar coordinate interpolation.

Due to the nature of such kind of machining however, the machined figure is not exactly polygonal. Typical applications are the heads of square and/or hexagon bolts or nuts.

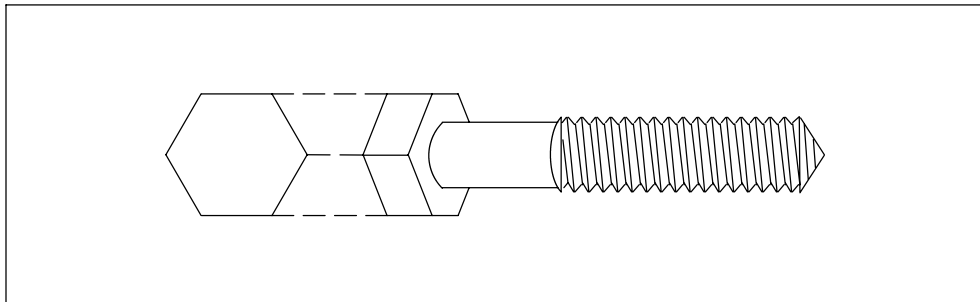


Fig. 4-44 Hexagon bolt

## Format

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

**P, Q:** Rotation ratio (spindle / Y axis)

Setting range: Integer 1 to 9 for both P and Q

The sign of address Q is used to specify the Y axis rotation direction.

**Example**

G00 X120.0 Z30.0 S1200.0 M03; set workpiece rotation speed to 1200 rpm  
G51.2 P1 Q2 ; start tool rotation (2400 rpm)  
G01 X80.0 F10.0 ; X axis infeed  
G04 X2. ;  
G00 X120.0 ; X axis retract  
G50.2 ; stop tool rotation  
M05 ; Spindle stop

G50.2 and G51.2 need to be specified in separate blocks.

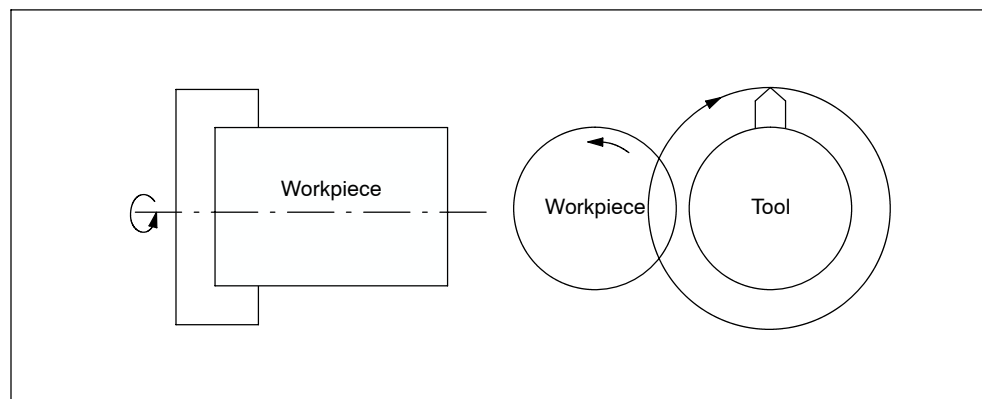


Fig. 4-45 Polygonal turning

## 4.6 Advanced functions

## 4.6.3 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
```

#### 4.6.4 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).

---

##### Notice

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!**

---

---

##### Notice

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!**

### 4.6.5 Interrupt programm with M96 / M97 (ASUP)

#### 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 REPOS)**
- 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

#### 4.6 Advanced functions

= 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, REPOS L 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

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

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

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

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

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

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



<b>PCMCIA</b>	Personal Computer Memory Card International Association
<b>PG</b>	Programming Device
<b>PLC</b>	Programmable Logic Control
<b>PP</b>	Production Planning
<b>RAM</b>	Random Access Memory (read-write memory)
<b>REF</b>	Reference Point Approach Function
<b>REPOS</b>	Reposition Function
<b>ROV</b>	Rapid Override
<b>RPA</b>	R Parameter Active: Memory area in the NCK for R-NCK for R parameter numbers
<b>RPY</b>	Roll Pitch Yaw: Type of coordinate system rotation
<b>RTS</b>	Request To Send (serial data interfaces)
<b>SBL</b>	Single Block
<b>SD</b>	Setting Data
<b>SDB</b>	System Data Block
<b>SEA</b>	Setting Data Active: Identification (file type) for setting data
<b>SFB</b>	System Function Block
<b>SFC</b>	System Function Call
<b>SK</b>	Softkey

---

<b>SKP</b>	Skip Block
<b>SM</b>	Stepper Motor
<b>SOP</b>	Shopfloor-Oriented Programming
<b>SPF</b>	Sub Program File (subroutine file)
<b>SR</b>	Subroutine
<b>SRAM</b>	Static RAM (battery-backed)
<b>STL</b>	Statement List
<b>SSI</b>	Serial Synchronous Interface
<b>SW</b>	Software
<b>SYF</b>	System Files
<b>T</b>	Tool
<b>TC</b>	Tool Change
<b>TEA</b>	Testing Data Active: Identifier for machine data
<b>TLC</b>	Tool length compensation
<b>TNRC</b>	Tool Nose Radius Compensation
<b>TO</b>	Tool Offset
<b>TOA</b>	Tool Offset Active: Identification (file type) for tool offsets
<b>TRANSMIT</b>	Transform Milling into Turning: Coordinate conversion on turning machines for milling operations
<b>TRC</b>	Tool Radius Compensation

<b>UFR</b>	User Frame: Zero offset
<b>V.24</b>	Serial Interface (definition of interchange lines between DTE and DCE)
<b>WCS</b>	Workpiece Coordinate System
<b>WPD</b>	Work Piece Directory
<b>ZO</b>	Zero Offset
<b>ZOA</b>	Zero Offset Active: Identification (file type) for zero offset data

## Notes

[illegible]

# 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.

<b>Activate/deactivate</b>	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.
<b>Address</b>	Addresses are fixed or variable identifiers for axes (X, Y, ...), spindle speed (S), feedrate (F), circle radius (CR), etc.
<b>Alarms</b>	<p>All -&gt; 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.</p> <p>Alarms and messages are displayed separately.</p> <ol style="list-style-type: none"><li>1. Alarms and messages in the part program Alarms and messages can be displayed directly from the part program in plaintext.</li><li>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.</li></ol>
<b>Analog input/output module</b>	<p>Analog input/output modules are signal transducers for analog process signals.</p> <p>Analog input modules convert analog measured values into digital values that can be processed in the CPU.</p> <p>Analog output modules convert digital values into manipulated variables.</p>
<b>Approach fixed machine point</b>	Approach motion towards one of the predefined -> fixed machine points.
<b>Archiving</b>	Exporting files and/or directories to an <b>external</b> storage device.

<b>Asynchronous subroutine</b>	<ul style="list-style-type: none"><li>• 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).</li><li>• 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).</li></ul>
<b>Automatic</b>	Control system operating mode (block-sequential to DIN): Mode in NC systems in which a -> part program is selected and continuously executed.
<b>Auxiliary functions</b>	Auxiliary functions can be used to pass -> parameters to the -> PLC in -> part programs, triggering reactions there which are defined by the machine manufacturer.
<b>Axes</b>	<p>CNC axes are classified according to their functional scope as:</p> <ul style="list-style-type: none"><li>• Axes: Interpolative path axes</li><li>• 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.</li></ul>
<b>Axis address</b>	See -> axis identifier
<b>Axis identifier</b>	<p>In compliance with DIN 66217, axes are identified as X, Y and Z for a right-handed rectangular -&gt; coordinate system.</p> <p>-&gt; 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.</p>
<b>Axis name</b>	See -> axis identifier
<b>Axis/spindle replacement</b>	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.



<b>Block search</b>	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.
<b>Booting</b>	Loading the system program after Power ON.
<b>Bus connector</b>	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.
<b>C</b>	
<b>C axis</b>	Axis about which the tool spindle describes a controlled rotational and positioning movement.
<b>C spline</b>	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.
<b>Channel structure</b>	The channel structure makes it possible to process the -> programs of individual channels simultaneously and asynchronously.
<b>Circular interpolation</b>	The -> tool is required to travel in a circle between defined points on the contour at a specified feed while machining the workpiece.
<b>Clearance control (3D), sensor-driven</b>	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.
<b>CNC</b>	-> NC
<b>CNC high-level language</b>	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.

<b>CNC programming language</b>	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).
<b>COM</b>	Numerical control component for the implementation and coordination of communication.
<b>Command axis</b>	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.
<b>Compensation axis</b>	Axis having a setpoint or actual value modified by the compensation value.
<b>Compensation table</b>	Table of interpolation points. It supplies the compensation values of the compensation axis for selected positions of the base axis.
<b>Compensation value</b>	Difference between the axis position measured by the position sensor and the desired, programmed axis position.
<b>Connecting cables</b>	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.
<b>Continuous-path mode</b>	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.
<b>Contour</b>	Outline of a -> workpiece.
<b>Contour monitoring</b>	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.

<b>Coordinate system</b>	See -> machine coordinate system, -> workpiece coordinate system
<b>CPU</b>	Central Processor Unit -> programmable controller
<b>Cycle</b>	Protected subroutine for executing a recurring machining operation on the -> workpiece.
<b>Cycles support</b>	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.
<b>D</b>	
<b>Data block</b>	<ol style="list-style-type: none"><li>1. Data unit of the -&gt; PLC which can be accessed by -&gt; HIGHSTEP programs.</li><li>2. Data unit of the -&gt; NC: Data blocks contain data definitions for global user data. These data can be initialized directly when they are defined.</li></ol>
<b>Data transfer program PCIN</b>	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.
<b>Data word</b>	A data unit, two bytes in size, within a -> PLC data block.
<b>Deletion of distance-to-go</b>	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.

<b>Feedrate override</b>	<p>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.</p> <p>An offset can also be applied via motion-synchronous actions independently of the running program.</p>
<b>Finished-part contour</b>	Contour of the finished workpiece. See also -> blank.
<b>Fixed machine point</b>	A point defined uniquely by the machine tool, such as the reference point.
<b>Fixed-point approach</b>	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.
<b>Frame</b>	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.
<b>G</b>	
<b>General reset</b>	<p>The following memories of the -&gt; CPU are erased by a general reset operation:</p> <ul style="list-style-type: none"><li>• -&gt; Working memory</li><li>• Read/write area of the -&gt; load memory</li><li>• -&gt; System memory</li><li>• -&gt; Backup memory</li></ul>
<b>Geometry</b>	Description of a -> workpiece in the -> workpiece coordinate system.
<b>Geometry axis</b>	Geometry axes are used to describe a 2 or 3-dimensional area in the workpiece coordinate system.

<b>Global main run/subroutine</b>	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.
<b>Ground</b>	"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.
<b>H</b>	
<b>Helical interpolation</b>	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: <ol style="list-style-type: none"> <li>1. Circular movement in one plane</li> <li>2. Linear movement perpendicular to this plane.</li> </ol>
<b>High-speed digital inputs/outputs</b>	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).
<b>HIGHSTEP</b>	Combination of the programming features for the -> PLC in the S7-300/400 range.
<b>I</b>	
<b>Identifier</b>	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.
<b>Inch system of measurement</b>	System of measurement that defines distances in "inches" and fractions thereof.
<b>Inclined axis</b>	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.

<b>Increment</b>	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.
<b>Increment</b>	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.
<b>Initialization block</b>	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.
<b>Initialization file</b>	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.
<b>Intermediate blocks</b>	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.
<b>Interpolation cycle</b>	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.
<b>Interpolative compensation</b>	Interpolative compensation provides a means of compensating for leadscrew errors (LEC) and measuring-system errors (MSEC) resulting from the production process.
<b>Interpolator</b>	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.



<b>Interrupt routine</b>	<p>Interrupt routines are special -&gt; 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.</p> <p>See -&gt; ASUB</p>
<b>Inverse-time feedrate</b>	<p>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).</p>
<b>I/O module</b>	<p>I/O modules create the link between the CPU and the process. I/O modules are:</p> <ul style="list-style-type: none"> <li>• -&gt;Digital input/output modules</li> <li>• -&gt;Analog input/output modules</li> <li>• -&gt;Simulator modules</li> </ul>
<b>J</b> <b>Jog</b>	<p>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 -&gt; reference point approach, -&gt; Repos and -&gt; Preset -&gt; (set actual value).</p>
<b>K</b> <b>Keyswitch</b>	<ol style="list-style-type: none"> <li><b>1. S7-300:</b> The keyswitch is the mode selector switch on the -&gt; CPU. The keyswitch is operated by means of a removable key.</li> <li><b>2. 840D/FM-NC:</b> The keyswitch on the -&gt; 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.</li> </ol>
<b>K<sub>Ü</sub></b>	Transmission Ratio
<b>K<sub>v</sub></b>	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.

<b>Machine coordinate system</b>	System of coordinates based on the axes of the machine tool.
<b>Machine zero</b>	A fixed point on the machine tool which can be referenced by all (derived) measurement systems.
<b>Machining channel</b>	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.
<b>Macros</b>	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.
<b>Main block</b>	A block prefixed by ":" containing all the parameters required to start execution of a -> part program.
<b>Main program</b>	-> Part program identified by a number or name in which other main programs, subroutines or -> cycles may be called.
<b>Main run</b>	Part program blocks which have been decoded and prepared by the preprocessor are executed during the "main run".
<b>MDA</b>	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.

<b>Measuring circuits</b>	<ul style="list-style-type: none"><li>• 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.</li><li>• 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.</li></ul>
<b>Messages</b>	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.
<b>Metric system</b>	Standardized system of units for lengths in millimeters (mm), meters (m), etc.
<b>Mirroring</b>	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.
<b>Mode</b>	An operating concept on a SINUMERIK control. The modes -> Jog, -> MDA, -> Automatic are defined.
<b>Mode group</b>	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.
<b>Motion synchronization</b>	<p>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:</p> <p>Transfer M and H auxiliary functions to the PLC or deletion of distance-to-go for specific axes.</p>

**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
- MMI (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

<b>PLC</b>	Programmable Logic Control -> Speicherprogrammierbare Steuerung. Component of the -> NC: Programmable controller for processing the control logic on the machine tool.
<b>PLC program memory</b>	<ul style="list-style-type: none"><li>• 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.</li><li>• 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.</li><li>• 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.</li></ul>
<b>PLC programming</b>	The PLC is programmed with the <b>STEP 7</b> software. The STEP 7 programming software is based on the standard <b>WINDOWS</b> operating system and incorporates the functionality of STEP 5 programming with innovative expansions and developments.
<b>Polar coordinates</b>	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.
<b>Polynomial interpolation</b>	Polynomial interpolation provides a means of generating a very wide range of curves, including <b>straight-line, parabolic and exponential functions</b> (SINUMERIK 840D/810D).
<b>Positioning axis</b>	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.
<b>Power ON</b>	The action of switching the control off and then on again.



<b>Preprocessing memory, dynamic</b>	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.
<b>Preprocessing stop</b>	<p>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.</p> <p>See also "Preprocessing memory".</p>
<b>Preset</b>	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.
<b>Program</b>	<ol style="list-style-type: none"><li>1. Control operating area</li><li>2. Sequence of instructions to the control system.</li></ol>
<b>Programmable frames</b>	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.
<b>Programmable logic controller</b>	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.
<b>Programmable working area limitation</b>	Limitation of the movement area of the tool to within defined, programmable limits.

<b>Programming key</b>	Characters and character sequences which have a defined meaning in the programming language for -> part programs (see Programming Guide).
<b>Protection zone</b>	Three-dimensional area within a -> working area which the tool tip is not permitted to enter (programmable via MD).
<b>Q Quadrant error compensation</b>	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.
<b>R R parameter</b>	Calculation parameter. The programmer can assign or request the values of the R parameter in the -> part program as required.
<b>Rail</b>	This rail is used to mount the modules of the S7-300 system.
<b>Rapid traverse</b>	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.
<b>Reference point</b>	Point on the machine tool with which the measuring system of the -> machine axes is referenced.
<b>Reference point approach</b>	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.

<b>Softkey</b>	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.
<b>Software limit switches</b>	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.
<b>Spindles</b>	<p>The spindle functionality is a two-level construct:</p> <ol style="list-style-type: none"><li>1. Spindles: Speed-controlled or position-controlled spindle drives, analog digital (SINUMERIK 840D)</li><li>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.</li></ol>
<b>Spline interpolation</b>	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.
<b>Standard cycles</b>	<p>Standard cycles are used to program machining operations which repeat frequently:</p> <ul style="list-style-type: none"><li>• For drilling/milling</li><li>• For measuring tools and workpieces</li></ul> <p>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.</p>
<b>Subblock</b>	Block prefixed by "N" containing information for a machining step such as a position parameter.
<b>Subroutine</b>	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.

<b>Synchronization</b>	Instructions in -> part programs for coordination of the operations in different -> channels at specific machining points.
<b>Synchronized actions</b>	<ol style="list-style-type: none"><li>1. Auxiliary function output While a workpiece is being machined, technological functions (-&gt; 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.</li><li>2. High-speed auxiliary function output The acknowledgement times for the -&gt; auxiliary functions can be minimized and unnecessary halts in the machining process avoided for time-critical switching functions.</li></ol> <p>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.</p>
<b>Synchronized axes</b>	Synchronized axes require the same amount of time to traverse their path as -> geometry axes for their path.
<b>Synchronous spindle</b>	<p>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.</p> <p>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.</p> <p>Several pairs of synchronous spindles can be implemented.</p>
<b>System variable</b>	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

<b>Teach In</b>	<b>Teach In</b> 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.
<b>Text editor</b>	-> Editor
<b>Tool</b>	A tool employed to shape the workpiece, for example, a turning tool, milling cutter, drill, laser beam, grinding wheel, etc.
<b>Tool nose radius compensation</b>	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.
<b>Tool offset</b>	<p>A tool is selected by programming a <b>T function</b> (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.</p> <p>Tool length compensation is selected by programming D numbers.</p>
<b>Tool radius compensation</b>	<p>In order to program a desired -&gt; 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).</p>
<b>Transformation</b>	<p>Programming in a Cartesian coordinate system, execution in a non-Cartesian coordinate system (e.g. with machine axes as rotary axes).</p> <p>Employed in conjunction with Transmit, Inclined Axis, 5-Axis Transformation.</p>

<b>Transmit</b>	<p>This function is used to mill the outside contours on turned parts, e.g. four-sided parts (linear axis with rotary axis).</p> <p>3D interpolation with two linear axes and one rotary axis is also possible.</p> <p>The benefits afforded by Transmit are simplified programming and improved machine efficiency through complete machining: Turning and milling on the same machine without reclamping.</p>
<b>Travel to fixed stop</b>	<p>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.</p>
<b>Traversing range</b>	<p>The maximum permissible travel range for linear axes is <math>\pm 9</math> decades. The absolute value depends on the selected input and position control resolution and the unit of measurement (inch or metric).</p>
<b>U</b>	
<b>User-defined variable</b>	<p>Users can define variables in the -&gt; 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 -&gt; system variable.</p>
<b>User memory</b>	<p>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.</p>
<b>User program</b>	<p>-&gt; Part program</p>
<b>V</b>	
<b>Variable definition</b>	<p>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.</p>
<b>Velocity control</b>	<p>In order to achieve an acceptable travel velocity in movements which call for very small adjustments of position in a block, the control can -&gt; look ahead.</p>
<b>Vocabulary words</b>	<p>Words with a specific notation which have a defined meaning in the programming language for -&gt; part programs.</p>



**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

Appendix 1 describes the G code and the functions.

### C.1 G code table

Table C-1 G code table

G code		Description	840D sl	802D sl
<b>Group 1</b>				
G00 <sup>1)</sup>	1	Rapid traverse	x	x
G01	2	Linear motion	x	x
G02	3	Circle/helix, clockwise	x	x
G03	4	Circle/helix, counterclockwise	x	x
G33	5	Thread cutting with constant lead	x	x
G34	9	Thread cutting with variable lead	x	x
G77	6	Longitudinal turning cycle	x	x
G78	7	Thread cutting cycle	x	x
G79	8	Face turning cycle	x	x
<b>Group 2</b>				
G96	1	Constant cutting rate ON	x	x
G97 <sup>1)</sup>	2	Constant cutting rate OFF	x	x
<b>Group 3</b>				
G90 <sup>1)</sup>	1	Absolute programming	x	x
G91	2	Incremental programming	x	x
<b>Group 4</b>				
G68	1	Dual slide / turret on	x	x
G69 <sup>1)</sup>	2	Dual slide / turret off	x	x
<b>Group 5</b>				
G94	1	Feed in [mm/min, inch/min]	x	x
G95 <sup>1)</sup>	2	Feed in [mm/rev, inch/rev]	x	x
<b>Group 6</b>				
G20 <sup>1)</sup>	1	Input system inch	x	x
G21	2	Input system metric	x	x

## C.1 G code table

Table C-1 G code table

G code		Description	840D sl	802D sl
<b>Group 7</b>				
G40 <sup>1)</sup>	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
<b>Group 8</b>				
<b>Group 9</b>				
G22	1	Working area limitation, protection zone 3 ON	x	x
G23 <sup>1)</sup>	2	Working area limitation, protection zone 3 OFF	x	x
<b>Group 10</b>				
G80 <sup>1)</sup>	1	Drilling cycle off	x	x
G83	2	Face deep hole drilling	x	x
G84	3	Face tapping	x	x
G85	4	Face drilling	x	x
G87	5	Side deep hole drilling	x	x
G88	6	Side tapping	x	x
G89	7	Side drilling	x	x
<b>Group 11</b>				
G98 <sup>1)</sup>	1	Return to starting point for drilling cycles	x	x
G99	2	Return to point R for drilling cycles	x	x
<b>Group 12</b>				
G66	1	Modal macro call	x	x
G67 <sup>1)</sup>	2	Delete modal macro call	x	x
<b>Group 13</b>				
<b>Group 14</b>				
G54 <sup>1)</sup>	1	Select zero offset	x	x
G55	2	Select zero offset	x	x
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
G54 P{1...48}1		extended zero offsets	x	x
G54 .1	7	extended zero offsets	x	x
G54 P0	1	external zero offsets EXOFS	x	x

Table C-1 G code table

G code		Description	840D sl	802D sl
<b>Group 15</b>				
<b>Group 16</b>				
G17	1	XY plane	x	x
G18 <sup>1)</sup>	2	ZX plane	x	x
G19	3	YZ plane	x	x
<b>Group 17</b>				
<b>Group 18 (non-modal)</b>				
G04	1	Dwell	x	x
G05	20	High-speed cycle cutting	x	x
G05.1	22	High-speed cycle -> Calling CYCLE305	x	x
G07.1	18	Cylindrical interpolation	x	x
G10	2	Write zero offset / tool offset	x	x
G10.6	19	Rapid lift ON/OFF	x	x
G27	16	Referencing check (available soon)	x	x
G28	3	Approach 1st reference point	x	x
G30	4	Approach 2nd/3rd/4th reference point	x	x
G30.1	21	Floating reference position	x	x
G31	5	Measurement with touch-trigger probe	x	x
G52	6	Additive zero offset	x	x
G53	17	Approach position in machine coordinate system	x	x
G60	24	Oriented positioning	x	x
G65	7	Call macro	x	x
G70	8	Finishing cycle	x	x
G71	9	Stock removal cycle longitudinal axis	x	x
G72	10	Stock removal cycle transverse axis	x	x
G73	11	Repeat contour	x	x
G74	12	Deep hole drilling and recessing in longitudinal axis (Z)	x	x
G75	13	Deep hole drilling and recessing in facing axis (X)	x	x
G76	14	Multiple thread cutting cycle	x	x
G92	15	Preset actual value memory, spindle speed limitation	x	x
G92.1	23	Delete actual value memory, reset of WCS	x	x
<b>Group 20</b>				
G50.2 <sup>1)</sup>	1	Synchronous spindle OFF	x	--
G51.2	2	Synchronous spindle ON	x	--

## C.1 G code table

Table C-1 G code table

G code		Description	840D sl	802D sl
<b>Group 21</b>				
G13.1 <sup>1)</sup>	1	TRANSMIT OFF	x	x
G12.1	2	TRANSMIT ON	x	x
<b>Group 22</b>				
<b>Group 25</b>				
<b>Group 31</b>				
G290 <sup>1)</sup>	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: The NC establishes the G code modes, identified by 1), when the power is turned ON or when the NC is reset.

# Machine and Setting Data

## D.1 Machine/Setting Data

<b>10604</b>	<b>WALIM_GEOAX_CHANGE_MODE</b>		
MD number	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		NCFRAME_POWERON_MASK			
MD number		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.					

## D.1 Machine/Setting Data

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:		<p>The setting is effective for Siemens G code programming only, i.e. G290.</p> <p>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.</p> <p>The name must be unique, i.e. axes, variables, macros, etc. must not exist with the same name.</p>	
This MD cannot SINUMERIK 802D sl.			

<b>10654</b>		<b>RADIUS_NAME</b>	
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.			

<b>10656</b>		<b>CHAMFER_NAME</b>	
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:	<p>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.</p> <p>The name must be unique, i.e. axes, variables, macros, etc. must not exist with the same name.</p> <p>The setting is effective for Siemens G code programming, i.e. G290.</p> <p>The chamfer in the original direction of movement. Alternatively, the chamfer length can be programmed with the name CHF.</p>		
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:	<p>DRYRUN_MASK == 0 Dryrun must only be activated or deactivated at the end of a block.</p> <p>DRYRUN_MASK == 1 Dry run feed may be activated or deactivated even during program execution</p> <p><u>Note:</u> Once dry run feed has been activated, the axes are stopped for the duration of the reorganization.</p> <p>DRYRUN_MASK == 2 Dryrun can be activated or deactivated in any phase and the axes are not stopped.</p> <p><u>Note:</u> 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.</p>		
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:	<p>SLASH_MASK == 0 The block skip function can only be switched over at the end of a block.</p> <p>SLASH_MASK == 1 When SLASH_MASK == 1 the block skip function may be activated even during program execution.</p> <p><u>Note:</u> Once block skip has been activated, the axes are stopped for the duration of the reorganization.</p> <p>SLASH_MASK == 2 Block switchover is possible in any phase.</p> <p><u>Note:</u> 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.</p>		
This MD cannot SINUMERIK 802D sl.			

## D.1 Machine/Setting Data

10715	M_NO_FCT_CYCLE[0]		
MD number	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:	<p>M number with which a subprogram is called.</p> <p>The name of the subprogram is stored in \$MN_M_NO_FCT_CYCLE_NAME. If the M function defined by \$MN_M_NO_FCT_CYCLE is programmed in a part program, 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.</p> <p>\$MN_M_NO_FCT_CYCLE is effective both in Siemens mode G290 and in external language mode G291.</p> <p>A subprogram call may not be superimposed on M functions with fixed meanings. In the event of a conflict, alarm 4150 is output:</p> <ul style="list-style-type: none"><li>- M0 to M5,</li><li>- M17, M30,</li><li>- M40 to M45,</li><li>- M function for spindle/axis mode switchover according to \$MC_SPIND_RIGID_TAPPING_M_NR (default M70)</li><li>- M functions for nibbling/punching according to configuration via \$MC_NIBBLE_PUNCH_CODE if activated via \$MC_PUNCHNIB_ACTIVATION.</li></ul> <p>- With applied external language (\$MN_MM_EXTERN_LANGUAGE) M19, M96-M99.</p> <p>Exception: The M functions defined for the tool change with \$MC_TOOL_CHANGE_M_CODE.</p> <p>\$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.</p> <p>Alarm 14016 is output in the event of a conflict.</p>		

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:	<p>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.</p> <p>\$MN_M_NO_FCT_CYCLE is effective both in Siemens mode G290 and in external language mode G291.</p> <p>If a T number is programmed in the calling block, the programmed T number can be scanned in the cycle in variable \$P_TOOL.</p> <p>\$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.</p> <p>Alarm 14016 is output in the event of a conflict.</p>		

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:	<p>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.</p> <p>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).</p> <p>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.</p> <p>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.</p> <p>\$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.</p> <p>\$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.</p> <p>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.</p> <p>Alarm 14016 is output in the event of a conflict.</p>		

## D.1 Machine/Setting Data

<b>10760</b>	<b>G53_TOOLCORR</b>		
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:	<p>The MD is effective in both Siemens mode and in external language mode.</p> <p>This machine data defines whether tool length compensation and tool radius compensation are suppressed with language commands G53, G153 and SUPA.</p> <p>0 = G53/G153/SUPA is non-modal suppression of zero offsets, tool length compensation and tool radius compensation remain active.</p> <p>1= G53/G153/SUPA is non-modal suppression of zero offsets, and active tool length and tool radius compensation.</p>		

<b>10800</b>	<b>EXTERN_CHAN_SYNC_M_NO_MIN</b>		
MD number	First M 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.			

<b>10802</b>	<b>EXTERN_CHAN_SYNC_M_NO_MAX</b>		
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 synchronization.		
	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.			

<b>10804</b>	<b>EXTERN_M_NO_SET_INT</b>		
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:	<p>Using the data bits described below, the behaviour of the interruption type subprogram activated by M96 P .. can be specified.</p> <p>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</p> <p>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</p> <p>Bit 2: =0, The current NC block is interrupted immediately and the subprogram is called if the interrupt signal is detected. =1, The subprogram is called after completion of the current NC block</p> <p>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</p>		

<b>10810</b>	<b>EXTERN_MEAS_G31_P_SIGNAL</b>		
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:	<p>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.</p> <p>For example:</p> <p>\$MN_EXTERN_MEAS_G31_P_SIGNAL[1], Bit 0=1, the 1st measuring input is activated by G31 P2.</p> <p>\$MN_EXTERN_MEAS_G31_P_SIGNAL[3] = 2, the 2nd measuring input is activated by G31 P4.</p> <p>Bit 0:   =0:    Deactivate measuring input 1 for G31 P1 (-P4)           =1    Activate measuring input 1 for G31 P1 (-P4)</p> <p>Bit 1:   =0    Deactivate measuring input 2 for G31 P1 (-P4)           =1    Activate measuring input 2 for G31 P1 (-P4)</p>		

## D.1 Machine/Setting Data

10812	EXTERN_DOUBLE_TURRET_ON		
MD number	Double turret with G68		
Default setting:		Minimum input limit:	Maximum input limit:
Changes effective after:		Protection level:	Unit: -
Data type: BOOLEAN		Applies with effect from SW version: 6.2	
Meaning:	<p>This MD defines whether G68 should activate double slide machining (channel synchronization for the first and second channel) or whether the second tool of a double revolver should be activated (= 2, with the spacing defined in setting data \$SC_EXTERN_DOUBLE_TURRET_DIST, fully interconnected tool).</p> <p>FALSE: Channel synchronization for double slide machining</p> <p>TRUE: Load the second tool of a double turret (=\$SC_EXTERN_DOUBLE_TURRET_DISTANCE as an additive zero offset and activate mirroring about the Z axis</p>		

10814		EXTERN_M_NO_MAC_CYCLE	
MD number		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:	<p>M number with which a macro is called.</p> <p>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.</p> <p>\$MN_EXTERN_M_NO_MAC_CYCLE_NAME[n] is active only in the external language mode G291.</p> <p>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:</p> <ul style="list-style-type: none"><li>- M0 to M5,</li><li>- M17, M30,</li><li>- M19,</li><li>- M40 to M45,</li><li>- M function up to switchover of spindle mode/axis mode according to \$MC_SPIND_RIGID_TAPPING_M_NR (default: M70),</li><li>- M function for nibbling/punching acc. to configuration via \$MC_NIBBLE_PUNCH_CODE if they have been activated via \$MC_PUNCHNIB_ACTIVATION.</li><li>- with external language applied (\$MN_MM_EXTERN_LANGUAGE) additionally M96 to M99</li><li>- M functions which are defined by \$MN_M_NO_FCT_CYCLE.</li></ul> <p>Exeption: The M function defined with \$MC_TOOL_CHANGE_M_CODE for tool change.</p> <p>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.</p>		

## D.1 Machine/Setting Data

10185	EXTERN_M_NO_MAC_CYCLE_NAME		
MD number	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].		

10816	EXTERN_G_NO_MAC_CYCLE		
MD number	Macro call with G function		
Default setting:		Minimum input limit:	Maximum input limit:
Changes effective after Power ON		Protection level:	Unit: -
Data type: DOUBLE		Applies with effect from SW version: 6.3	
Meaning:	<p>G number with which a macro is called.</p> <p>The name of the subprogram is specified in \$MN_EXTERN_G_NO_MAC_CYCLE_NAME[n].</p> <p>If the G function defined with \$MN_EXTERN_G_NO_MAC_CYCLE[n] is programmed in a parts program block, the subprogram defined in EXTERN_M_NO_MAC_CYCLE_NAME[n] is started, all the addresses programmed in the block are written to the associated \$C_xx variables.</p> <p>If a subprogram call is already active via a M/G macro or an M substitution, no subprogram call will be executed. If a standard G function is programmed in this case, it will be executed, otherwise alarm 12470 is output.</p> <p>\$MN_EXTERN_G_NO_MAC_CYCLE[n] is only effective in external language mode G291.</p> <p>A block can only contain one subprogram call, i.e. only one M/G function substitution may be programmed in a block and the block must not contain any additional subprogram (M98) or cycle call.</p> <p>A subprogram return jump or end of parts program is not allowed in the same block. Alarm 14016 is output in the event of a conflict.</p>		

10817	EXTERN_G_NO_MAC_CYCLE_NAME		
MD number	Subprogram name for G function macro call		
Default setting:		Minimum input limit:	Maximum input limit:
Changes effective after Power ON		Protection level:	Unit: -
Data type: STRING		Applies with effect from SW version: 6.3	
Meaning:	Cycle name if called with G function defined with \$MN_EXTERN_G_NO_MAC_CYCLE[n].		

## D.1 Machine/Setting Data

<b>10818</b>	<b>EXTERN_INTERRUPT_NUM_ASUP</b>		
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:	<p>Selection of the external language</p> <p>1 = ISO-2: System Fanuc0 Milling (from 5.1) 2 = ISO-3: System Fanuc0 Turning (from 5.2)</p> <p>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.</p>		

10881	EXTERN_GCODE_SYSTEM		
SD number	ISO mode T: G code system		
Default setting: 0		Minimum input limit: 0	Maximum input limit: 2
Changes effective after Power On		Protection level: 2/7	Unit: -
Data type: DWORD		Applies with effect from SW version: 6.2	
Meaning:	This MD determines the G code system used for ISO dialect T mode: setting value = 0: ISO_T: G code system B setting value = 1: ISO_T: G code system A setting value = 2: ISO_T: G code system C		



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: -
Datatype: STRING		Applies with effect from SW version: 5	
Meaning:	<p>Code B is implemented by default for external programming language ISO Dialect0-T. Code</p> <p>A and Code C have different G function names.</p> <p>\$MN_NC_USER_EXTERN_GCODES_TAB can be used to rename the G functions.</p> <p>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.</p> <p>Up to 30 code changes are possible. Example:</p> <p>\$MN_NC_USER_EXTERN_GCODES_TAB[0]="G20"</p> <p>\$MN_NC_USER_EXTERN_GCODES_TAB[1]="G70"</p> <p>--&gt; G20 is reassigned to G70;</p> <p>If G70 already exists, an error message appears on NCK reset.</p>		

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:	<p>This machine data is effective for external programming languages, i.e. if MD 18800: MM_EXTERN_LANGUAGE = 1.</p> <p>The machine data defines how programmed values without decimal points are evaluated.</p> <p>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</p> <p>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</p>		

## D.1 Machine/Setting Data

<b>10886</b>	<b>EXTERN_INCREMENT_SYSTEM</b>		
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:	<p>This machine data is effective for external programming languages, i.e. if MD 18800: MM_EXTERN_LANGUAGE = 1.</p> <p>This machine data defines which increment system is active</p> <p>0: Increment system IS-B = 0.001 mm/degree = 0.0001 inch</p> <p>1: Increment system IS-C = 0.0001 mm/degree = 0.00001 inch</p>		

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.	

10890	EXTERN_TOOLPROG_MODE		
MD number	Tool change programming with external programming language		
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:	<p>Configuration of tool change programming for external programming language:</p> <p>Bit0 = 0: Effective only with \$MN_MM_EXTERN_CNC_LANGUAGE =2: The tool number and offset number are programmed in the T value. \$MN_DIGITS_TOOLNO determines the number of leading digits representing the tool number.</p> <p>Example: \$MN_DIGITS_TOOL_NO = 2 T=1234 ; tool no. 12, ; offset no. 34</p> <p>Bit0 = 1: Effective only with \$MN_MM_EXTERN_CNC_LANGUAGE =2: Only the tool number is programmed in the T value. Offset number = tool number. \$MN_DIGITS_TOOL_NO is irrelevant.</p> <p>Example: T=12 ; tool no. 12 ; offset no. 12</p> <p>Bit1 = 0: Effective only with \$MN_MM_EXTERN_CNC_LANGUAGE =2: If the number of digits programmed in the T value is equal to the number in \$MN_EXTERN_DIGITS_TOOL_NO, leading zeroes are added.</p> <p>Bit1 = 1: Effective only with \$MN_MM_EXTERN_CNC_LANGUAGE =2: If the number of digits programmed in the T value is equal to the number of digits specified in \$MN_EXTERN_DIGITS_TOOL_NO, the programmed number is used as the offset number and the tool number</p>		

<b>18800</b>	<b>MM_EXTERN_LANGUAGE</b>		
MD number	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

## D.2 Channel-specific machine data

20050	AXCONF_GEOAX_ASSIGN_TAB		
MD number	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).		

<b>20060</b>	<b>AXCONF_GEOAX_NAME_TAB</b>		
MD number	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		AXCONF_MACHAX_USED	
MD number		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.		

<b>20080</b>	<b>AXCONF_CHANAX_NAME_TAB</b>		
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:	<p>In this MD you can set the name of the channel axis.</p> <p>The channel axis is displayed with this identifier in the WCS. This identifier is also written in the program.</p> <p>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.</p>		

## D.2 Channel-specific machine data

20150		GCODE_RESET_VALUES	
MD Number		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)
	...		

<b>20154</b>	<b>EXTERN_GCODE_RESET_VALUES[n]: 0, ..., 30</b>		
MD number	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 documenta- tion.		
	The following groups can be defined within MD EXTERN_GCODE_RESET_VALUES:		
	<b>ISO dialect M:</b>	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	
<b>ISO dialect T:</b>	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.			

## D.2 Channel-specific machine data

<b>20380</b>	<b>TOOL_CORR_MODE_G43/G44</b>		
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:	<p>The machine data is only effective if MD EXTERN_CNC_LANGUAGE = 1.</p> <p>When G43/G44 is active, it determines how length offsets programmed with H are processed.</p> <p>0: mode A The tool length H always acts on the Z axis, independent of the current plane.</p> <p>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)</p> <p>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.</p> <p>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 described under mode B.</p>		
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.			

## D.2 Channel-specific machine data

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	External language function mask		
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:	<p>This MD affects functions included in the ISO mode.</p> <p>Bit 0 =0: ISO mode T: "A" and "C" are interpreted as axis designation. When programming a contour, "A" or "C" must be preceded by a comma. =1: "A" and "C" within a part program are always interpreted as contour definition. An A- or C axis must not exist.</p> <p>Bit 1 =0: ISO mode M G10 P&lt;100 tool geometry   </p>		

22420	FGROUP_DEFAULT_AXIS[n]: 0, ..., 7		
MD number	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:	<p>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.</p> <p>Example: The first 4 axes in the channel are relevant for the path feed:</p> <p>\$MC_FGROUP_DEFAULT_AXES[0] = 1</p> <p>\$MC_FGROUP_DEFAULT_AXES[2] = 2</p> <p>\$MC_FGROUP_DEFAULT_AXES[3] = 3</p> <p>\$MC_FGROUP_DEFAULT_AXES[4] = 4</p>		
This MD cannot SINUMERIK 802D sl.			

## D.2 Channel-specific machine data

22512	EXTERN_GCODE_GROUPS_TO_PLC[n]: 0, ..., 7		
MD number	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:	<p>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.</p> <p>Default 0: No output</p> <p>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).</p> <p>The same applies to \$MC_GCODE_GROUPS_TO_PLC</p>		
This MD cannot SINUMERIK 802D sl.			

22515	GCODE_GROUPS_TO_PLC_MODE		
MD number	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:	<p>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.</p> <p>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.</p> <p>Bit 0(LSB = 0: Behavior as before, the 64 byte array is used for the G code.</p> <p>Bit 0(LSB = 1: The user sets for which G groups the first 8 bytes will be used</p>		
This MD cannot SINUMERIK 802D sl.			

22900		STROKE_CHECK_INSIDE	
MD number		Determine internal/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.			



## D.2 Channel-specific machine data

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		

<b>22914</b>	<b>AXES_SCALE_ENABLE</b>		
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: -
Datatype: 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		
SD number	Activation of fixed feedrates		
Default setting: FALSE		Minimum input limit:	Maximum input limit:
Changes effective after POWER ON		Protection level: 2/7	Unit:
Datatype: BOOLEAN		Applies with effect from SW version: 6.2	
Meaning:	FALSE: feedrate values stored in setting data \$SC_FEEDRATE_F1_9[ ] cannot be activated by F1 - F9. TRUE When programming F1 - F9, the feedrate values stored in setting data \$SC_FEEDRATE_F1_9[ ] become effective. F0 activates rapid traverse.		

## D.2 Channel-specific machine data

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.		

<b>24004</b>		<b>CHBFRAME_POWERON_MASK</b>	
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.			

## D.3 Axis-specific setting data

43120	DEFAULT_SCALE_FACTOR_AXIS		
MD number	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.		

<b>43240</b>	<b>M19_SPOS</b>		
MD number	Spindle position in degrees for spindle positions with 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:	The setting data is also effective in Siemens mode.		

<b>43240</b>	<b>M19_SPOS</b>		
MD number	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

## 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_EX-TERN_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.		

<b>42150</b>	<b>DEFAULT_ROT_FACTOR_R</b>		
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 F0 - 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 F0 - F9 when G01 is active.		

## D.4 Channel-specific setting data

<b>42162</b>	<b>EXTERN_DOUBLE_TURRET_DIST</b>		
SD number	Tool spacing on the double turret		
Default setting:		Minimum input limit:	Maximum input limit:
Changes effective		Protection level:	Unit:
Data type: DOUBLE		Applies with effect from SW version:	
Meaning:	Spacing of both the tools on a double slide turret. The spacing is activated as an additive zero offset when code G68 is used, if \$MN_EXTERN_DOUBLE_TURRET_ON = TRUE is set.		

[illegible]

## Data Fields, Lists

### E.1 Machine data

Number	Identifier	Name	Reference
<b>General (\$MN_ ... )</b>			
10604	WALIM_GEOAX_CHANGE_MODE	Work area limitation when switching geometrical 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.	
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..	
10812	EXTERN_DOUBLE_TURRET_ON	Double turret with G68	
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	
10816	EXTERN_G_NO_MAC_CYCLE	Macro call with G function	
10817	EXTERN_G_NO_MAC_CYCLE_NAME	Subprogram name for G function macro call	
10818	EXTERN_INTERRUPT_NUM_ASUP	Interruptnumber for ASUP start (M96)	
10820	EXTERN_INTERRUPT_NUM_RETRAC	Interruptnumber for retract (G10.6)	

## E.1 Machine data

<b>General (\$MN_ ...)</b>			
10880	EXTERN_CNC_SYSTEM	External control system whose programs are to be executed	
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	
<b>Channel-specific (\$MC_ ...)</b>			
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
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	
22515	GCODE_GROUPS_TO_PLC_MODE	Behavior of G group to PLC	
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	



Channel-specific (\$MC_...)			
28080	NUM_USER_FRAMES	Number of zero offsets	
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

## E.2 Setting data

Number	Identifier	Name	Reference
<b>Axis-specific</b>			
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	
<b>Channel-specific</b>			
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	
42162	\$SC_EXTERN_DOUBLE_TURRET_DIST	Tool spacing on the double turret	

## 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	REAL	Value of programmed address L in ISO Dialect mode for cycle programming
....	....	....
\$C_Z	REAL	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)

Identifier	Type	Description
\$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
\$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

[illegible]

# 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_LANGUAGE 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 code not possible	CYCLE383T, CYCLE384T, CYCLE385T	
61811	ISO axis designation not admissible	CYCLE328, CYCLE330, CYCLE371T, CYCLE374T, CYCLE376T, CYCLE383T, 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 admissible numeral value in cycle setting data
61814	Polar coordinate not possible	CYCLE381M, CYCLE383M, CYCLE384M, CYCLE387M	
61815	G40 not active	CYCLE374T, CYCLE376T	G40 was not active prior to the cycle call

# Commands

## C

COMPON, COMPCURV, COMPCAD, 4-150

## F

F command, 1-17

## G

G00, 2-23, C-195

G01, 2-25, C-195

G02, C-195

G02, G03, 2-27

G03, C-195

G04, 3-72, C-197

G05, 4-147, C-197

G05.1, C-197

G07.1, 2-33, C-197

G10, C-197

G10.6, 2-53, C-197

G12.1, C-198

G12.1, G13.1, 2-35

G13.1, C-198

G17, C-197

G18, C-197

G19, C-197

G20, C-195

G20, G21, 3-70

G21, C-195

G22, C-196

G23, C-196

G27, 2-49, C-197

G28, 2-47, C-197

G290, 1-12, C-198

G291, 1-12, C-198

G30, 2-50, C-197

G30.1, C-197

G31, 4-135, C-197

G31, P1-P2, 4-138

G33, 2-38, 2-43, C-195

G34, 2-46, C-195

G40, C-196

G40, G41/G42, 3-73

G41, C-196

G42, C-196

G50.2, C-197

G51.2, C-197

G52, C-197

G53, 3-56, C-197

G54, C-196

G54 P{1...48}, C-196

G54 P0, C-196

G55, C-196

G56, C-196

G57, C-196

G58, C-196

G59, C-196

G60, C-197

G65, C-197

G65, G66, G67, 4-139

G66, C-196

G67, C-196

G68, C-195

G69, C-195

G70, C-197

G70-G76, 4-117

G71, 4-102, C-197

G72, 4-105, C-197

G73, C-197

G74, C-197

G75, 4-113, C-197

G76, 4-114, C-197

G77, C-195

G78, C-195

G79, C-195

G80, C-196

G80 to G89, 4-118

G83, 4-121, C-196

G83 or G87, 4-124

G83, G87, 4-122, 4-123

G84, 4-126, C-196

G85, 4-127, C-196

G87, 4-121, C-196

G88, 4-126, C-196

G89, 4-127, C-196

G90, C-195

G91, C-195

G92, 3-57, C-197

G92.1, 3-57, C-197

G94, 1-21, C-195  
G94/G95, 1-21  
G95, 1-17, 1-21, C-195  
G96, C-195  
G96, G97, 3-81  
G97, C-195  
G98, C-196  
G98/G99, 4-120  
G99, C-196

**M**

M function, 3-85  
M96, 4-152  
M97, 4-152

**S**

S command, 3-80



# Index

## A

Absolute/incremental designation, 3-66  
 Alarms, F-229  
 Argument specification, 4-143  
 Automatic return to reference point, 2-47  
 Automating support functions, 4-135

## B

Backlash compensation, 3-71  
 Base coordinate system, 3-56, 3-57

## C

Canned cycles, 4-89  
 Circular interpolation, 2-27  
 Circular interpolation with R designation, 2-30  
 Comments, 1-15  
 compressor function, 4-150  
 Constant surface speed control, 3-81  
 Continuous thread cutting, 2-38  
 Control point, 3-74  
 Coordinate system, 3-55  
 Cutting cycle A, 4-91  
 Cutting feed, 1-17  
 Cylindrical interpolation, 2-33

## D

Decimal point, 1-13  
 Designation of multiple M codes in a single block, 3-87  
 Determining the coordinate value input modes, 3-66  
 Diametric and radial commands for X-axis, 3-69  
 DryRun mode, 4-151  
 Dwell, 3-72

## E

Eight-digit program number, 4-134  
 Error messages, F-229

## F

F command, 1-17

Feed per minute mode, 1-21  
 Feed per revolution mode, 1-17, 1-21

## G

G code display, 1-12  
 G Code system A, 1-13  
 G code table, C-195  
 General purpose M codes, 3-86

## H

HMI, A-158  
 Hole-machining canned cycles, 4-118

## I

Inch/metric input designation, 3-70  
 Internally processed M codes, 3-85  
 ISO Dialect mode, 1-11

## L

Linear interpolation, 2-25

## M

M codes relating to stop operation, 3-84  
 M function, 3-84  
 Machine data  
   channel-specific, D-212  
   list, E-223  
 Machining cycle, Interrupt program, 4-153  
 Macroprogram call, 4-139, 4-148  
 Macroprograms, 4-139, 4-147  
 Maximum programmable values for axis movement, 1-12  
 Miscellaneous function, 3-84  
 MMC, A-160  
 Modal call up, 4-143  
 Multiple repetitive cycles, 4-101  
 Multiple-thread cutting, 2-43

## N

Nose R offset function, 3-73

**O**

Optional block skip, 1-16

**P**

Pattern repeat cycle, 4-112, 4-113, 4-114, 4-117

Polar coordinate interpolation, 2-35

Positioning, 2-23

Positioning in the error detect ON mode, 2-23

Program support functions, 4-89, 4-129

Programmable data input, 4-129

**R**

Rapid lift, 2-51

Rapid traverse, 1-17

Reference point return, 2-47

Reference point return check, 2-49

**S**

S function, 3-80

S5-digit command, 3-80

Second to fourth reference point return, 2-50

Setting data

axis-specific, D-219

channel-specific, D-220

list, E-226

Siemens mode, 1-11

Simple call up, 4-140

Skip function, 4-135

Skip level, 1-16, 4-151

Spindle command, 3-80

Spindle function, 3-80

STOPRE, 3-60

Straight facing cycle, 4-98

Subprogram call up function, 4-130

Subprograms, 4-139

Switching between feed per minute mode and feed per revolution mode, 1-21

Switchover, 1-12

**T**

T function, 3-84

Thread cutting, 2-38

Thread cutting cycle, 4-93, 4-102, 4-105, 4-106, 4-108

Thread cutting function, 2-38

Time-controlling commands, 3-72

Tool function, 3-84

Tool life control function, 4-138

Tool offset data memory, 3-73

Tool offset functions, 3-73

Tool position offset, 3-73

**V**

Variable lead thread cutting, 2-46

To  
SIEMENS AG  
A&D MC MS  
P.O. Box 3180  
D-91050 Erlangen, Germany  
Fax: +49-(0)9131 / 98-63315 [Documentation]  
mailto:documotioncontrol.@siemens.com  
http://www.siemens.com/automation/service&support

<b>From</b>	<b>Suggestions</b> <b>Corrections</b>  for Publication/Manual:  SINUMERIK 802D sl/840D sl/840D /840Di sl/840Di/810D Programming Manual ISO Turning  User Documentation
Name _____ Address of your company/department _____ Street _____ _____ Telephone: _____ / _____ Telefax: _____ / _____	Programming Guide  Edition: 04.2007  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**