

*Changes for the Better*

**MITSUBISHI CNC**

**EZMotion-NC E60/E68 Series**

**PROGRAMMING MANUAL  
(Lathe System)**






EZMotion and MELDAS are registered trademarks of Mitsubishi Electric Corporation. Other company and product names that appear in this manual are trademarks or registered trademarks of the respective companies.

## Introduction

This manual is a guide for using the EZMotion-NC E60/E68. Programming is described in this manual, so read this manual thoroughly before starting programming. Thoroughly study the "Precautions for Safety" on the following page to ensure safe use of this NC unit.

### Details described in this manual

#### CAUTION

-  For items described as "Restrictions" or "Usable State" in this manual, the instruction manual issued by the machine tool builder takes precedence over this manual.
-  Items not described in this manual must be interpreted as "not possible".
-  This manual is written on the assumption that all option functions are added. Refer to the specifications issued by the machine tool builder before starting use.
-  Refer to the Instruction Manual issued by each machine tool builder for details on each machine tool.
-  Some screens and functions may differ depending on the NC system (or its version), and some functions may not be possible. Please confirm the specifications before use.

### General precautions

- (1) Refer to the following documents for details on handling  
Operation Manual ..... IB-1500172

## Precautions for Safety

Always read the specifications issued by machine tool builder, this manual, related manuals and attached documents before installation, operation, programming, maintenance or inspection to ensure correct use.

Understand this numerical controller, safety items and cautions before using the unit.

This manual ranks the safety precautions into "**DANGER**", "**WARNING**" and "**CAUTION**".



When the user may be subject to imminent fatalities or major injuries if handling is mistaken.



When the user may be subject to fatalities or major injuries if handling is mistaken.



When the user may be subject to injuries or when physical damage may occur if handling is mistaken.

Note that even items ranked as "**CAUTION**", may lead to major results depending on the situation. In any case, important information that must always be observed is described.

### **DANGER**






Not applicable in this manual.

### **WARNING**

Not applicable in this manual.






### **CAUTION**

#### 1. Items related to product and manual





-  For items described as "Restrictions" or "Usable State" in this manual, the instruction manual issued by the machine tool builder takes precedence over this manual.
-  Items not described in this manual must be interpreted as "not possible".
-  This manual is written on the assumption that all option functions are added. Refer to the specifications issued by the machine tool builder before starting use.
-  Refer to the Instruction Manual issued by each machine tool builder for details on each machine tool.
-  Some screens and functions may differ depending on the NC system (or its version), and some functions may not be possible. Please confirm the specifications before use.

## CAUTION

### 2. Items related to operation

-  Before starting actual machining, always carry out dry run operation to confirm the machining program, tool offset amount and workpiece offset amount, etc.
-  If the workpiece coordinate system offset amount is changed during single block stop, the new setting will be valid from the next block.
-  Turn the mirror image ON and OFF at the mirror image center.
-  If the tool offset amount is changed during automatic operation (including during single block stop), it will be validated from the next block or blocks onwards.
-  Do not make the synchronous spindle rotation command OFF with one workpiece chucked by the basic spindle and synchronous spindle during the spindle synchronous control mode.  
Failure to observe this may cause the synchronous spindle stop, and hazardous situation.

### 3. Items related to programming

-  The commands with "no value after G" will be handled as "G00".
-  ";", "EOB", "%", and "EOR" are symbols used for explanation. The actual codes for ISO are "CR, LF" ("LF") and "%".  
The programs created on the Edit screen are stored in the NC memory in a "CR, LF" format, however, the programs created with external devices such as the FLD or RS-232C may be stored in an "LF" format.  
The actual codes for EIA are "EOB (End of Block)" and "EOR (End of Record)".
-  When creating the machining program, select the appropriate machining conditions, and make sure that the performance, capacity and limits of the machine and NC are not exceeded. The examples do not consider the machining conditions.
-  Do not change fixed cycle programs without the prior approval of the machine tool builder.

# CONTENTS

<b>1. Control Axes</b> .....	<b>1</b>
1.1 Coordinate word and control axis .....	1
1.2 Coordinate systems and coordinate zero point symbols .....	2
<b>2. Input Command Units</b> .....	<b>3</b>
2.1 Input command units .....	3
2.2 Input setting units .....	3
<b>3. Data Formats</b> .....	<b>5</b>
3.1 Tape codes .....	5
3.2 Program formats .....	8
3.3 Program address check function .....	11
3.4 Tape memory format .....	12
3.5 Optional block skip ("/") .....	12
3.6 Program/sequence/block numbers; O, N .....	13
3.7 Parity H/V .....	14
3.8 G code lists .....	15
3.9 Precautions before starting machining .....	19
<b>4. Buffer Register</b> .....	<b>20</b>
4.1 Input buffer .....	20
4.2 Pre-read buffers .....	21
<b>5. Position Commands</b> .....	<b>22</b>
5.1 Incremental/absolute value commands .....	22
5.2 Radius/diameter commands .....	23
5.3 Inch/metric conversion; G20, G21 .....	24
5.4 Decimal point input .....	28
<b>6. Interpolation Functions</b> .....	<b>32</b>
6.1 Positioning (Rapid Traverse); G00 .....	32
6.2 Linear interpolation; G01 .....	38
6.3 Circular Interpolation; G02, G03 .....	41
6.4 R specification circular interpolation; G02, G03 .....	45
6.5 Plane selection; G17, G18, G19 .....	47
6.6 Thread cutting .....	49
6.6.1 Constant lead thread cutting; G33 .....	49
6.6.2 Inch thread cutting; G33 .....	53
6.6.3 Continuous thread cutting .....	54
6.6.4 Variable Lead Thread Cutting; G34 .....	55
6.7 Helical Interpolation; G17, G18, G19, and G02, G03 .....	58
6.8 Cylindrical Interpolation; G07.1 (6 and 7 Only in G Code System) .....	61
6.9 Pole Coordinate Interpolation; G12.1 (Only 6, 7 in G Code System) .....	69
<b>7. Feed Functions</b> .....	<b>76</b>
7.1 Rapid traverse rate .....	76
7.2 Cutting feedrate .....	76
7.3 F1-digit feed .....	77
7.4 Synchronous/asynchronous feed; G94, G95 .....	78
7.5 Feedrate designation and effects on control axes .....	80
7.6 Thread cutting mode .....	84
7.7 Automatic acceleration/deceleration .....	85
7.8 Speed clamp .....	85
7.9 Exact stop check; G09 .....	86
7.10 Exact stop check mode ; G61 .....	89
7.11 Deceleration check .....	90

7.11.1 G1 → G0 deceleration check .....	92
7.11.2 G1 → G1 deceleration check .....	93
7.12 Automatic corner override ; G62.....	94
7.13 Tapping mode ; G63.....	99
7.14 Cutting mode ; G64 .....	99
<b>8. Dwell .....</b>	<b>100</b>
8.1 Per-second Dwell ; G04.....	100
<b>9. Miscellaneous Functions .....</b>	<b>103</b>
9.1 Miscellaneous functions (M8-digits BCD).....	103
9.2 2nd miscellaneous functions (A8-digits, B8-digits or C8-digits).....	106
9.3 Index Table Indexing .....	107
<b>10. Spindle Functions.....</b>	<b>109</b>
10.1 Spindle functions (S function).....	109
10.1.1 Spindle functions (S6-digits Analog) .....	109
10.1.2 Spindle functions (S8-digits).....	109
10.2 Constant surface speed control; G96, G97 .....	110
10.3 Spindle clamp speed setting; G92.....	111
10.4 Spindle/C Axis Control.....	112
10.5 Spindle Synchronization Control .....	115
10.5.1 Spindle Synchronization Control I .....	116
10.5.2 Spindle Synchronization Control II .....	122
10.5.3 Precautions for Using Spindle Synchronization Control.....	127
10.6 Multiple Spindle Control.....	129
10.6.1 Multiple Spindle Control I (Multiple Spindle Command) .....	130
10.6.2 Multiple Spindle Control I (Spindle Selection Command).....	131
10.6.3 Multiple Spindle Control II .....	133
<b>11. Tool Functions .....</b>	<b>135</b>
11.1 Tool functions (T8-digits BCD) .....	135
<b>12. Tool Offset Functions.....</b>	<b>136</b>
12.1 Tool offset.....	136
12.1.1 Tool offset start.....	137
12.1.2 Expanded method at starting tool offset.....	138
12.2 Tool length offset .....	140
12.3 Tool nose wear offset .....	142
12.4 Tool nose radius compensation; G40, G41, G42, G46 .....	143
12.4.1 Tool nose point and compensation directions .....	145
12.4.2 Tool nose radius compensation operations.....	148
12.4.3 Other operations during tool nose radius compensation .....	166
12.4.4 G41/G42 commands and I, J, K designation.....	174
12.4.5 Interrupts during tool nose radius compensation .....	179
12.4.6 General precautions for tool nose radius compensation .....	181
12.4.7 Interference check.....	182
12.5 Programmed offset input; G10, G11.....	187
12.6 Tool life management II .....	190
12.6.1 Counting the tool life.....	193
<b>13. Program Support Functions .....</b>	<b>195</b>
13.1 Fixed cycles for turning.....	195
13.1.1 Longitudinal cutting cycle; G77 .....	196
13.1.2 Thread cutting cycle; G78 .....	198
13.1.3 Face cutting cycle; G79.....	201
13.2 Fixed Cycle for Turning (MELDAS Special Format) .....	204
13.3 Compound Type Fixed Cycle for Turning Machining .....	205
13.3.1 Longitudinal rough cutting cycle; G71 .....	206

13.3.2	Face rough cutting cycle; G72.....	220
13.3.3	Formed material rough cutting cycle; G73 .....	222
13.3.4	Finishing cycle; G70.....	226
13.3.5	Face cut-off cycle; G74 .....	227
13.3.6	Longitudinal cut-off cycle; G75.....	229
13.3.7	Compound thread cutting cycle; G76.....	231
13.3.8	Precautions for Compound Type Fixed Cycle for Turning Machining; G70 ~ G76 .....	235
13.4	Compound Fixed Cycle (MELDAS Special Format).....	238
13.5	Hole drilling fixed cycles; G80 ~ G89.....	242
13.5.1	Face deep hole drilling cycle 1; G83 (Longitudinal deep hole drilling cycle 1; G87) ....	250
13.5.2	Face tapping cycle; G84 (Longitudinal tapping cycle; G88) / Face reverse tapping cycle; G84.1 (Longitudinal reverse tapping cycle; G88.1) .....	251
13.5.3	Face boring cycle; G85 (Longitudinal boring cycle; G89).....	255
13.5.4	Deep hole drilling cycle 2; G83.2.....	255
13.5.5	Hole drilling fixed cycle cancel; G80.....	258
13.5.6	Precautions when using a hole drilling fixed cycle .....	258
13.6	Hole Drilling Fixed Cycle; G80~G89 (MELDAS Special Format) .....	260
13.6.1	Initial Point And R Point Level Return; G98, G99.....	279
13.6.2	Setting of Workpiece Coordinates in Fixed Cycle Mode .....	280
13.7	Subprogram control; M98, M99.....	281
13.7.1	Calling subprogram with M98 and M99 commands .....	281
13.8	Variable commands.....	286
13.9	User macro.....	288
13.9.1	User macro commands; G65, G66, G66.1, G67.....	288
13.9.2	Macro call instruction.....	289
13.9.3	ASCII code macro .....	297
13.9.4	Variables .....	302
13.9.5	Types of variables .....	304
13.9.6	Operation commands.....	339
13.9.7	Control commands .....	345
13.9.8	External output commands.....	348
13.9.9	Precautions .....	350
13.10	Corner chamfering/corner rounding I.....	352
13.10.1	Corner chamfering ",C" (or "I_", "K_", "C_") .....	352
13.10.2	Corner rounding ",R_" (or "R_") .....	354
13.10.3	Corner chamfering/corner rounding expansion.....	356
13.10.4	Interrupt during corner chamfering/corner rounding.....	358
13.11	Corner chamfering/corner rounding II.....	360
13.11.1	Corner chamfering ",C_" (or "I_", "K_", "C_") .....	360
13.11.2	Corner rounding ",R_" (or "R_") .....	363
13.11.3	Corner chamfering/corner rounding expansion.....	364
13.11.4	Interrupt during corner chamfering/corner rounding.....	364
13.12	Linear Angle Command.....	365
13.13	Geometric.....	366
13.13.1	Geometric I.....	366
13.13.2	Geometric IB .....	369
13.14	Program parameter input; G10, G11 .....	383
13.15	Macro interrupt .....	384
13.16	Tool change position return; G30.1 ~ G30.5 .....	392
<b>14</b>	<b>Coordinate System Setting Functions.....</b>	<b>395</b>
14.1	Coordinate words and control axes .....	395
14.2	Basic machine, workpiece and local coordinate systems.....	396
14.3	Machine zero point and 2nd reference point (Zero point).....	397
14.4	Automatic coordinate system setting.....	398
14.5	Machine coordinate system selection; G53.....	399
14.6	Coordinate system setting; G92 .....	400



14.7 Reference point (Zero point) return; G28, G29.....	401
14.8 2nd, 3rd, and 4th reference point (Zero point) return; G30.....	405
14.9 Simple zero point return .....	408
14.10 Reference point check; G27 .....	413
14.11 Workpiece coordinate system setting and offset; G54 ~ G59 (G54.1) .....	414
14.12 Local coordinate system setting; G52 .....	420
14.13 Workpiece Coordinate System Preset; G92.1 .....	421
14.14 Coordinate system for rotary axis.....	426
<b>15. Protection Function .....</b>	<b>429</b>
15.1 Chuck barrier/tailstock barrier; G22, G23 .....	429
15.2 Stored stroke limit (6 and 7 only in G code list) .....	434
<b>16. Measurement Support Functions.....</b>	<b>436</b>
16.1 Automatic tool length measurement; G37 .....	436
16.2 Skip function; G31 .....	440
16.3 Multi-stage skip function; G31.n, G04 .....	446
16.4 Multi-stage skip function 2 .....	448
16.5 Programmable current limitation.....	450
<b>Appendix 1. Program Parameter Input N No. Correspondence Table.....</b>	<b>451</b>
<b>Appendix 2. Program Error .....</b>	<b>454</b>

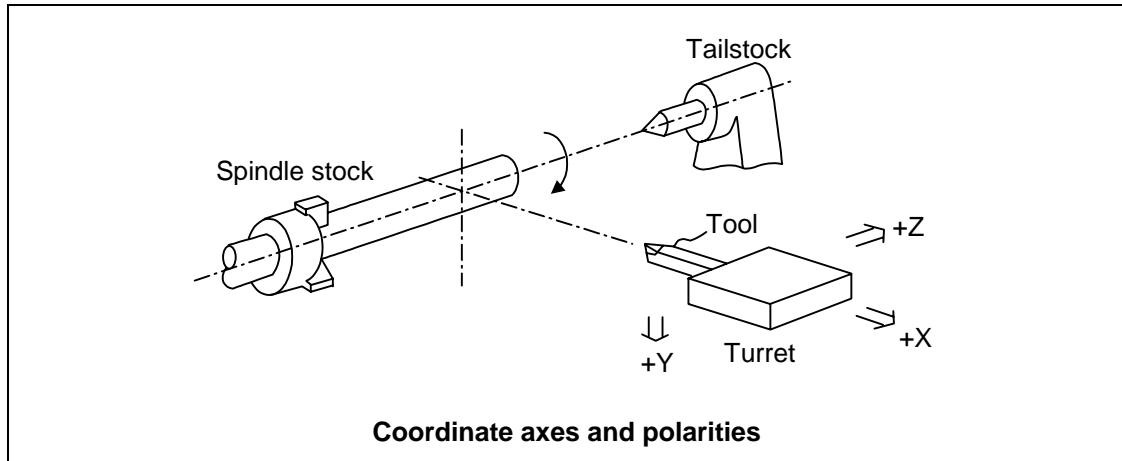
### 1. Control Axes

#### 1.1 Coordinate word and control axis

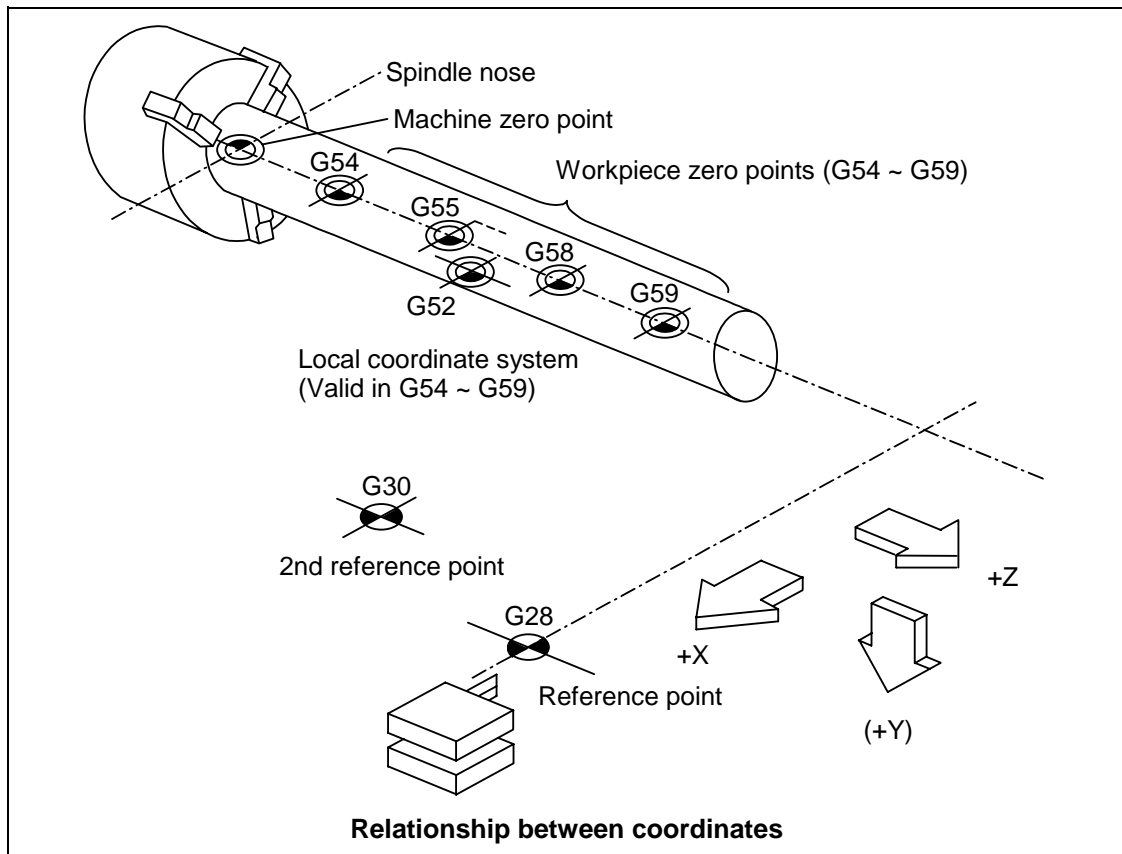


#### Function and purpose

In the case of a lathe, the axis parallel to the spindle is known as the Z axis and its forward direction is the direction in which the turret moves away from the spindle stock while the axis at right angles to the Z axis is the X axis and its forward direction is the direction in which it moves away from the Z axis, as shown in the figure below.



Since coordinates based on the right hand rule are used with a lathe, the forward direction of the Y axis in the above figure which is at right angles to the X-Z plane is downward. It should be borne in mind that an arc on the X-Z plane is expressed as clockwise or counterclockwise as seen from the forward direction of the Y axis. (Refer to the section on circular interpolation.)



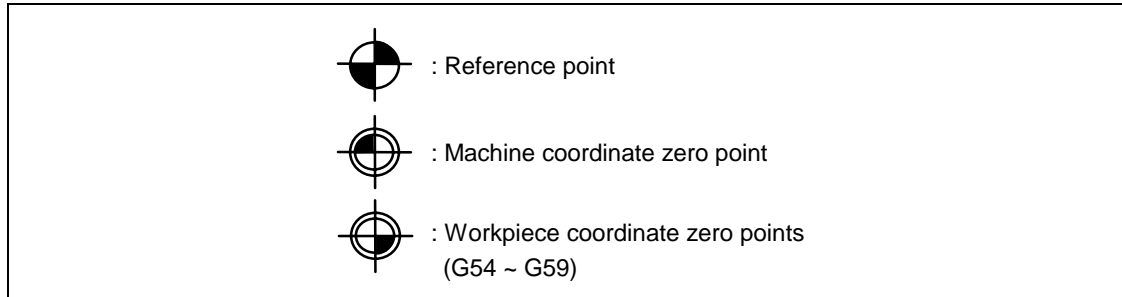
# 1. Control Axes

## 1.2 Coordinate systems and coordinate zero point symbols

### 1.2 Coordinate systems and coordinate zero point symbols

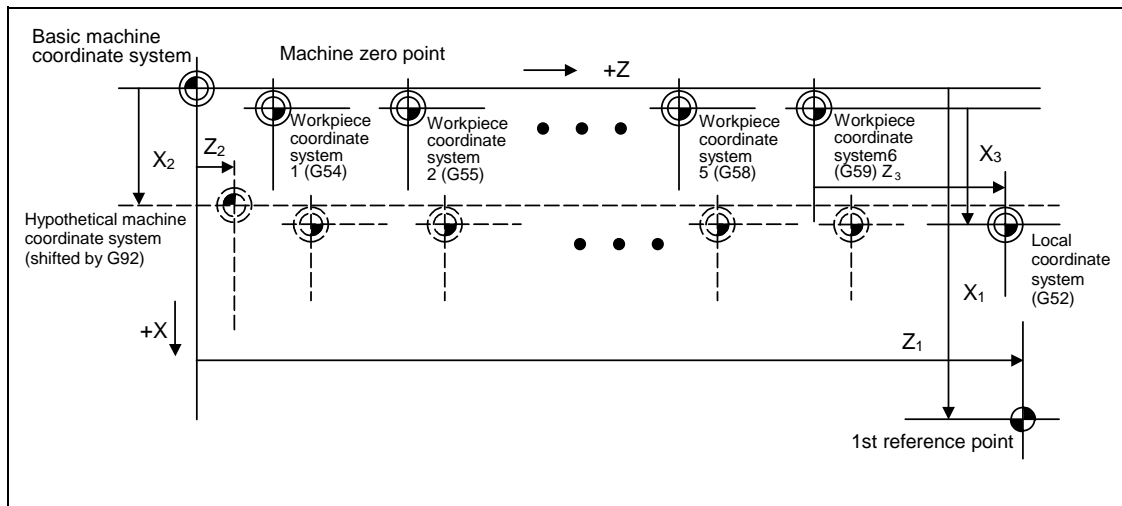


#### Function and purpose



Upon completion of the reference point return, the parameters are referred to and automatically set for the basic machine coordinate system and workpiece coordinate systems (G54 ~ G59).

The basic machine coordinate system is set so that the first reference point is at the position designated by the parameter from the basic machine coordinate zero point (machine zero point).



The local coordinate system (G52) is valid on the coordinate systems designated by the commands for the workpiece coordinate systems 1 to 6.

Using the G92 command, the basic machine coordinate system can be shifted and made the hypothetical machine coordinate system. At the same time, workpiece coordinate systems 1 to 6 are also shifted.

## 2. Input Command Units

### 2.1 Input command units

## 2. Input Command Units

### 2.1 Input command units



#### Function and purpose

These are the units used for the movement amounts in the program as commanded by the MDI input and command tape. They are expressed in millimeters, inches or degrees (°).

### 2.2 Input setting units



#### Function and purpose

These are the units of setting data which are used, as with the compensation amounts, in common for all axes.

The input setting units and input command units can be selected from the following types for each axis with the parameters. (For further details on settings, refer to the Operation Manual.)

(1) When using E60

	Input unit parameters	Linear axis				Rotation axis (°)
		Millimeter		Inch		
		Diameter command	Radius command	Diameter command	Radius command	
Input command unit	#1015 cunit = 10	0.001	0.001	0.0001	0.0001	0.001
Min. movement unit	#1003 iunit = B	0.0005	0.001	0.00005	0.0001	0.001
Input setting unit	#1003 iunit = B	0.001	0.001	0.0001	0.0001	0.001

(2) When using E68

	Input unit parameters	Linear axis				Rotation axis (°)
		Millimeter		Inch		
		Diameter command	Radius command	Diameter command	Radius command	
Input command unit	#1015 cunit = 10	0.001	0.001	0.0001	0.0001	0.001
	= 1	0.0001	0.0001	0.00001	0.00001	0.0001
Min. movement unit	#1003 iunit = B	0.0005	0.001	0.00005	0.0001	0.001
	= C	0.00005	0.0001	0.000005	0.00001	0.0001
Input setting unit	#1003 iunit = B	0.001	0.001	0.0001	0.0001	0.001
	= C	0.0001	0.0001	0.00001	0.00001	0.0001

## 2. Input Command Units

### 2.2 Input setting units

**(Note 1)** Inch/metric conversion is performed in either of 2 ways: conversion from the parameter screen (#1041 I\_inch: valid only when the power is turned ON) and conversion using the G command (G20 or G21).

However, when a G command is used for the conversion, the conversion applies only to the input command units and not to the input setting units.

Consequently, the tool offset amounts and other compensation amounts as well as the variable data should be preset to correspond to inches or millimeters.

**(Note 2)** The millimeter and inch systems cannot be used together.

**(Note 3)** During circular interpolation on an axis where the input command units are different, the center command (I, J, K) and the radius command (R) can be designated by the input setting units. (Use a decimal point to avoid confusion.)



#### Detailed description

##### (1) Units of various data

These input setting units determine the parameter setting unit, program command unit and the external interface unit for the PLC axis, DDB and handle pulse, etc. The following rules show how the unit of each data changes when the input setting unit is changed. This table applies to the NC axis and PLC axis.

Data	Metric /inch	Setting value	Input setting unit		Remarks
			B (0.001mm)	C (0.0001mm)	
Speed data Example: rapid	Metric	20000 (mm/min)	20000	200000	(Note 1)
		Setting range	1 to 999999	1 to 999999	
	Inch	2000 (inch/min)	2000	20000	
		Setting range	1 to 999999	1 to 999999	
Position data Example: SoftLimit+	Metric	123.123 (mm)	123.123	123.1230	
		Setting range	±99999.999	±9999.9999	
	Inch	12.1234 (inch)	12.1234	12.1234	
		Setting range	±9999.9999	±999.99999	
Interpolation unit data	Metric	10 (μm)	20	20	
		Setting range	±9999	±9999	
	Inch	0.01 (inch)	20	20	
		Setting range	±9999	±9999	

**(Note 1)** Set the data with a mm/min or inch/min unit for the input setting unit B (0.001). When using the E68, input setting unit C (0.0001) can be set also. When setting C, set a value ten times the B value.

##### (2) Program command

The program command unit follows the above table.

If the data has a decimal point, the number of digits in the integer section will decrease and the number of digits in the decimal point section will increase as the input setting unit becomes smaller.

When setting data with no decimal point, and which is a position command, the data will be affected by the input setting unit and input command unit.

For the feedrate, as the input setting unit becomes smaller, the number of digits in the integer section will remain the same, but the number of digits in the decimal point section will increase

## 3. Data Formats

### 3.1 Tape codes



#### Function and purpose

The tape command codes used for this controller are combinations of alphabet letters (A, B, C, ... Z), numbers (0, 1, 2, ... 9) and signs (+, -, /, ...). These alphabet letters, numbers and signs are referred to as characters. Each character is represented by a combination of 8 holes which may, or may not, be present.


These combinations make up what is called codes.

This controller uses the ISO code (R-840).

**(Note 1)** If a code not given in the "Table of tape codes" is assigned during operation, a program error (P32) will result.

**(Note 2)** For the sake of convenience, a ";" has been used in the CNC display to indicate the End of Block (EOB/LF) which separates one block from another. Do not use the ";" key, however, in actual programming but use the keys in the following table instead.

#### CAUTION

 ";", "EOB", "%", and "EOR" are symbols used for explanation. The actual codes for ISO are "CR, LF" ("LF") and "%".

The programs created on the Edit screen are stored in the NC memory in a "CR, LF" format, however, the programs created with external devices such as the FLD or RS-232C may be stored in an "LF" format.

The actual codes for EIA are "EOB (End of Block)" and "EOR (End of Record)".



#### Detailed description

(1) Use the keys in the following table for programming.

Key used	Code used	ISO	Screen display
End of Block		LF or NL	;
End of Record		%	%

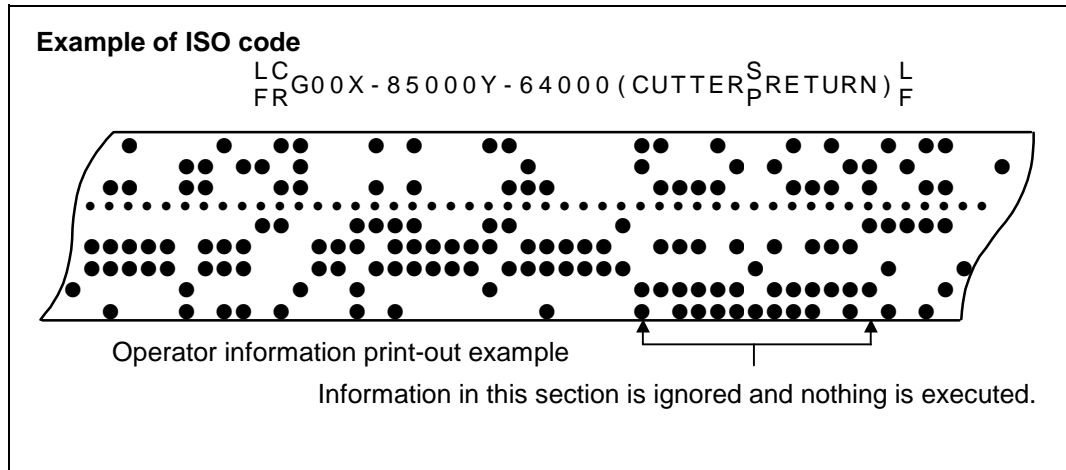
(2) **Significant data section (label skip function)**

All data up to the first EOB (;), after the power has been turned ON or after operation has been reset, are ignored during automatic operation based on tape, memory loading operation or during a search operation. In other words, the significant data section of a tape extends from the character or number code after the initial EOB (;) code after resetting to the point where the reset command is issued.

**(3) Control out, control in**

When the ISO code is used, all data between control out "(" and control in ")" or ";" are ignored, although these data appear on the setting and display unit. Consequently, the command tape name, number and other such data not directly related to control can be inserted in this section.

This information (except (B) in the "Table of tape codes") will also be loaded, however, during tape loading. The system is set to the "control in" mode when the power is turned ON.

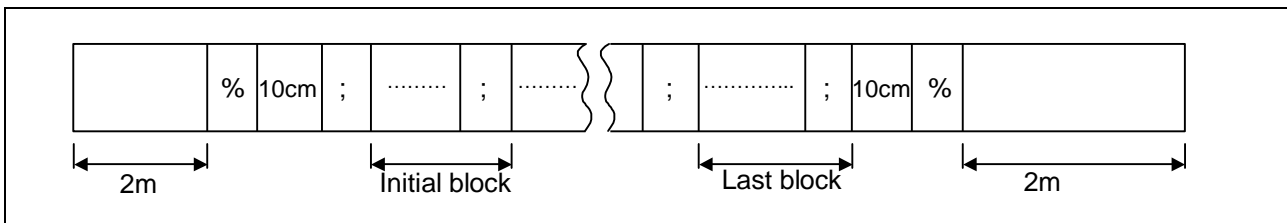


**(4) EOR (%) code**

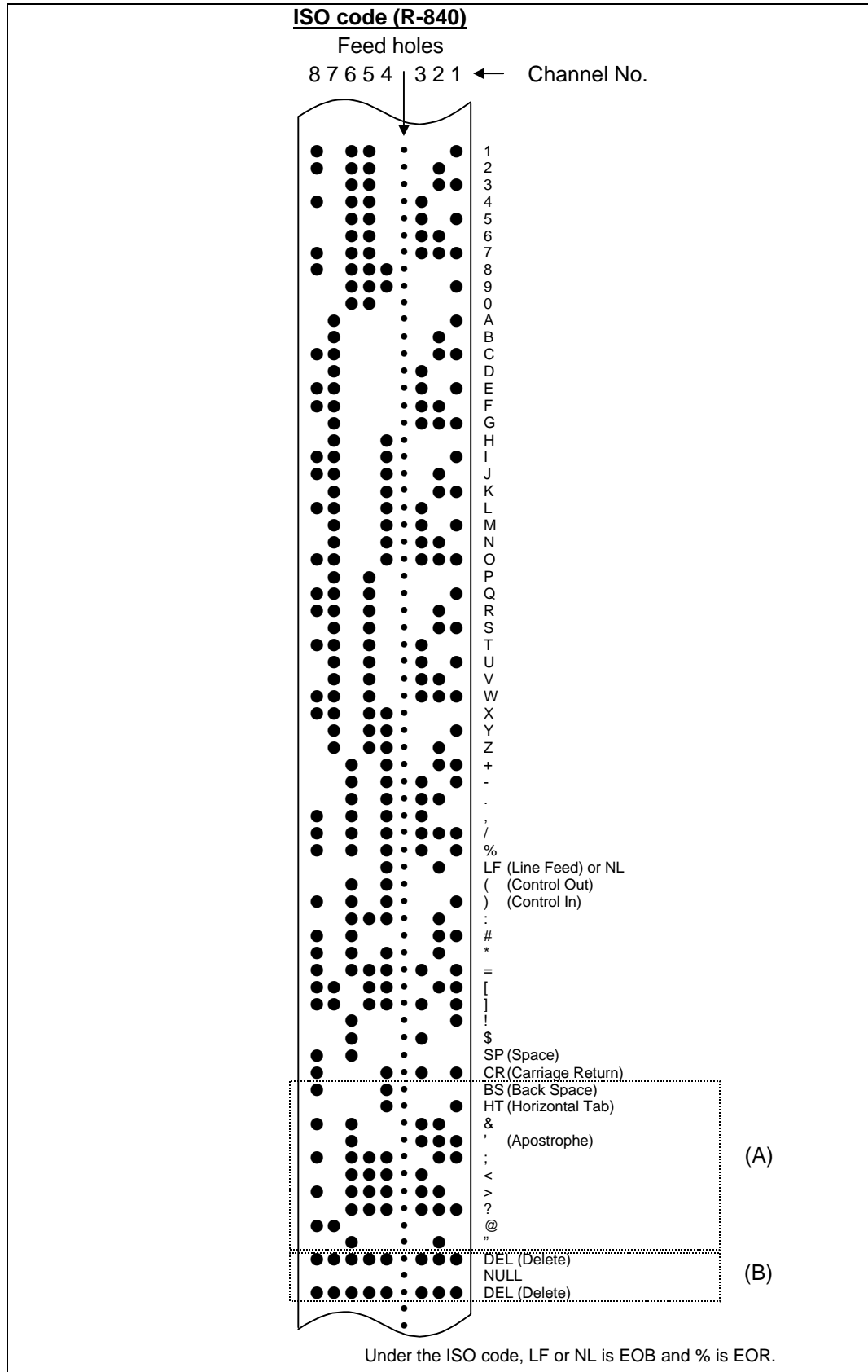
Generally, the End of Record code is punched at both ends of the tape. It has the following functions:

- (a) Rewind stop when rewinding tape (with tape rewinder)
- (b) Rewind start during tape search (with tape rewinder)
- (c) Completion of loading during tape loading into memory

**(5) Tape preparation for tape operation (with tape rewinder)**



If a tape rewinder is not used, there is no need for the 2-meter dummy at both ends of the tape and for the head EOR (%) code.



The (A) codes are stored on tape but an error results (except when they are used in the comment section) during operation.

The (B) codes are non-working codes and are always ignored. (Parity V check is not executed.)

Table of tape codes



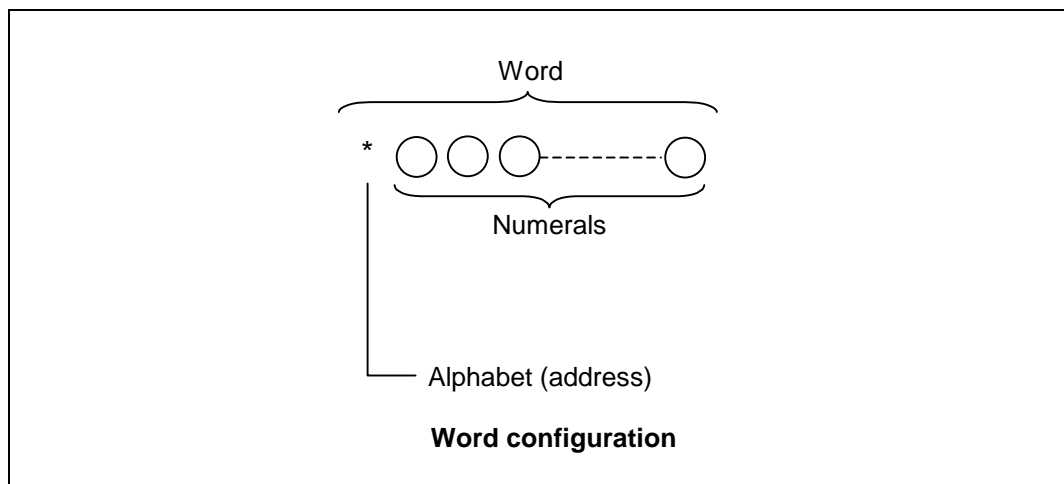
## 3.2 Program formats

**Function and purpose**

The prescribed arrangement used when assigning control information to the controller is known as the program format, and the format used with this controller is called the "word address format".

**Detailed description****(1) Word and address**

A word is a collection of characters arranged in a specific sequence. This entity is used as the unit for processing data and for causing the machine to execute specific operations. Each word used for this controller consists of an alphabet letter and a number of several digits (sometimes with a "-" sign placed at the head of the number).



The alphabet letter at the head of the word is the address. It defines the meaning of the numerical information which follows it.

For details of the types of words and the number of significant digits of words used for this controller, refer to "Format details".

**(2) Blocks**

A block is a collection of words. It includes the information which is required for the machine to execute specific operations. One block unit constitutes a complete command. The end of each block is marked with an EOB (End of Block) code.

**(3) Programs**

A program is a collection of several blocks.

### 3. Data Formats

### 3.2 Program formats

	Metric command	Inch command	Rotary axis (Metric command)	Rotary axis (Inch command)
Program number	08	←	←	←
Sequence number	N5	←	←	←
Preparatory function	G3/G21	←	←	←
Movement axis	0.001(°) mm/ 0.0001 inch	X+53 Y+53 Z+53 α+53	X+44 Y+44 Z+44 α+44	X+53 Y+53 Z+53 α+53
	0.0001(°) mm/ 0.00001 inch	X+44 Y+44 Z+44 α+44	X+35 Y+35 Z+35 α+35	X+44 Y+44 Z+44 α+44
Arc and cutter radius	0.001(°) mm/ 0.0001 inch	I+53 J+53 K+53 R+53	I+44 J+44 K+44 R+44	I+53 J+53 K+53 R+53
	0.0001(°) mm/ 0.00001 inch	I+44 J+44 K+44 R+44	I+35 J+35 K+35 R+35	I+44 J+44 K+44 R+44
Feed function	0.001(°) mm/ 0.0001 inch	F63(Feed per minute) F43(Feed per revolution)	F44(Feed per minute) F34(Feed per revolution)	F63(Feed per minute) F43(Feed per revolution)
	0.0001(°) mm/ 0.00001 inch	F54(Feed per minute) F34(Feed per revolution)	F35(Feed per minute) F25(Feed per revolution)	F54(Feed per minute) F34(Feed per revolution)
Dwell	X+53/U+53/P8 or X+53/U+53/P+53 (U for only E68)	←	←	←
Tool offset	T1/T2	←	←	←
Miscellaneous function (M)	M8	←	←	←
Spindle function (S)	S8	←	←	←
Tool function (T)	T8	←	←	←
2nd miscellaneous function	A8/B8/C8	←	←	←
Subprogram	P8 H5 L4	←	←	←
Fixed cycle	0.001(°) mm/ 0.0001 inch	R+53 Q53 P8 L4	←	←
	0.0001(°) mm/ 0.00001 inch	R+44 Q44 P8 L4	←	←

#### [Format details]

**(Note 1)** α indicates the additional axis address, such as A, B or C.

**(Note 2)** The No. of digits check for a word is carried out with the maximum number of digits of that address.

**(Note 3)** Numerals can be used without the leading zeros.

**(Note 4)** The meanings of the details are as follows :

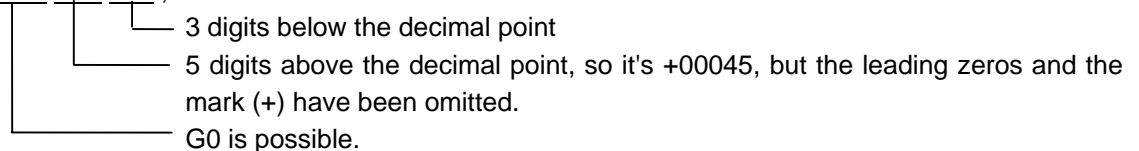
Example 1 : 08 : 8-digit program number

Example 2 : G21 : Dimension G is 2 digits to the left of the decimal point, and 1 digit to the right.

Example 3 : X+53 : Dimension X uses + or - sign and represents 5 digits to the left of the decimal point and 3 digits to the right.

For example, the case for when the X axis is positioned (G00) to the 45.123 mm position in the absolute value (G90) mode is as follows :

G00 X45.123 ;



**(Note 5)** If an arc is commanded using a rotary axis and linear axis while inch commands are being used, the degrees will be converted into 0.1 inches for interpolation.

**(Note 6)** While inch commands are being used, the rotary axis speed will be in increments of 10 degrees. Example: With the F1. (per-minute-feed) command, this will become the 10 degrees/minute command.

**(Note 7)** The decimal places below the decimal point are ignored when a command, such as an S command, with an invalid decimal point has been assigned with a decimal point.

- (Note 8)** This format is the same for the value input from the memory, MDI or setting and display unit.
- (Note 9)** Command the program No. in an independent block. Command the program No. in the head block of the program.
- (Note 10)** Minimum input setting unit "C" (0.0001 (°) mm/0.00001 inch) is a specification of the E68 system. When using E60, the unit cannot be used.

## 3.3 Program address check function



## Function and purpose

The program can be checked in word units when operating machining programs.



## Detailed description

This function enables simple checking of program addresses in word units. If the alphabetic characters are continuous, the program error (P32) will occur.

Availability of this function is selected by the parameter "#1227 aux11/bit4". Note that an error will not occur for the following:

- (1) Reserved words
- (2) Comment statements



## Example of program

## (1) Example of program for address check

**(Example 1)** When there are no numbers following an alphabetic character.  
G28X; → An error will occur. Change to "G28 X0;", etc.

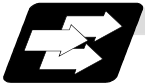
**(Example 2)** When a character string is illegal.  
TEST; → An error will occur. Change to "(TEST);", etc.

## (2) Example of program for word range check

An error (P33) will occur.		Change as shown below.
X-100.-#24	→	X-[100.+#24]
Z+#500*TAN[#1]		Z+[#500*TAN[#1]]
Y-SIN[60.]		Y-SIN[60.]

The numbers of brackets that can be used for one word data are maximum five. If six brackets or more are commanded, the program error (P280) will occur.

#### 3.4 Tape memory format



##### Function and purpose

##### (1) Storage tape and significant sections (ISO, EIA automatic judgment)

Both ISO and EIA tape codes can be stored in the memory in the same way as tape operation. After resetting, ISO/EIA is automatically judged by the EOB code at the head.

The interval to be stored in the memory is from the next character after the head EOB to the EOR code after resetting.

The significant codes listed in the "Table of tape code" in Section 3.1 "Tape codes", in the above significant section are actually stored into the memory. All other codes are ignored and are not stored.

The data between control out "(" and control in ")" are stored into the memory.

#### 3.5 Optional block skip ("/")



##### Function and purpose

This function selectively ignores specific blocks in a machining program which starts with the "/" (slash) code.



##### Detailed description

(1) Provided that the optional block skip switch is ON, blocks starting with the "/" code are ignored. They are executed if the switch is OFF.

Parity check is valid regardless of whether the optional block skip switch is ON or OFF.

When, for instance, all blocks are to be executed for one workpiece but specific block are not to be executed for another workpiece, the same command tape can be used to machine different parts by inserting the "/" code at the head of those specific blocks.



##### Precautions for using optional block skip

(1) Put the "/" code for optional block skip at the beginning of a block. If it is placed inside the block, it is assumed as a user macro, a division instruction.

**(Example)** N20 G1 X25. /Z25. ; ..... NG (User macro, a division instruction;  
a program error results.)

/N20 G1 X25. Z25. ; ..... OK

(2) Parity checks (H and V) are conducted regardless of the optional block skip switch position.

(3) The optional block skip is processed immediately before the pre-read buffer. Consequently, it is not possible to skip up to the block which has been read into the pre-read buffer.

(4) This function is valid even during a sequence number search.

(5) All blocks with the "/" code are also input and output during tape storing and tape output, regardless of the position of the optional block skip switch.

## 3. Data Formats

### 3.6 Program/sequence/block numbers; O, N

#### 3.6 Program/sequence/block numbers; O, N



##### Function and purpose

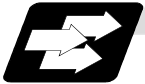
These numbers are used for monitoring the execution of the machining programs and for calling both machining programs and specific stages in machining programs.

- (1) Program numbers are classified by workpiece correspondence or by subprogram units, and they are designated by the address "O" followed by a number with up to 8 digits.
- (2) Sequence numbers are attached where appropriate to command blocks which configure machining programs, and they are designated by the address "N" followed by a number with up to 5 digits.
- (3) Block numbers are automatically provided internally. They are preset to zero every time a program number or sequence number is read, and they are counted up one at a time unless program numbers or sequence numbers are commanded in blocks which are subsequently read.

Consequently, all the blocks of the machining programs given in the table below can be determined without further consideration by combinations of program numbers, sequence numbers and block numbers.

Machining program	Monitor display		
	Program No.	Sequence No.	Block No.
O12345678 (DEMO, PROG) ;	12345678	0	0
N100 G00 G90 X120. Z100. ;	12345678	100	0
G94 S1000 ;	12345678	100	1
N102 G71 P210 Q220 I0.2 K0.2 D0.5 F600 ;	12345678	102	0
N200 G94 S1200 F300 ;	12345678	200	0
N210 G01 X0 Z95. ;	12345678	210	0
G01 X20. ;	12345678	210	1
G03 X50. Z80. K-15. ;	12345678	210	2
G01 Z55. ;	12345678	210	3
G02 X80. Z40. I15. ;	12345678	210	4
G01 X100. ;	12345678	210	5
G01 Z30. ;	12345678	210	6
G02 Z10. K-15. ;	12345678	210	7
N220 G01 Z0 ;	12345678	220	0
N230 G00 X120. Z150. ;	12345678	230	0
N240 M02 ;	12345678	240	0
%	12345678	240	0

## 3.7 Parity H/V



## Function and purpose

Parity check provides a mean of checking whether the tape has been correctly perforated or not. This involves checking for perforated code errors or, in other words, for perforation errors. There are two types of parity check: Parity H and Parity V.

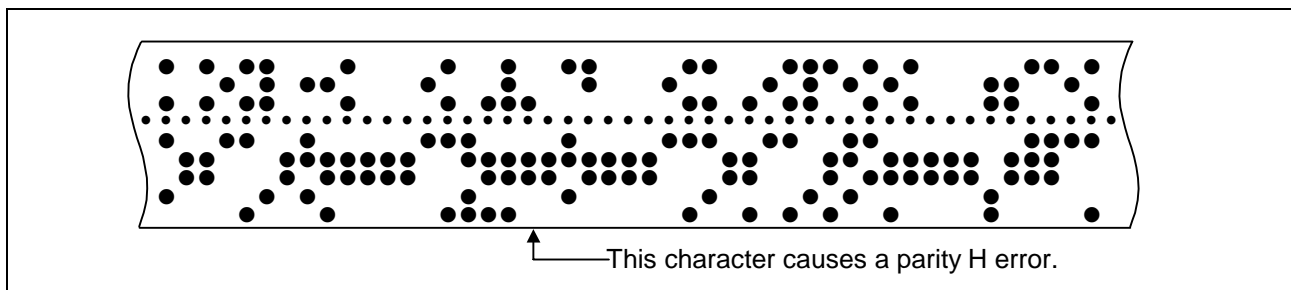
**(1) Parity H**

Parity H checks the number of holes configuring a character and it is done during tape operation, tape input and sequence number search.

A parity H error is caused in the following cases.

**(a) ISO code**

When a code with an odd number of holes in a significant data section has been detected.

**(Example 1) Parity H error example (For ISO codes)**

When a parity H error occurs, the tape stops following the alarm code.

**(2) Parity V**

A parity V check is done during tape operation, tape input and sequence number search when the I/O PARA #9n15 (n is the unit No.1 to 5) parity V check function is set to "1". It is not done during memory operation.

A parity V error occurs in the following case: when the number of codes from the first significant code to the EOB (;) in the significant data section in the vertical direction of the tape is an odd number, that is, when the number of characters in one block is odd.

When a parity V error is detected, the tape stops at the code following the EOB (;).

**(Note 1)** Among the tape codes, there are codes which are counted as characters for parity and codes which are not counted as such. For details, refer to the "Table of tape code" in Section 3.1 "Tape codes".

**(Note 2)** Any space codes which may appear within the section from the initial EOB code to the address code or "/" code are counted for parity V check.

#### 3.8 G code lists



#### Function and purpose

G codes include the four G code lists 2, 3, 6 and 7. One list is selected by setting in parameter "#1037 cmdtyp".

cmdtyp	G code list
3	List 2
4	List 3
7	List 6
8	List 7

G functions are explained using the G code list 3.

**(Note 1)** A program error (P34) will result if a G code that is not in the Table of G code lists is commanded.

**(Note 2)** An alarm will result if a G code without additional specifications is commanded.

**Table of G code lists**

G code list				Group	Function	E60	E68
2	3	6	7				
ΔG00	ΔG00	ΔG00	ΔG00	01	Positioning	○	○
ΔG01	ΔG01	ΔG01	ΔG01	01	Linear interpolation	○	○
G02	G02	G02	G02	01	Circular interpolation CW	○	○
G03	G03	G03	G03	01	Circular interpolation CCW	○	○
G04	G04	G04	G04	00	Dwell	○	○
		G07.1/ G107	G07.1/ G107	19	Cylindrical interpolation	-	○
G09	G09	G09	G09	00	Exact stop check	○	○
G10	G10	G10	G10	00	Program parameter/Compensation input	○	○
G11	G11	G11	G11	00	Program parameter input mode cancel	○	○
		G12.1/ G112	G12.1/ G112	19	Pole coordinate interpolation ON	-	○
		G13.1/ G113	G13.1/ G113	19	Pole coordinate interpolation cancel	-	○
ΔG17	ΔG17	ΔG17	ΔG17	02	Plane selection X-Y	○	○
ΔG18	ΔG18	ΔG18	ΔG18	02	Plane selection Z-X	○	○
ΔG19	ΔG19	ΔG19	ΔG19	02	Plane selection Y-Z	○	○
ΔG20	ΔG20	ΔG20	ΔG20	06	Inch command	○	○
ΔG21	ΔG21	ΔG21	ΔG21	06	Metric command	○	○
G22	G22			04	Barrier check ON	○	○
*G23	*G23			04	Barrier check OFF	○	○
		G22	G22	00	Soft limit ON	○	○
		G23	G23	00	Soft limit OFF	○	○



### 3. Data Formats

### 3.8 G code lists

G code list				Group	Function	E60	E68
2	3	6	7				
G27	G27	G27	G27	00	Reference point return check	○	○
G28	G28	G28	G28	00	Automatic reference point return	○	○
G29	G29	G29	G29	00	Return from reference point	○	○
G30	G30	G30	G30	00	2nd, 3rd and 4th reference point return	○	○
G30.1	G30.1	G30.1	G30.1	00	Tool change position return 1	○	○
G30.2	G30.2			00	Tool change position return 2	○	○
G30.3	G30.3			00	Tool change position return 3	○	○
G30.4	G30.4			00	Tool change position return 4	○	○
G30.5	G30.5			00	Tool change position return 5	○	○
G31	G31	G31	G31	00	Skip function / Multiple-step skip function 2	○	○
G31.1	G31.1	G31.1	G31.1	00	Multi-step skip function 1-1	○	○
G31.2	G31.2	G31.2	G31.2	00	Multi-step skip function 1-2	○	○
G31.3	G31.3	G31.3	G31.3	00	Multi-step skip function 1-3	○	○
G32	G33	G32	G33	01	Thread cutting	○	○
G34	G34	G34	G34	01	Variable lead thread cutting	○	○
G37	G37	G36/G37	G36/G37	00	Automatic tool length measurement	○	○
*G40	*G40	*G40	*G40	07	Tool nose R compensation cancel	○	○
G41	G41	G41	G41	07	Tool nose R compensation left	○	○
G42	G42	G42	G42	07	Tool nose R compensation right	○	○
G46	G46	G46	G46	07	Tool nose R compensation (direction automatically selected) ON	○	○
G43.1	G43.1	G43.1	G43.1	20	1st spindle control mode	-	○
G44.1	G44.1	G44.1	G44.1	20	Selected spindle control mode	-	○

### 3. Data Formats

#### 3.8 G code lists

G code list				Group	Function	E60	E68
2	3	6	7				
G47.1	G47.1	G47.1	G47.1	20	All spindles simultaneous control mode	-	○
G50	G92	G50	G92	00	Coordinate system setting/Spindle clamp speed setting	○	○
G52	G52	G52	G52	00	Local coordinate system setting	○	○
G53	G53	G53	G53	00	Machine coordinate system selection	○	○
*G54	*G54	*G54	*G54	12	Workpiece coordinate system selection 1	○	○
G55	G55	G55	G55	12	Workpiece coordinate system selection 2	○	○
G56	G56	G56	G56	12	Workpiece coordinate system selection 3	○	○
G57	G57	G57	G57	12	Workpiece coordinate system selection 4	○	○
G58	G58	G58	G58	12	Workpiece coordinate system selection 5	○	○
G59	G59	G59	G59	12	Workpiece coordinate system selection 6	○	○
G54.1	G54.1	G54.1	G54.1	12	Workpiece coordinate system 48 sets expanded	○	○
G61	G61	G61	G61	13	Exact stop check mode	○	○
G62	G62	G62	G62	13	Automatic corner override	○	○
G63	G63	G63	G63	13	Tapping mode	○	○
*G64	*G64	*G64	*G64	13	Cutting mode	○	○
G65	G65	G65	G65	00	User macro call	○	○
G66	G66	G66	G66	14	User macro modal call A	○	○
G66.1	G66.1	G66.1	G66.1	14	User macro modal call B	○	○
*G67	*G67	*G67	*G67	14	User macro modal call cancel	○	○
G70	G70	G70	G70	09	Finishing cycle	○	○
G71	G71	G71	G71	09	Longitudinal rough cutting cycle	○	○
G72	G72	G72	G72	09	Face rough cutting cycle	○	○
G73	G73	G73	G73	09	Formed material rough cutting cycle	○	○
G74	G74	G74	G74	09	Face cut-off cycle	○	○
G75	G75	G75	G75	09	Longitudinal cut-off cycle	○	○
G76	G76	G76	G76	09	Compound thread cutting cycle	○	○

G code list				Group	Function	E60	E68
2	3	6	7				
G90	G77	G90	G77	09	Longitudinal cutting fixed cycle	○	○
G92	G78	G92	G78	09	Thread cutting fixed cycle	○	○
G94	G79	G94	G79	09	Face cutting fixed cycle	○	○
*G80	*G80	*G80	*G80	09	Hole drilling fixed cycle cancel	○	○
G79	G83.2	G79	G83.2	09	Deep hole drilling cycle 2	○	○
G83	G83	G83	G83	09	Deep hole drilling cycle (Z axis)	○	○
G84	G84	G84	G84	09	Tap cycle (Z axis)	○	○
G85	G85	G85	G85	09	Boring cycle (Z axis)	○	○
G87	G87	G87	G87	09	Deep hole drilling cycle (X axis)	○	○
G88	G88	G88	G88	09	Tap cycle (X axis)	○	○
G89	G89	G89	G89	09	Boring cycle (X axis)	○	○
G84.1	G84.1			09	Reverse tap cycle (Z axis)	○	○
G88.1	G88.1			09	Reverse tap cycle (X axis)	○	○
G50.3	G92.1	G50.3	G92.1	00	Workpiece coordinate preset	-	○
ΔG96	ΔG96	ΔG96	ΔG96	17	Constant surface speed control ON	○	○
ΔG97	ΔG97	ΔG97	ΔG97	17	Constant surface speed control OFF	○	○
ΔG98	ΔG94	ΔG98	ΔG94	05	Asynchronous feed (Feed per minute)	○	○
ΔG99	ΔG95	ΔG99	ΔG95	05	Synchronous feed (Feed per rotation)	○	○
-	ΔG90	-	ΔG90	03	Absolute value command	○	○
-	ΔG91	-	ΔG91	03	Incremental value command	○	○
-	*G98	-	*G98	10	Fixed cycle initial return	○	○
-	G99	-	G99	10	Fixed cycle R point return	○	○
					Spindle synchronization		
G114.1	G114.1			00	Absolute value command	-	○

**(Note 1)** A (\*) symbol indicates the G code to be selected in each group when the power is turned ON or when a reset is executed to initialize the modal.

**(Note 2)** A (Δ) symbol indicates the G code for which parameters selection is possible as an initialization status when the power is turned ON or when a reset is executed to initialize the modal. Note that inch/metric changeover can only be selected when the power is turned ON.


**(Note 3)** If two or more G codes from the same group are commanded, the last G code will be valid.

**(Note 4)** This G code list is a list of conventional G codes. Depending on the machine, movements that differ from the conventional G commands may be included when called by the G code macro. Refer to the Instruction Manual issued by the machine tool builder.

(Note 5) Whether the modal is initialized differs for each reset input.

- (1) "Reset 1"  
The modal is initialized when the reset initialization parameter (#1151 rstinit) is ON.
- (2) "Reset 2 "and "Reset and Rewind"  
The modal is initialized when the signal is input.
- (3) Reset at emergency stop release  
Conforms to "Reset 1".
- (4) When an automatic reset is carried out at the start of individual functions, such as reference point return.  
Conforms to "Reset and Rewind".

#### CAUTION



 The commands with "no value after G" will be handled as "G00".

### 3.9 Precautions before starting machining



#### Precautions before machining

#### CAUTION

-  When creating the machining program, select the appropriate machining conditions, and make sure that the performance, capacity and limits of the machine and NC are not exceeded. The examples do not consider the machining conditions.
-  Before starting actual machining, always carry out dry run operation to confirm the machining program, tool offset amount and workpiece offset amount, etc.

## 4. Buffer Register

### 4.1 Input buffer

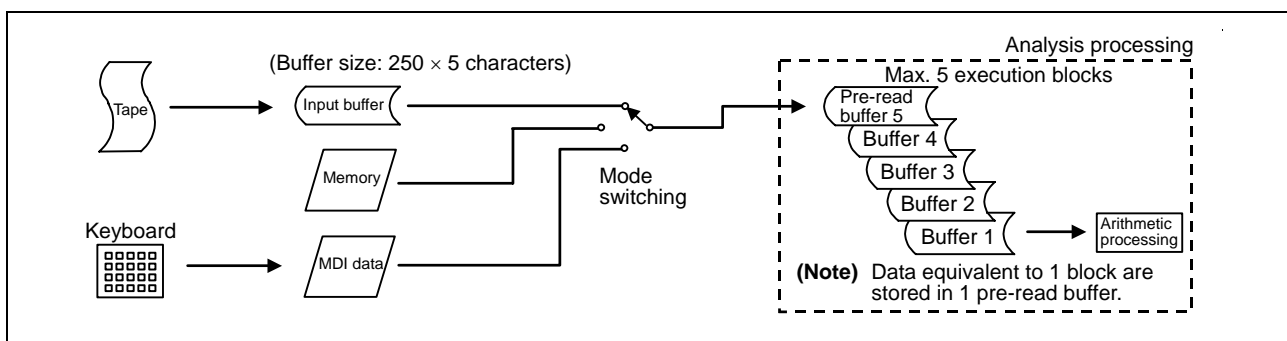


#### Function and purpose

When the pre-read buffer is empty during a tape operation or RS-232C operation, the contents of the input buffer are immediately transferred to the pre-read buffers, and provided that the data stored in the input buffer do not exceed  $250 \times 4$  characters, the following data (Max. 250 characters) are read and loaded into the input buffer.

This buffer is designed to eliminate the operational delay originating in the readout time of the tape reader and to smooth out the block joints.

The pre-reading effects are lost, however, when the block execution time is shorter than the tape readout time of the following block.



The input buffer has a memory capacity of  $250 \times 5$  characters (including the EOB code).

- (1) The contents of the input buffer register are updated in 250-character units.
- (2) Only the significant codes in the significant data section are read into the input buffer.
- (3) When codes (including "(" and ")") are sandwiched in the control in or control out mode and the optional block skip function is ON, the data extending from the "/" (slash) code up to the EOB code are read into the input buffer.
- (4) The input buffer contents are cleared with resetting.

**(Note 1)** The input buffer size (250 characters) differs according to the model.

#### 4.2 Pre-read buffers



##### Function and purpose

During automatic processing, the contents of 1 block are normally pre-read so that program analysis processing is conducted smoothly. However, during nose R compensation, a maximum of 5 blocks are pre-read for the intersection point calculation including interference check.

The specifications of the data in 1 block are as follows:

- (1) The data of 1 block are stored in this buffer.
- (2) Only the significant codes in the significant data section are read into the pre-read buffer.
- (3) When codes are sandwiched in the control in and control out, and the optional block skip function is ON, the data extending from the "/" (slash) code up to the EOB code are not read into the pre-read buffer.
- (4) The pre-read buffer contents are cleared with resetting.
- (5) When the single block function is ON during continuous operation, the pre-read buffer stores the following block data and then stops operation.



##### Other precautions

- (1) Depending on whether the program is executed continuously or by single blocks, the timing of the valid/invalid for the external control signals for the optional block skip and others will differ.
- (2) If the external control signal such as optional block skip is turned ON/OFF with the M command, the external control operation will not be effective on the program pre-read with the buffer register.
- (3) According to the M command that operates the external controls, it prohibits pre-reading, and the recalculation is as follows:  
The M command that commands the external controls is distinguished at the PLC, and the "recalculation request" for PLC → NC interface table is turned ON.  
(When the "recalculation request" is ON, the program that has been pre-read is reprocessed.)

## 5. Position Commands

### 5.1 Incremental/absolute value commands

## 5. Position Commands

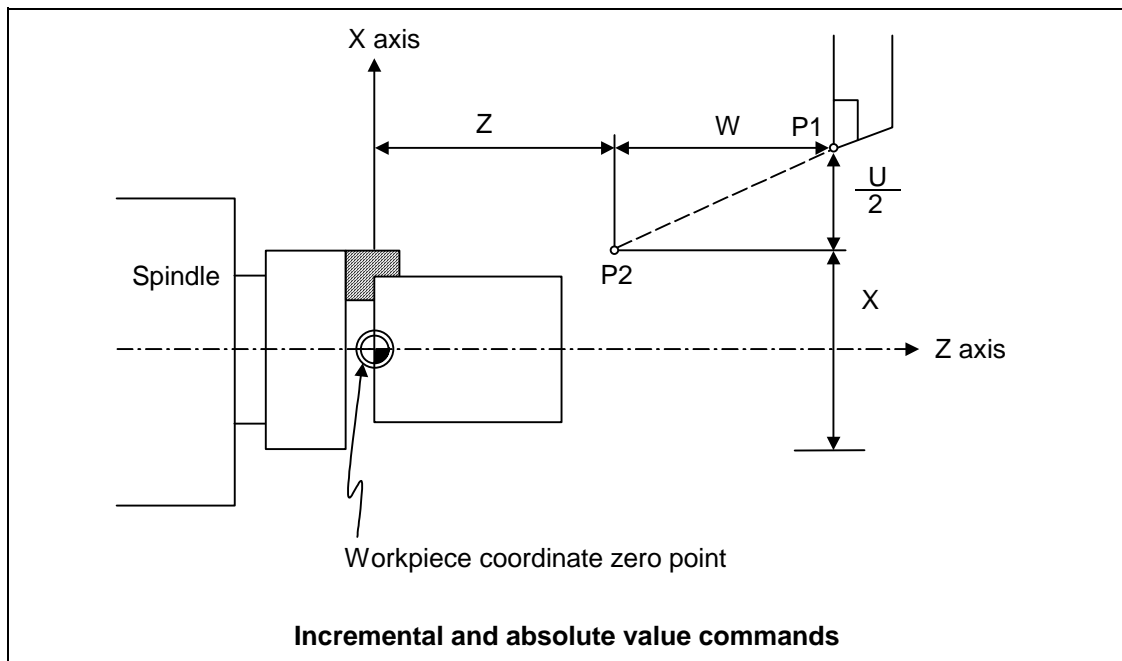
### 5.1 Incremental/absolute value commands



#### Function and purpose

There are 2 methods of issuing tool movement amount commands: the incremental value method and the absolute value method.

The incremental value method applies for coordinates of a point which is to be moved and it issues a command using the distance from the present point; on the other hand, the absolute value system issues a command using the distance from the coordinate zero point. The following figure shows what happens when the tool is moved from point P1 to point P2.



Incremental value commands and absolute value commands for the X axis and Z axis are identified by address when parameter "#1076 AbsInc" is set to 1, and identified by G code (G90/G91) when set to 0. Similarly, even with additional axes (C axis or Y axis), they are differentiated by addresses, or G code.

		Command system	Remarks
Absolute value	X axis	Address X	<ul style="list-style-type: none"> <li>For setting correspondence between addresses and axes into "#1013 axname" and "#1014 incax".</li> </ul>
	Z axis	Address Z	
	C/Y axis	Address C/Y	
Incremental value	X axis	Address U	<ul style="list-style-type: none"> <li>Absolute and incremental values can be used together in the same block.</li> </ul>
	Z axis	Address W	
	C/Y axis	Address H/V	

#### (Example)

X\_\_\_\_\_ W\_\_\_\_\_ ;



Incremental value command for Z axis  
Absolute value command for X axis

**(Note 1)** When parameter "#1076 AbsInc" is 1, and H is used for the incremental command address, address H of blocks in M98, G114.2, and G10 L50 modal will be handled as the parameter of each command, and the axis will not be moved.

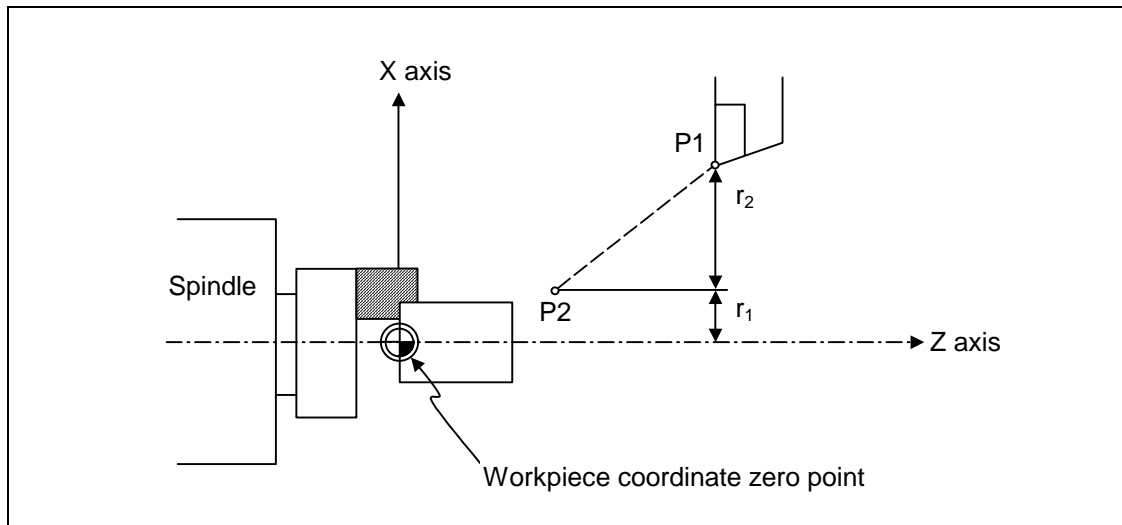
### 5.2 Radius/diameter commands



#### Function and purpose

The cross sections of workpieces machined on a lathe are circular, and the diameter or radius value of those circles can be used for movement commands in the X-axis direction. A radius command will move the tool by the commanded amount only, but a diameter command will move the tool both in the X-axis direction by an amount equivalent to one-half the commanded amount only and in the Z-axis direction by the commanded amount only.

This system permits radius or diameter commands to be issued, depending on the parameter (#1019 dia) setting. The figure below shows the command procedure when the tool is to be moved from point P1 to point P2.



X command		U command		Remarks
Radius	Diameter	Radius	Diameter	
$X = r_1$	$X = 2r_1$	$U = r_2$	$U = 2r_2$	Even when a diameter command has been selected, only the U command can be made a radius command by parameter "#1077 radius".

Radius and diameter commands



#### Precautions and Restrictions

- (1) In the above example, the tool moves from P1 to P2 in the minus direction of the X axis and so when an incremental value is issued, the minus sign is given to the numerical value being commanded.
- (2) In this manual, diameter commands are used in descriptions of both the X and U axes for the sake of convenience.



#### 5.3 Inch/metric conversion; G20, G21



#### Function and purpose

The commands can be changed between inch and metric with the G20/G21 command.



#### Command format

**G20/G21;**  
 G20 Inch command  
 G21 Metric command



#### Detailed description

The G20 and G21 commands merely select the command units. They do not select the Input units. G20 and G21 selection is meaningful only for linear axes and it is meaningless for rotation axes.

**(Example)** Relationship between input command units and G20/G21 commands (with decimal point input type I)

(1) When using E60

Axis	Input command cunit	Command example	mm output (#1016 iout=0)		inch output (#1016 iout=1)	
			G21	G20	G21	G20
X	10	X100 ;	0.100mm	0.254mm	0.0039inch	0.0100inch
Z	10	Z100 ;	0.100mm	0.254mm	0.0039inch	0.0100inch

(2) When using E68

Axis	Input command cunit	Command example	mm output (#1016 iout=0)		inch output (#1016 iout=1)	
			G21	G20	G21	G20
X	10	X100 ;	0.100 mm	0.254 mm	0.0039 inch	0.0100 inch
Z	10	Z100 ;	0.100 mm	0.254 mm	0.0039 inch	0.0100 inch
X	1	X100 ;	0.0100mm	0.0254 mm	0.00039 inch	0.00100 inch
Z	1	Z100 ;	0.0100mm	0.0254 mm	0.00039 inch	0.00100 inch

## 5. Position Commands

### 5.3 Inch/metric conversion



#### Output unit, command unit and setting unit

Select whether to issue the counter or parameter setting and display unit as a command unit determined by the G20/G21 command modal, or as the internal unit determined by parameter "#1041 I\_inch". If the command unit setting display is selected when initial metric (internal unit metric) is set with the base specification parameter "#1226 aux10/bit6", the counter and parameters will be displayed as metric units during the G21 command mode. The internal unit metric data will be converted into an inch unit and displayed during the G20 command mode. The command unit for when the power is turned ON and reset can be selected regardless of the internal unit by setting the base specification parameter "#1152 I\_G20".

#### NC axis

Item	Initial inch OFF (metric internal unit) #1041 I_inch=0				Initial inch ON (inch internal unit) #1041 I_inch=1			
	#1226 aux10/bit6=0 Follows internal unit		#1226 aux10/bit6=1 Follows command unit		#1226 aux10/bit6=0 Follows internal unit		#1226 aux10/bit6=1 Follows internal unit	
	G21	G20	G21	G20	G21	G20	G21	G20
Movement/ speed command	Metric	Inch	Metric	Inch	Metric	Inch	The "#1226 aux10/bit6" parameter is invalid when initial inch is ON. Even if 1 is set, the setting and display will follow the internal unit in the same manner as when 0 is set.	
Counter display	Metric	Metric	Metric	Inch	Inch	Inch		
Speed display	Metric	Metric	Metric	Inch	Inch	Inch		
User parameter setting/display	Metric	Metric	Metric	Inch	Inch	Inch		
Workpiece/ tool offset setting/display	Metric	Metric	Metric	Inch	Inch	Inch		
Handle feed command	Metric	Metric	Metric	Inch	Inch	Inch		
Setup parameter setting/display	Follows "#1040 M_inch"							

#### PLC axis

Item	#1042 pcinch=0 (metric)	#1042 pcinch=1 (inch)
Movement/ speed command	Metric	Inch
Counter display	Metric	Inch
User parameter setting/display	Metric	Inch
Setup parameter setting/display	Follows "#1040 M_inch"	

When using both the NC axis and PLC axis as rotation axes, three digits below the decimal point will be displayed for the coordinate data, etc., even for the inch command.

# 5. Position Commands

## 5.3 Inch/metric conversion



### Usage examples

The screen display will change in the following manner according to the parameter settings.

Parameter setting Program command/ screen operation	#1041 I_inch 0 #1152 I_G20 0 #1226 aux10/bit6 0 Initial metric G21 (metric) mode at reset Setting and display unit is internal unit	#1041 I_inch 0 #1152 I_G20 0 #1226 aux10/bit6 1 Initial metric G21 (metric) mode at reset Setting and display unit is command unit
	<div data-bbox="512 577 756 741" style="border: 1px solid black; padding: 5px;">                     [POSITION]                      X 0.000                      Y 0.000                      Z 0.000                      mm.                 </div> <div data-bbox="512 763 756 927" style="border: 1px solid black; padding: 5px;">                     [TOOL OFFSET]                      #                      1 0.100 11 0.000                      2 0.200 12 0.000                      3 0.300 13 0.000                      mm.                 </div>	<div data-bbox="979 577 1224 741" style="border: 1px solid black; padding: 5px;">                     [POSITION]                      X 0.000                      Y 0.000                      Z 0.000                      mm.                 </div> <div data-bbox="979 763 1224 927" style="border: 1px solid black; padding: 5px;">                     [TOOL OFFSET]                      #                      1 0.100 11 0.000                      2 0.200 12 0.000                      3 0.300 13 0.000                      mm.                 </div>
Inch command G20 G0 X1. Y2. Z3. ;	<div data-bbox="512 1016 756 1180" style="border: 1px solid black; padding: 5px;">                     [POSITION]                      X 25.400                      Y 50.800                      Z 76.200                      in.                 </div> <div data-bbox="767 1003 906 1137" style="margin-left: 20px;">                     Internal data                      25.4(mm)                      50.8(mm)                      76.2(mm)                 </div> <div data-bbox="512 1234 756 1397" style="border: 1px solid black; padding: 5px;">                     [TOOL OFFSET]                      #                      1 0.100 11 0.000                      2 0.200 12 0.000                      3 0.300 13 0.000                      in.                 </div>	<div data-bbox="959 943 1358 1003" style="margin-left: 20px;">                     Display unit is also inch following the                      command unit                 </div> <div data-bbox="979 1016 1224 1180" style="border: 1px solid black; padding: 5px;">                     [POSITION]                      X 1.0000                      Y 2.0000                      Z 3.0000                      in.                 </div> <div data-bbox="1235 1003 1374 1137" style="margin-left: 20px;">                     Internal data                      25.4(mm)                      50.8(mm)                      76.2(mm)                 </div> <div data-bbox="979 1234 1224 1397" style="border: 1px solid black; padding: 5px;">                     [TOOL OFFSET]                      #                      1 0.0039 11 0.0000                      2 0.0078 12 0.0000                      3 0.0118 13 0.0000                      in.                 </div> <div data-bbox="1235 1220 1374 1281" style="margin-left: 20px;">                     Internal data                      #10.1(mm)                 </div>
#(1)( 1) Input	<div data-bbox="512 1487 756 1650" style="border: 1px solid black; padding: 5px;">                     [TOOL OFFSET]                      #                      1 1.000 11 0.000                      :                      in.                 </div> <div data-bbox="767 1473 906 1541" style="margin-left: 20px;">                     Internal data                      1.0(mm)                 </div>	<div data-bbox="959 1435 1150 1464" style="margin-left: 20px;">                     Set with inch unit                 </div> <div data-bbox="979 1487 1224 1650" style="border: 1px solid black; padding: 5px;">                     [TOOL OFFSET]                      #                      1 1.0000 11 0.0000                      :                      in.                 </div> <div data-bbox="1235 1473 1374 1541" style="margin-left: 20px;">                     Internal data                      25.4(mm)                 </div>

## 5. Position Commands

### 5.3 Inch/metric conversion

Parameter setting	#1041 I_inch 0 #1152 I_G20 0	Initial metric G21 (metric) mode at reset	#1041 I_inch 0 #1152 I_G20 0	Initial metric G21 (metric) mode at reset
Program command/ screen operation	#1226 aux10/bit6 0	Setting and display unit is internal unit	#1226 aux10/bit6 1	Setting and display unit is command unit
Reset	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <pre>[POSITION] X 25.400 Y 50.800 Z 76.200  mm.</pre> </div> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> <pre>[TOOL OFFSET] # 1 1.000 11 0.000 2 0.200 12 0.000 3 0.300 13 0.000  mm.</pre> </div>		<p>Display unit is also metric following the command unit</p> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> <pre>[POSITION] X 25.400 Y 50.800 Z 76.200  mm.</pre> </div> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> <pre>[TOOL OFFSET] # 1 25.400 11 0.000 2 0.200 12 0.000 3 0.300 13 0.000  mm.</pre> </div>	



#### Precautions

- (1) If the tool and workpiece offset data are added with an inch command unit and metric internal unit, an error may occur.
- (2) The internal unit is a unit that follows #1041 I\_inch.  
Internal unit when #1041 I\_inch=0 : mm  
Internal unit when #1041 I\_inch=1 : inch
- (3) The parameters are input and output with an internal unit.
- (4) If the tool data input/output unit selection (#1220 aux04/bit4:1) is valid, the tool data will be input and output with the "#1152 I\_G20" setting unit. However, in all other cases, the data will be input/output with the internal unit.

### 5.4 Decimal point input



#### Function and purpose

This function enables the decimal point to be input. It assigns the decimal point in millimeter or inch units for the machining program input information that defines the tool paths, distances and speeds.

A parameter "#1078 Decpt2" selects whether minimum input command unit (type I) or zero point (type II) is to apply for the least significant digit of data without a decimal point.



#### Command format

○○○○○.○○○	Metric system
○○○○.○○○○	Inch system



#### Detailed description

- (1) The decimal point command is valid for the distances, angles, times and speeds in machining programs.
- (2) Refer to the Page 29, Table rising the "Addresses used and valid/invalid decimal point commands" for details on the valid addresses for the decimal point commands.
- (3) The number of significant digits in a decimal point command is shown below (for input command unit cunit=10).

	Movement command (linear)		Movement command (rotation)		Feedrate		Dwell (X)	
	Integer	Decimal part	Integer	Decimal part	Integer	Decimal part	Integer	Decimal part
MM (millimeter)	0 ~ 99999.	.000 ~ .999	0 ~ 99999.	.000 ~ .999	0 ~ 60000.	.000 ~ .999	0 ~ 99999.	.000 ~ .999
					0 ~ 999.	.0000 ~ .9999		
INCH (inch)	0 ~ 9999.	.0000 ~ .9999	99999. (359.)	.0 ~ .999	0 ~ 2362.	.0000 ~ .9999	0 ~ 99.	.000 ~ .999
					0 ~ 99.	.000000 ~ .999999		

**(Note)** The top row gives the feedrate as a per-minute rate and the bottom row as a per-rotation rate.

- (4) The decimal point command is valid even for commands defining the variable data used in subprograms.
- (5) Decimal point commands for decimal point invalid addresses are processed as integer data only and everything below the decimal point is ignored. Addresses which are invalid for the decimal point are D, H, L, M, N, O, P, S and T. All variable commands, however, are treated as data with decimal points.



#### Precautions

- (1) If an arithmetic operator is inserted, the data will be handled as data with a decimal point.

**(Example1)** G00 X123+0 ;

This is the X axis command 123mm command. It will not be 123μm.



#### Example of program

##### (1) Example of program for decimal point valid address

Specification division Program example	Decimal point command 1		Decimal point command 2 When 1 = 1μm
	When 1 = 1μm	When 1 = 10μm	
G0 X123.45 (decimal points are all mm points)	X123.450mm	X123.450mm	X123.450mm
G0 X12345	X12.345mm (last digit is 1μm unit)	X123.450mm	X12345.000mm
#111 = 123, #112 = 5.55 X#111 Z#112	X123.000mm, Z5.550mm	X123.000mm, Z5.550mm	X123.000mm, Z5.550mm
#113 = #111 + #112 (addition)	#113 = 128.550	#113 = 128.550	#113 = 128.550
#114 = #111 - #112 (subtraction)	#114 = 117.450	#114 = 117.450	#114 = 117.450
#115 = #111 * #112 (multiplication)	#115 = 682.650	#115 = 682.650	#115 = 682.650
#116 = #111/#112 #117 = #112/#111 (division)	#116 = 22.162, #117 = 0.045	#116 = 22.162, #117 = 0.045	#116 = 22.162, #117 = 0.045



#### Decimal point input I, II and decimal point command validity

In the table on the next page, decimal point input I and II result in the following for commands in which a decimal point is not used in an address where a decimal point command is valid. Both decimal point input I and II become the same for commands using a decimal point.

##### (1) Decimal point input I

The lowest order digit of command data matches the command unit.

**(Example)** When "X1" is commanded in 1μm system, the same result occurs as for an "X0.001" command.

##### (2) Decimal point input II

The lowest order digit of command data matches the command unit.

**(Example)** When "X1" is commanded in 1μm system, the same result occurs as for an "X1." command.

## 5. Position Commands

### 5.4 Decimal point input

#### Addresses used and valid/invalid decimal point commands

Address	Decimal point command	Application	Remarks
A	Valid	Coordinate position data	
	Invalid	2nd miscellaneous function code	
	Valid	Angle data	
	Invalid	MRC program number	
	Invalid	Program parameter input, axis No.	
	Valid	Deep hole drilling cycle (2) Safety distance	
	Valid	Spindle synchronous acceleration/deceleration time constant	
B	Valid	Coordinate position data	
	Invalid	2nd miscellaneous function code	
C	Valid	Coordinate position data	
	Invalid	2nd miscellaneous function code	
	Valid	Corner chamfering amount	,C
	Valid	Program tool offset input Nose R compensation amount (incremental)	
D	Valid	Automatic tool length measurement, deceleration range d	
	Invalid	Program parameter input, byte type data	
E	Invalid	Synchronous spindle No. at spindle synchronization	
	Valid	Inch threads Precision thread lead	
F	Valid	Corner cutting feedrate	
	Valid	Feedrate	
G	Valid	Thread lead	
	Valid	Preparatory function code	
H	Valid	Coordinate position data	
	Invalid	Sequence numbers in subprograms	
	Invalid	Program parameter input, bit type data	
	Invalid	Selection of linear - arc intersection (geometric)	
I	Invalid	Basic spindle No. at spindle synchronization	
	Valid	Circular center coordinates	
	Valid	Nose R compensation/ tool radius compensation vector components	
	Valid	Deep hole drilling (2) First cut amount	
J	Valid	G0/G1 in-position width Hole drilling cycle G0 in-position width	,I
	Valid	Deep hole drilling (2) Dwell at return point	
	Invalid	Hole drilling cycle G1 in-position width	,J
	Invalid	Return sequence number from subprogram	
K	Valid	Coordinate position data	
	Invalid	Skip signal command	
	Valid	Arc center coordinates (absolute value) (geometric)	
	Invalid	Extended work coordinate system No.	
L	Valid	Circular center coordinates	
	Valid	Nose R compensation/tool radius compensation vector components	
	Invalid	Hole machining cycle Number of repetitions	
	Valid	Deep hole drilling cycle (2) Second and subsequent cut amounts	
M	Valid	Thread lead increase/decrease amount (variable lead thread cutting)	
	Invalid	Subprogram Number of repetitions	
	Invalid	Program tool compensation input type selection	L2 L10 L11
	Invalid	Programmable Current Limitation selection L14	
N	Invalid	Program parameter input, selection	L50
	Invalid	Program parameter input, two-word type data	4 bytes
O	Invalid	Miscellaneous function codes	
	Invalid	Sequence numbers	
P	Invalid	Program parameter input, data No.	
	Invalid	Program numbers	
	Invalid	Dwell time	
	Invalid	Subprogram call program numbers	
Q	Invalid	2nd, 3rd and 4th reference point number	
	Invalid	Constant surface speed control, axis number	
	Invalid	MRC finishing shape start sequence number	
	Invalid	Program numbers	

Address	Decimal point command	Application	Remarks
A	Valid	Coordinate position data	
	Invalid	2nd miscellaneous function code	
	Valid	Angle data	
	Invalid	MRC program number	
	Invalid	Program parameter input, axis No.	
	Valid	Deep hole drilling cycle (2) Safety distance	
	Valid	Spindle synchronous acceleration/deceleration time constant	
B	Valid	Coordinate position data	
	Invalid	2nd miscellaneous function code	
C	Valid	Coordinate position data	
	Invalid	2nd miscellaneous function code	
	Valid	Corner chamfering amount	,C
	Valid	Program tool offset input Nose R compensation amount (incremental)	
D	Valid	Automatic tool length measurement, deceleration range d	
	Invalid	Program parameter input, byte type data	
E	Invalid	Synchronous spindle No. at spindle synchronization	
	Valid	Inch threads Precision thread lead	
F	Valid	Corner cutting feedrate	
	Valid	Feedrate	
G	Valid	Thread lead	
	Valid	Preparatory function code	
H	Valid	Coordinate position data	
	Invalid	Sequence numbers in subprograms	
	Invalid	Program parameter input, bit type data	
	Invalid	Selection of linear - arc intersection (geometric)	
I	Invalid	Basic spindle No. at spindle synchronization	
	Valid	Circular center coordinates	
	Valid	Nose R compensation/ tool radius compensation vector components	
	Valid	Deep hole drilling (2) First cut amount	
J	Valid	G0/G1 in-position width Hole drilling cycle G0 in-position width	,I
	Valid	Deep hole drilling (2) Dwell at return point	
	Invalid	Hole drilling cycle G1 in-position width	,J
	Invalid	Return sequence number from subprogram	
K	Valid	Coordinate position data	
	Invalid	Skip signal command	
	Valid	Arc center coordinates (absolute value) (geometric)	
	Invalid	Extended work coordinate system No.	
L	Valid	Circular center coordinates	
	Valid	Nose R compensation/tool radius compensation vector components	
	Invalid	Hole machining cycle Number of repetitions	
	Valid	Deep hole drilling cycle (2) Second and subsequent cut amounts	
M	Valid	Thread lead increase/decrease amount (variable lead thread cutting)	
	Invalid	Subprogram Number of repetitions	
	Invalid	Program tool compensation input type selection	L2 L10 L11
	Invalid	Programmable Current Limitation selection L14	
N	Invalid	Program parameter input, selection	L50
	Invalid	Program parameter input, two-word type data	4 bytes
O	Invalid	Miscellaneous function codes	
	Invalid	Sequence numbers	
P	Invalid	Program parameter input, data No.	
	Invalid	Program numbers	
	Invalid	Dwell time	
	Invalid	Subprogram call program numbers	
Q	Invalid	2nd, 3rd and 4th reference point number	
	Invalid	Constant surface speed control, axis number	
	Invalid	MRC finishing shape start sequence number	
	Invalid	Program numbers	

**(Note 1)** Decimal points are all valid in user macro arguments.

## 5. Position Commands

### 5.4 Decimal point input

Address	Decimal point command	Application	Remarks
P	Valid	Cut-off cycle shift amount/cut amount	
	Invalid	Compound thread cutting cycle, number of cutting passes, chamfering, tool nose angle	
	Valid	Compound thread cutting cycle Thread height	
	Invalid	Program tool compensation input compensation No.	
	Invalid	Program parameter input, section No.	
	Invalid	Return sequence number from subprogram	
	Valid	Coordinate position data	
	Invalid	Skip signal command	
	Valid	Arc center coordinates (absolute value) (geometric)	
	Invalid	Extended work coordinate system No.	
Q	Invalid	Minimum spindle clamp rotation speed	
	Invalid	MRC finishing shape end sequence number	
	Valid	Cut-off cycle Cut amount/shift amount	
	Valid	Compound thread cutting cycle Minimum cut amount	
	Valid	Compound thread cutting cycle First cut amount	
	Valid	Deep hole drilling cycle 1 Cut amount of each pass	
	Invalid	Program tool compensation input Hypothetical tool nose point number	
	Invalid	Deep hole drilling cycle (2) Dwell at cut point	
	Valid	Arc center coordinates (absolute value) (geometric)	
	Valid	Thread cutting start shift angle	
R	Valid	R-designated arc radius	
	Valid	Corner rounding circular radius	,R
	Valid	Automatic tool length measurement, deceleration range r	
	Valid	MRC longitudinal/face escape amount	
	Invalid	MRC shaping division number	
	Valid	Cut-off cycle, return amount	

Address	Decimal point command	Application	Remarks
R	Valid	Cut-off cycle, escape amount	
	Valid	Compound thread cutting cycle, finishing allowance	
	Valid	Synchronous spindle phase shift amount	
	Valid	Compound thread cutting cycle/turning cycle, taper difference	
	Valid	Hole drilling cycle/deep hole drilling cycle (2), distance to reference point	
	Valid	Program tool compensation input/nose R compensation amount (absolute)	
	Valid	Coordinate position data	
	Valid	Rough cutting cycle (longitudinal) (face) pull amount	
	Valid	Synchronous tap/asynchronous tap changeover	,R
	Invalid	Extended work coordinate system No.	
S	Invalid	Spindle function codes	
	Invalid	Maximum spindle clamp rotation speed	
	Invalid	Constant surface speed control, surface speed	
	Invalid	Program parameter input, word type data	2 bytes
T	Invalid	Tool function codes	
U	Valid	Coordinate position data	
	Valid	Program tool compensation input	
	Valid	Rough cutting cycle (longitudinal) cutting amount	
	Valid	Dwell time	
V	Valid	Coordinate position data	
	Valid	Program tool compensation input	
W	Valid	Coordinate position data	
	Valid	Program tool compensation input	
	Valid	Rough cutting cycle (face) cutting amount	
X	Valid	Coordinate position data	
	Valid	Dwell	
	Valid	Program tool compensation input	
Y	Valid	Coordinate position data	
	Valid	Program tool compensation input	
Z	Valid	Coordinate position data	
	Valid	Program tool compensation input	



## 6. Interpolation Functions

### 6.1 Positioning (Rapid Traverse); G00



#### Function and purpose

This command is accompanied by coordinate words. It positions the tool along a linear or non-linear path from the present point as the start point to the end point which is specified by the coordinate words.



#### Command format

**G00 Xx/Uu Zz/Ww ;**

x, u, z, w                      Coordinate values


The command addresses are valid for all additional axes.



#### Detailed description

- (1) Once this command has been issued, the G00 mode is retained until it is changed by another G function or until the G01, G02, G03, G33 or G34 command in the 01 group is issued. If the next command is G00, all that is required is simply that the coordinate words be specified.
- (2) In the G00 mode, the tool is always accelerated at the start point of the block and decelerated at the end point. Execution proceeds to the next block after it has been confirmed that the command of the present block is 0, and the tracking error status of the acceleration/ deceleration has been confirmed. The in-position width is set with commanded block's address (,I) or the parameters.
- (3) Any G command (G83 ~ G89) in the 09 group is cancelled (G80) by the G00 command.
- (4) Whether the tool moves along a linear or non-linear path is determined by parameter, but the positioning time does not change.
  - (a) · Linear path..... This is the same as linear interpolation (G01), and the speed is limited by the rapid traverse rate of each axis.
  - (b) · Non-linear path ..... The tool is positioned at the rapid traverse rate independently for each axis.
- (5) When no number following the G address, this is treated as G00.

#### CAUTION

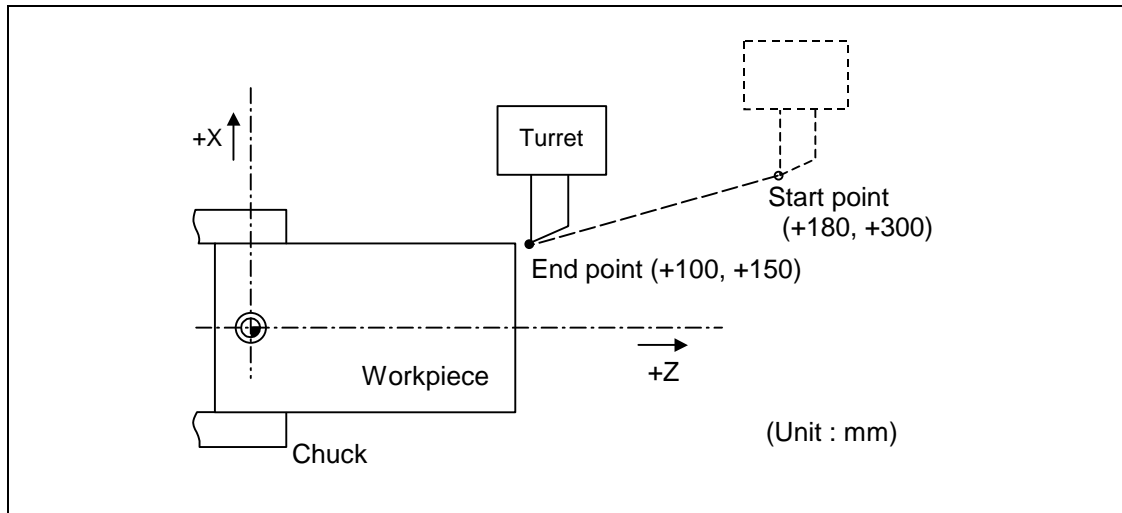
 The commands with "no value after G" will be handled as "G00".

## 6. Interpolation Functions

### 6.1 Positioning (Rapid Traverse)



#### Example of program



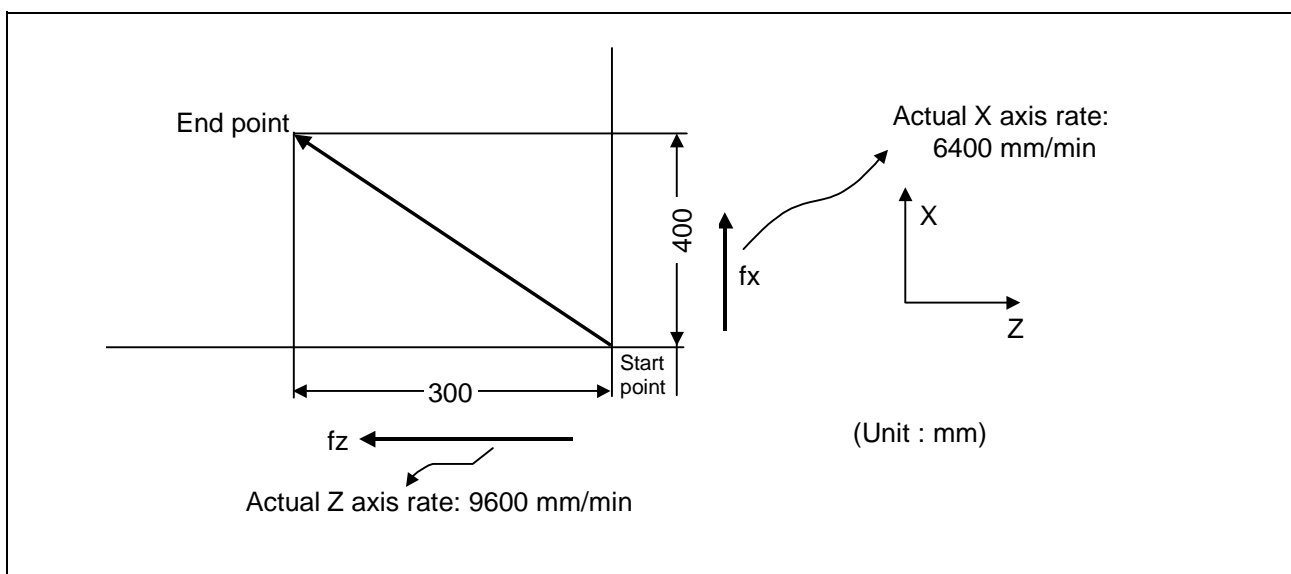
G00 X100000 Z150000 ;	Absolute value command
G00 U-80000 W-150000 ;	Incremental value command (With an input setting unit of 0.001mm)



#### Precautions

**(Note 1)** When the parameter "#1086 G0Intp" is 0, the path along which the tool is positioned is the shortest path connecting the start and end points. The positioning speed is automatically calculated so that the shortest distribution time is obtained in order that the commanded speeds for each axis do not exceed the rapid traverse rate. When, for instance, the X-axis and Z-axis rapid traverse rates are both 9600mm/min, the tool will follow the path in the figure below if the following is programmed:

G00 Z-300000 X400000 ; (With an input setting unit of 0.001mm)



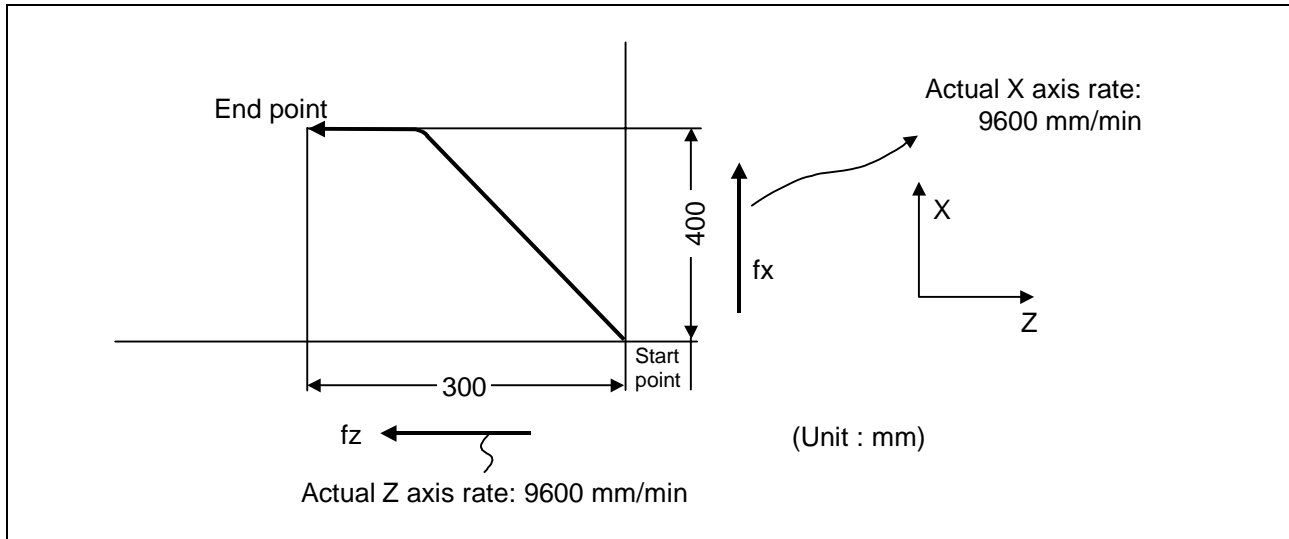
## 6. Interpolation Functions

### 6.1 Positioning (Rapid Traverse)

**(Note 2)** When parameter "#1086 G0Intp" is 1, the tool will move along the path from the start point to the end point at the rapid traverse rate of each axis.

When, for instance, the X-axis and Z-axis rapid traverse rates are both 9600 mm/min, the tool will follow the path in the figure below if the following is programmed:

G00 Z - 300000 X400000 ; (With an input setting unit of 0.001mm)



**(Note 3)** The rapid traverse rate for each axis with the G00 command differs according to the individual machine and so reference should be made to the machine specifications manual.

**(Note 4)** Rapid traverse (G00) deceleration check

There are two methods for the deceleration check at rapid traverse; commanded deceleration method and in-position check method. Select a method with the parameter "#1193 inpos".

■ When "inpos" = "1"

Upon completion of the rapid traverse (G00), execute the next block after confirming that the remaining distances for each axis are below the fixed amounts. (Refer to Fig. 1.)

The confirmation of the remaining distance should be done with the imposition width,  $L_R$ .  $L_R$  is the setting value for the servo parameter "#2224 SV024" or "#2077 G0inps".

The setting for the parameter "#2224 SV024" can be made in increments of 0.0005mm or 0.00005 inch.

The purpose of checking the rapid traverse deceleration is to minimize the time it takes for positioning. The bigger the setting value for the in-position, the longer the reduced time is, but the remaining distance of the previous block at the starting time of the next block also becomes larger, and this could become an obstacle in the actual processing work. The check for the remaining distance is done at set intervals. Accordingly, it may not be possible to get the actual amount of time reduction for positioning with the setting value in-position.

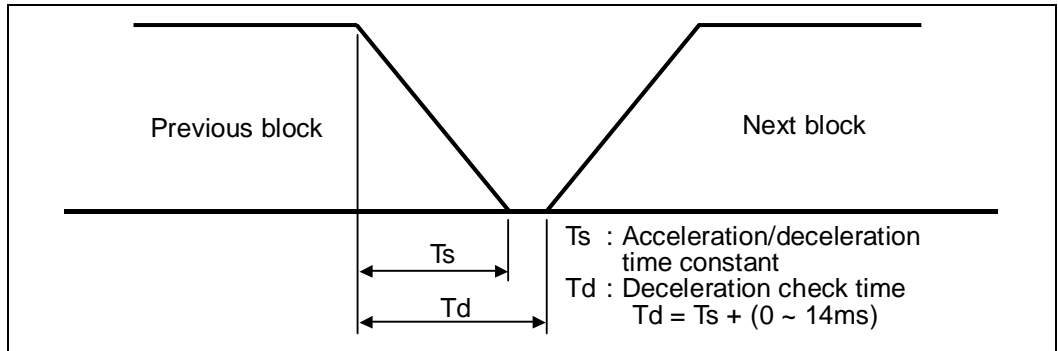
## 6. Interpolation Functions

### 6.1 Positioning (Rapid Traverse)

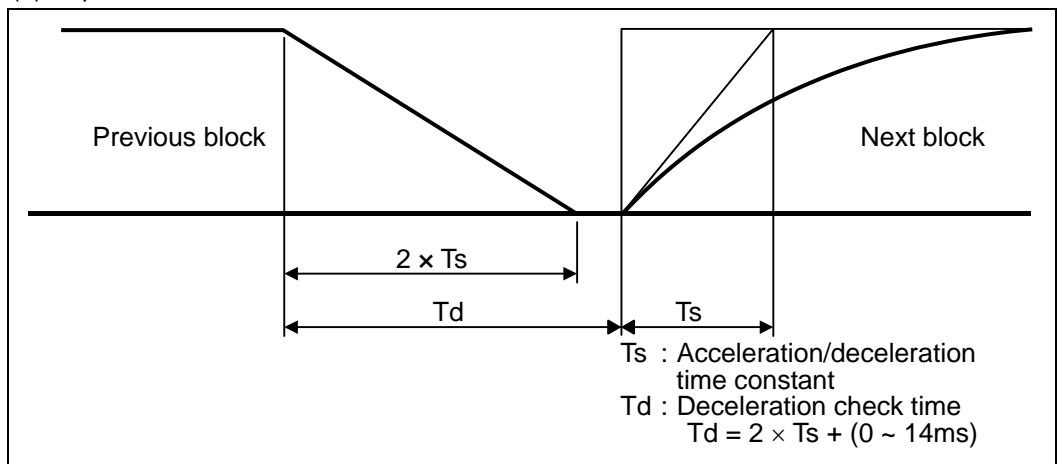
■ When “inpos” = “0”

Upon completion of the rapid traverse (G00), the next block will be executed after the deceleration check time (Td) has elapsed. The deceleration check time (Td) is as follows, depending on the acceleration/deceleration type.

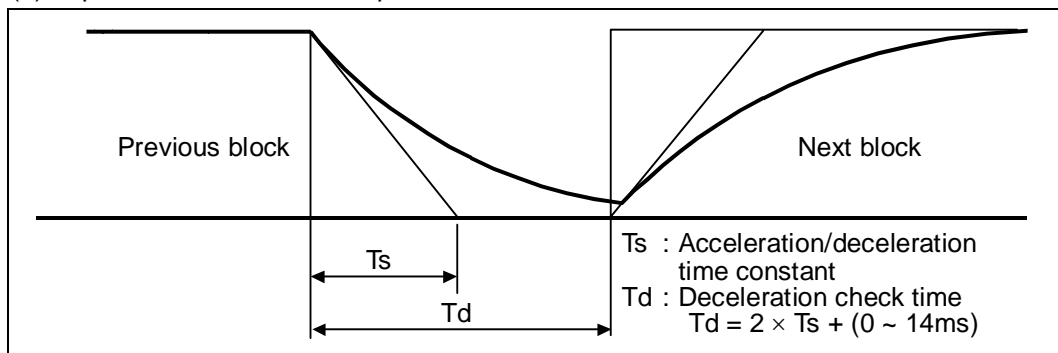
(1) Linear acceleration/linear deceleration .....  $T_d = T_s + \alpha$



(2) Exponential acceleration/linear deceleration.....  $T_d = 2 \times T_s + \alpha$



(3) Exponential acceleration/exponential deceleration.....  $T_d = 2 \times T_s + \alpha$



Where  $T_s$  is the acceleration time constant,  $\alpha = 0$  to  $14\text{ms}$

The time required for the deceleration check during rapid traverse is the longest among the rapid traverse deceleration check times of each axis determined by the rapid traverse acceleration/deceleration time constants and by the rapid traverse acceleration/deceleration mode of the axes commanded simultaneously.

## 6. Interpolation Functions

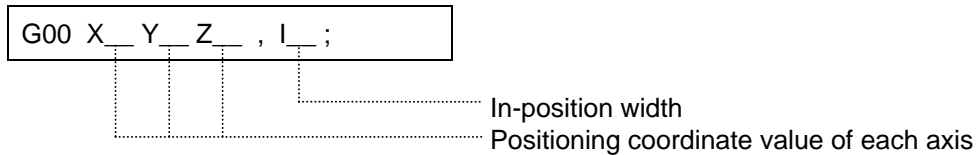
### 6.1 Positioning (Rapid Traverse)



#### Programmable in-position width command for positioning

[E68]

This command commands the in-position width for the positioning command from the machining program.



`,I_` : In-position width. A program error occurs if a decimal point command is issued. This is valid only in the commanded block. A block that does not contain this address will follow the parameter "#1193 inpos" settings.



#### Operation during in-position check

[E68]

Execution of the next block starts after confirming that the position error amount of the positioning (rapid traverse: G00) command block and the block that carries out deceleration check with the linear interpolation (G01) command is less than the in-position width issued in this command.

The in-position width in this command is valid only in the command block, so the deceleration check method set in base specification parameter "#1193 inpos" is used for blocks that do not have the in-position width command.

When there are several movement axes, the system confirms that the position error amount of each movement axis is less than the in-position width issued in this command before executing the next block.

The differences of when the in-position check is validated with the parameter (base specification parameter "#1193 inpos" set to 1; refer to next page for in-position width) and when validated with this command are shown in the following drawing.

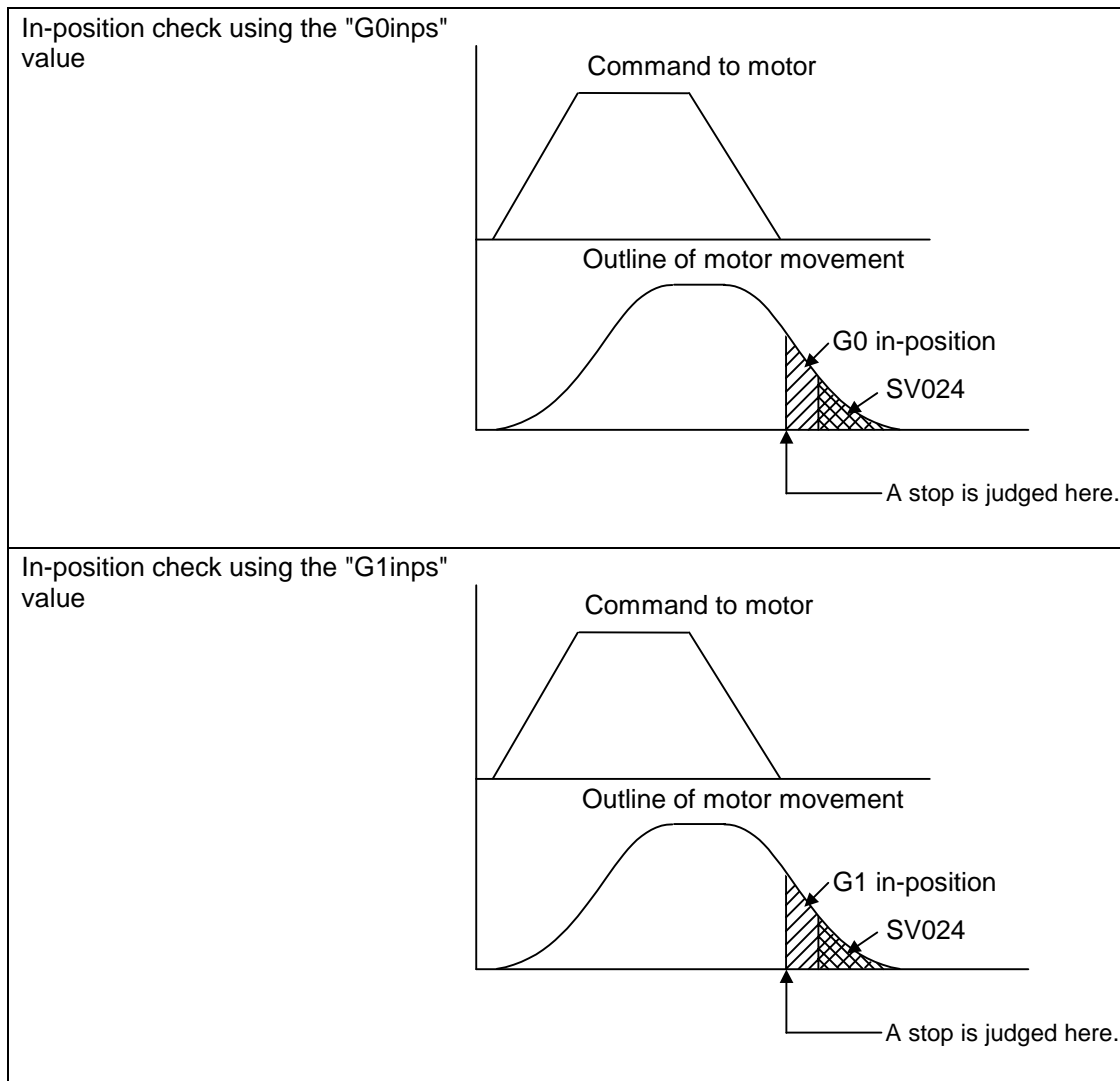
Differences between in-position check with this command and in-position check with parameter

In-position check with ",I" address command	In-position check with parameter
After starting deceleration of the command system, the position error amount and commanded in-position width are compared.	After starting deceleration of the command system, the servo system's position error amount and the parameter setting value (in-position width) are compared.
<p>Ts : Acceleration/deceleration time constant</p> <p>Td : Deceleration check time</p> <p>Td = Ts + (0 to 14ms)</p>	



#### G0/G1 in-position width setting

When the setting value of the servo parameter "#2224 SV024" is smaller than the setting value of the G0 in-position width "#2077 G0inps" and the G1 in-position width "#2078 G1inps", the in-position check is carried out with the G0 in-position width and the G1 in-position width.



When the SV024 value is larger, the in-position check is completed when the motor position becomes within the specified with SV024.

The in-position check method depends on the method set in the deceleration check parameter.

**(Note 1)** For E68, when the in-position width (programmable in-position check width) is set in the machining program, either the in-position width set with the parameter (SV024, G0inps, G1inps) or that set in the program, whichever larger, is applied when performing an in-position check.

**(Note 2)** When the SV024 setting value is larger than the G0 in-position width/G1 in-position width, the in-position check is carried out with the SV024 value.

## 6. Interpolation Functions

### 6.2 Linear interpolation

#### 6.2 Linear interpolation; G01



##### Function and purpose

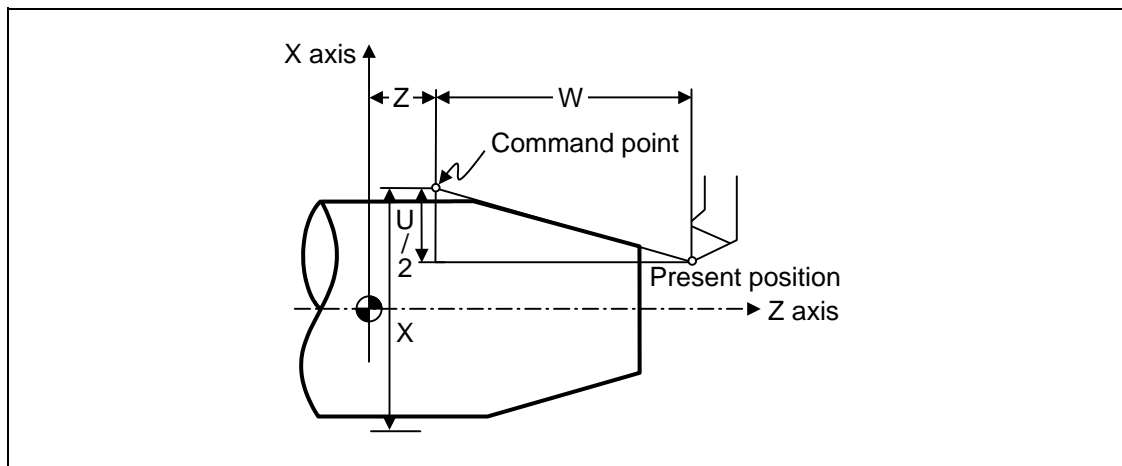
This command is accompanied by coordinate words and a feedrate command. It makes the tool move (interpolate) linearly from its present position to the end point specified by the coordinate words at the speed specified by address F. In this case, the feedrate specified by address F always acts as a linear speed in the tool nose center advance direction.



##### Command format

**G01 Xx/Uu Zz/Ww αα Ff ; ("α" is an additional axis)**

x, u, z, w, α	Coordinate values
f	Feedrate (mm/min or °/min)



##### Detailed description

Once this command is issued, the mode is maintained until another G function (G00, G02, G03, G33, G34) in the 01 group which changes the G01 mode is issued. Therefore, if the next command is also G01 and if the feedrate is the same all that is required to be done is to specify the coordinate words. If no F command is given in the first G01 command block, program error (P62) results.

The feedrate for a rotation axis is commanded by °/min (decimal point position unit). (F300 = 300°/min)

The G functions (G70 ~ G89) in the 09 group are cancelled (G80) by the G01 command.

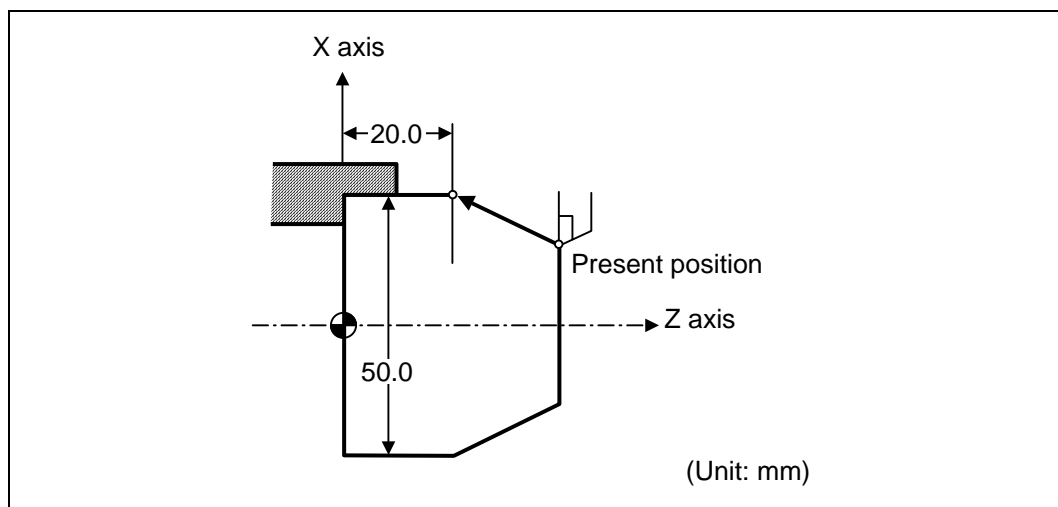
## 6. Interpolation Functions

### 6.2 Linear interpolation



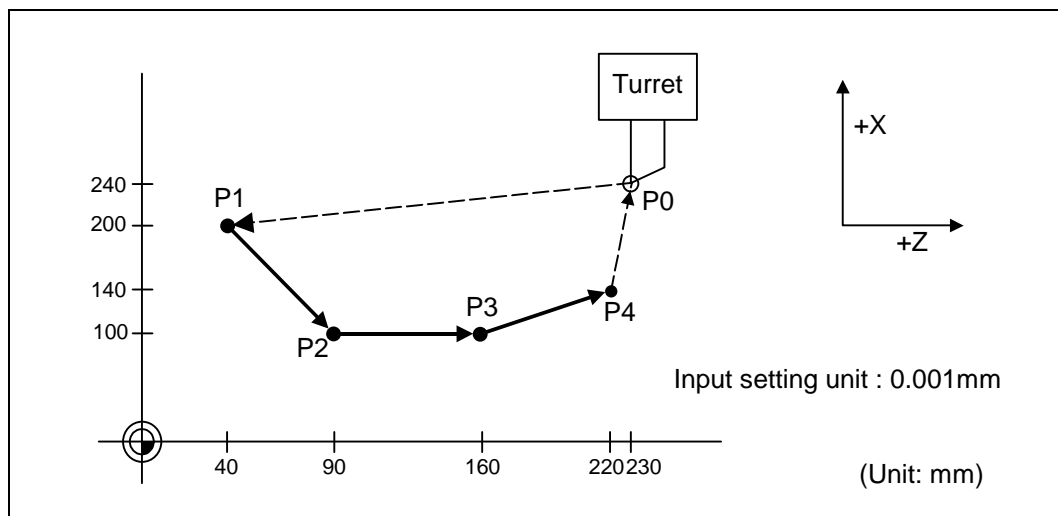
#### Example of program

#### (Example 1)



G01 X50.0 Z20.0 F300 ;	
------------------------	--

#### (Example 2) Cutting in the sequence of P1 → P2 → P3 → P4 at 300mm/min federate P0 → P1 , P4 → P0 is for tool positioning



G00 X200000 Z40000 ;	P0 → P1
G01 X100000 Z90000 F300 ;	P1 → P2
Z160000 ;	P2 → P3
X140000 Z220000 ;	P3 → P4
G00 X240000 Z230000 ;	P4 → P0

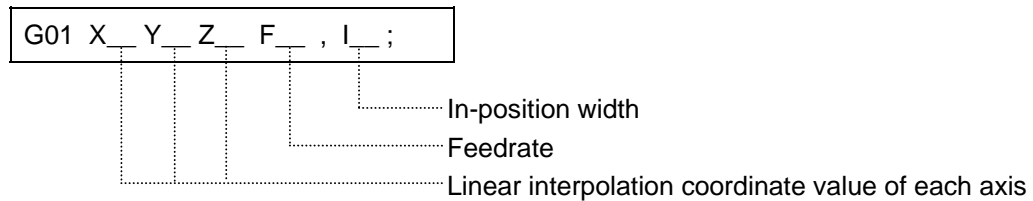



**Programmable in-position width command for linear interpolation**
**[E68]**

This command commands the in-position width for the linear interpolation command from the machining program.

The commanded in-position width is valid in the linear interpolation command only when carrying out deceleration check.

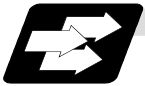
- When the error detect switch is ON.
- When G09 (exact stop check) is commanded in the same block.
- When G61 (exact stop check mode) is selected.



,I__	: In-position width. A program error occurs if a decimal point command is issued. This is valid only in the commanded block. A block that does not contain this address will follow the parameter "#1193 inpos" settings.
------	---

**(Note 1)** Refer to section "6.1 Positioning (rapid traverse); G00" for details on the in-position check operation.

#### 6.3 Circular Interpolation; G02, G03



##### Function and purpose

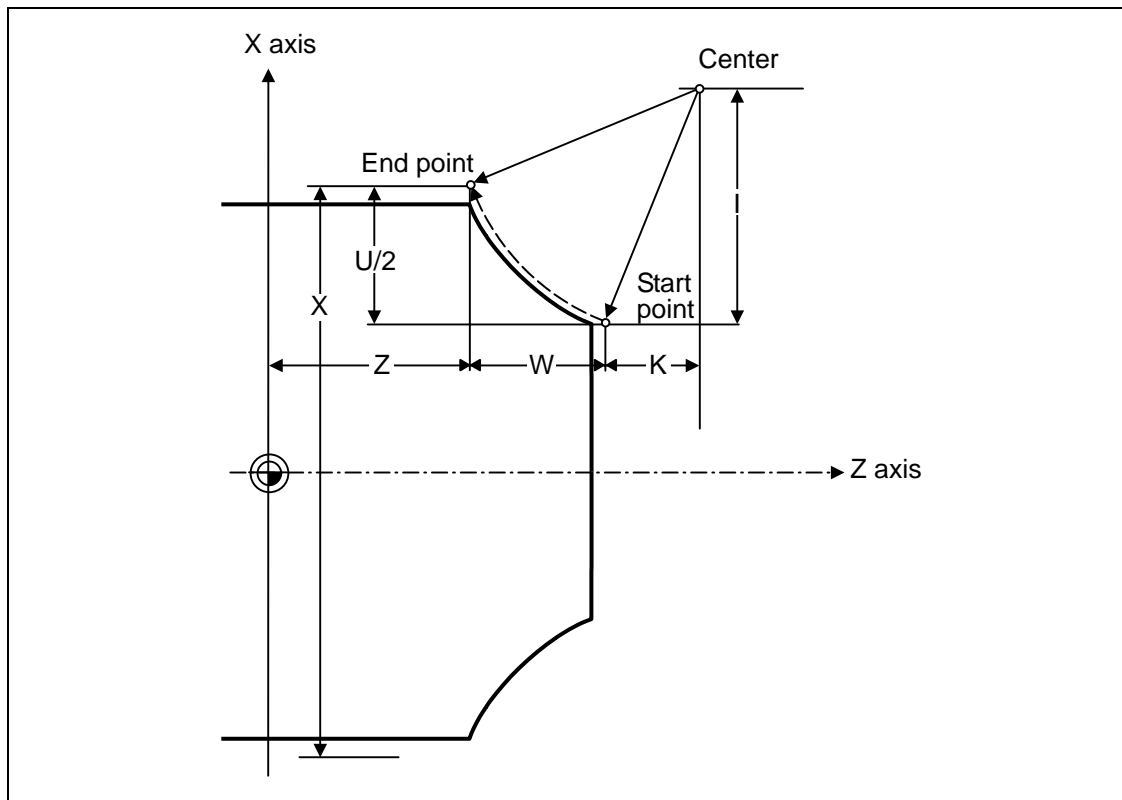
These commands serve to move the tool along a circular.



##### Command format

<b>G02 (G03) Xx/Uu Zz/Ww Ii Kk Ff ;</b>	
G02	Clockwise (CW)
G03	Counterclockwise (CCW)
Xx/Uu	Circular end point coordinates, X axis (absolute value of workpiece coordinate system for X, incremental value from present position for U)
Zz/Ww	Circular end point coordinates, Z axis (absolute value of workpiece coordinate system for Z, incremental value from present position for W)
Ii	Circular center, X axis (for I, radius command/incremental value of X coordinate at center as seen from start point)
Kk	Circular center, Z axis (for K, incremental value of Z coordinate at center as seen from start point)
Ff	Feedrate

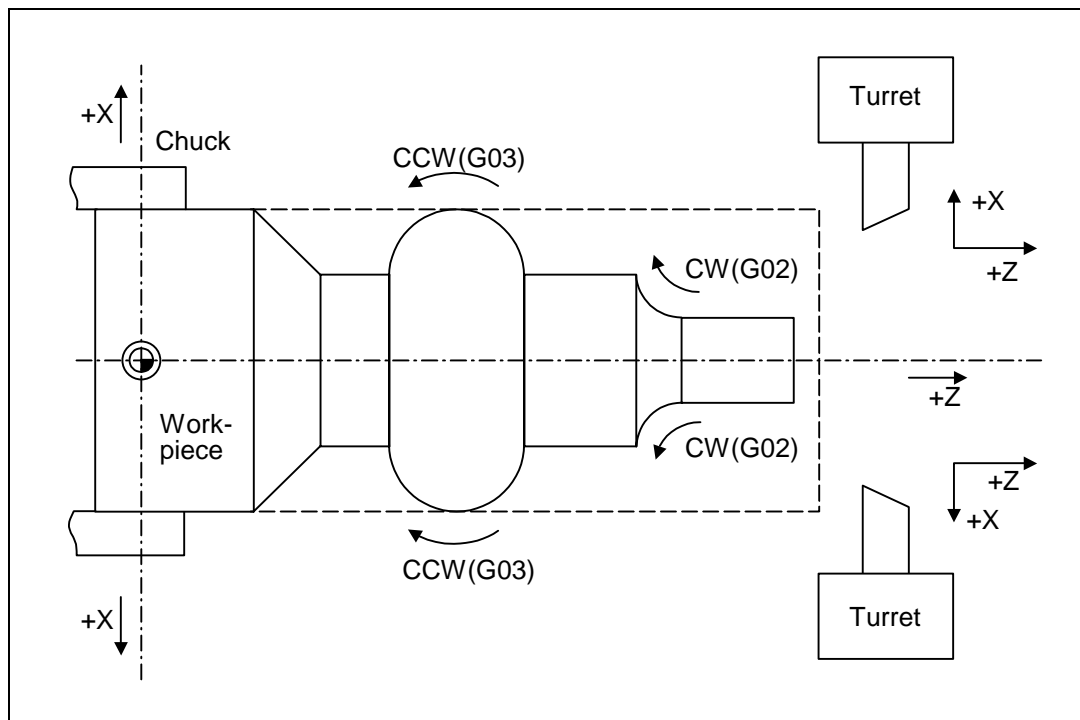
The circular center coordinate value is commanded with an input setting unit. Caution is required for the arc command of an axis for which the input command unit differs. Command with a decimal point to avoid confusion.





#### Detailed description

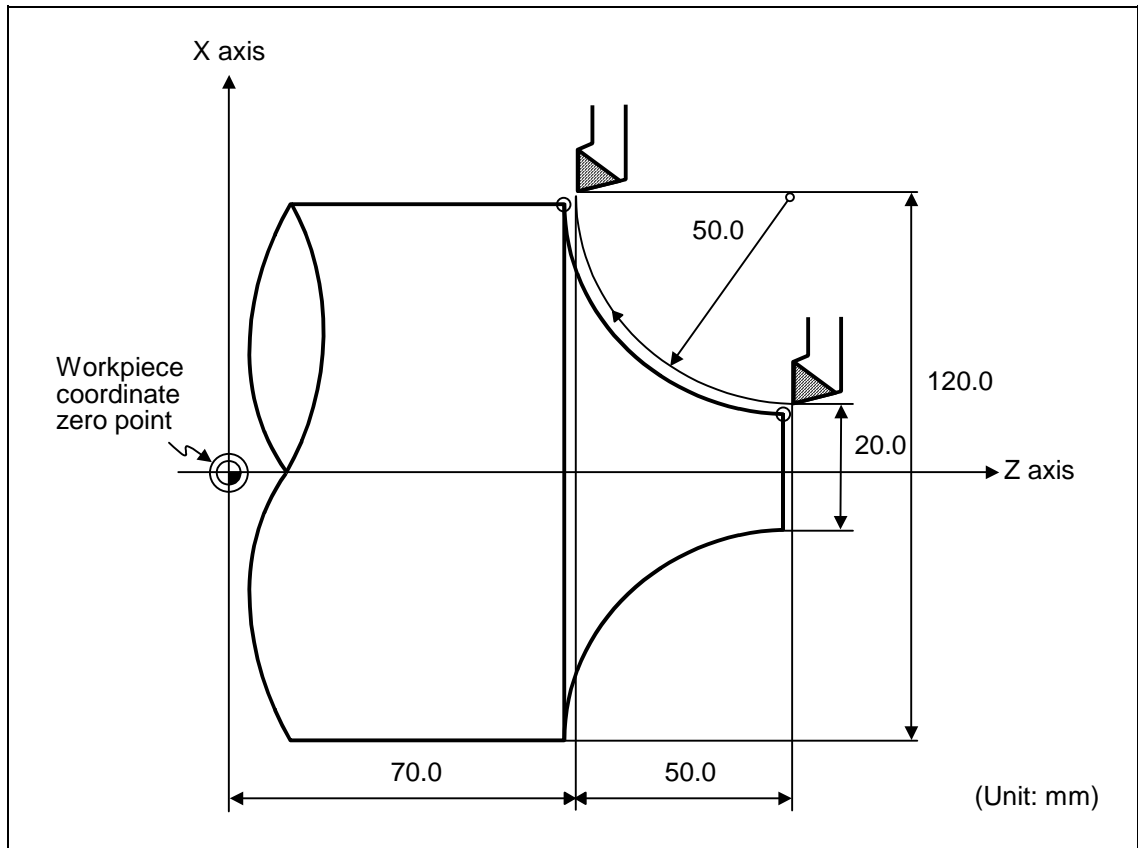
- (1) G02 (or G03) is retained until another G command (G00, G01 or G33) in the 01 group that changes its mode is issued.  
The direction of the circular rotation is differentiated by G02 and G03:  
G02 : Clockwise (CW)  
G03 : Counterclockwise (CCW)



- (2) An arc which extends for more than one quadrant can be executed with a single block command.
- (3) The following information is needed for circular interpolation.
  - (a) Rotation direction :Clockwise (G02) or counterclockwise (G03)?
  - (b) Circular end point coordinates:Given by addresses X, Z, U, W
  - (c) Circular center coordinates :Given by addresses I, K (incremental value commands)
  - (d) Feedrate :Given by address F
- (4) A program error results when I, K or R is not commanded.  
Consideration must be given to the sign for I and K since I is the distance in the X-axis direction to the arc center as seen from the start point and K is the distance in the Z-axis direction.
- (5) No T commands can be issued in the G2/G3 modal status.  
A program error (P151) results if a T command is issued in the G2/G3 modal status.



### Example of program

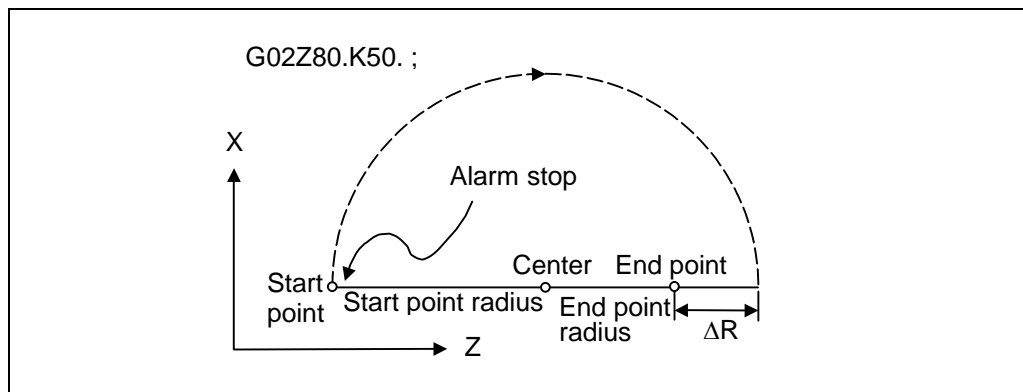


G2 X120.0 Z70.0 I50.0 F200 ;	Absolute value command
G2 U100.0 W-50.0 I50.0 F200 ;	Incremental value command

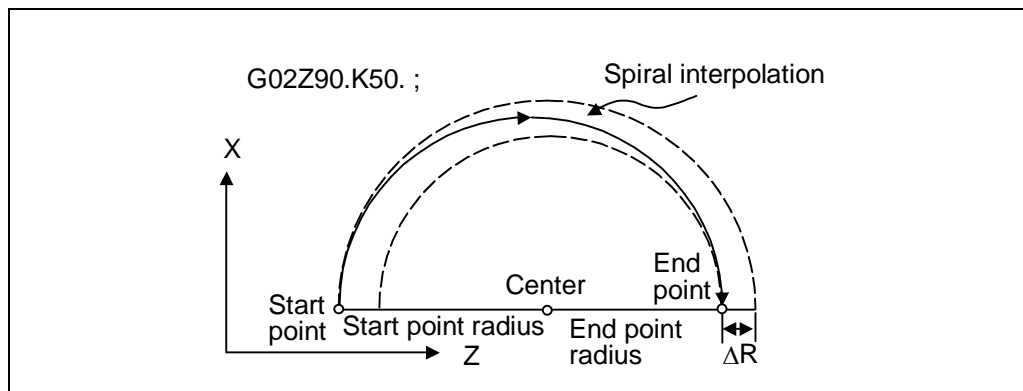


#### Cautions for circular interpolation

- (1) The terms "clockwise" (G02) and "counterclockwise" (G03) used for circular operations are defined as a case where in a right-hand coordinate system, the negative direction is viewed from the positive direction of the coordinate axis which is at right angles to the plane in question.
- (2) If all the end point coordinates are omitted or the end point is at the same position as the start point, commanding the center using I and K is the same as commanding a 360° arc (perfect circle).
- (3) The following occurs when the start and end point radii do not match in a circular command:
  - (a) Program error (P70) results at the circular start point when error  $\Delta R$  is greater than the parameter "#1084 RadErr".



- (b) Spiral interpolation in the direction of the commanded end point results when error  $\Delta R$  is less than the parameter value.



## 6. Interpolation Functions

### 6.4 R specification circular interpolation

#### 6.4 R specification circular interpolation; G02, G03



##### Function and purpose

Along with the conventional circular interpolation commands based on the circular center coordinate (I, K) designation, these commands can also be issued by directly designating the circular radius R.



##### Command format

**G02 (G03) Xx/Uu Zz/Ww Rr Ff;**

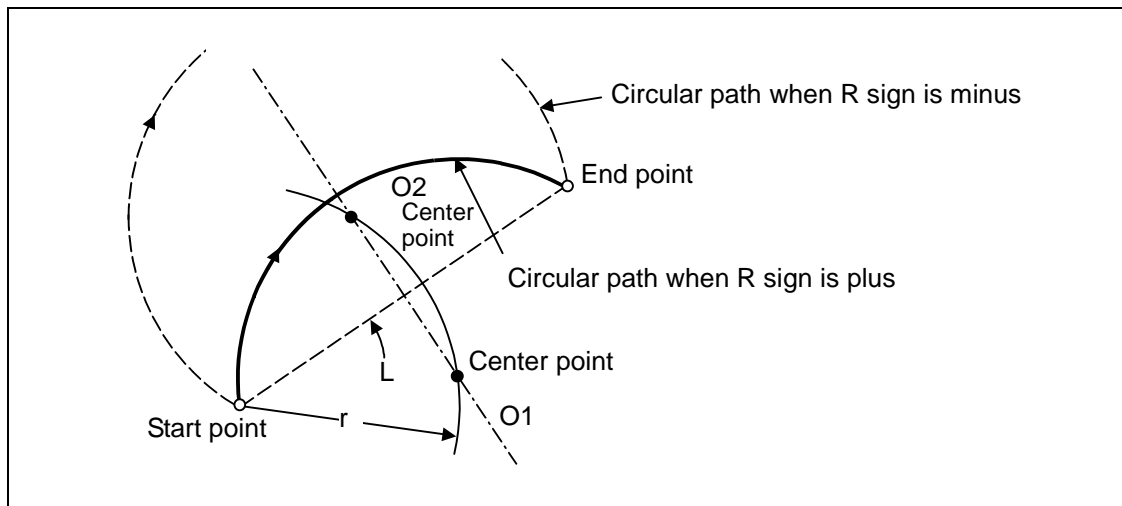
x/u	X-axis end point coordinate
z/w	Z-axis end point coordinate
r	Circular radius
f	Feedrate



##### Detailed description

The circular center is on the bisector line which is perpendicular to the line connecting the start and end points of the circular. The point, where the circular with the specified radius whose start point is the center intersects the perpendicular bisector line, serves as the center coordinates of the circular command.

If the R sign of the commanded program is plus, the circular is smaller than a semicircular; if it is minus, the circular is larger than a semicircular.



The following condition must be met with an R specification circular interpolation command:

$$\frac{L}{2 \times r} \leq 1$$

When  $L/2 - r >$  (parameter : #1084 RadErr), an alarm will occur.

Where L is the line from the start point to end point.

If an R specification and I, K specification are given at the same time in the same block, the circular command with the R specification takes precedence.

In the case of a full-circle command (where the start and end points coincide), an R specification circular command will be completed immediately if it is issued and no operation will result. An I, K specification circular command should therefore be used in such a case.

## 6. Interpolation Functions

### 6.4 R specification circular interpolation



#### Example of program

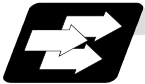
##### (Example 1)

G03 Zz1 Xx1 Rr1 Ff1 ;	R specification circular on Z-X plane
-----------------------	---------------------------------------

##### (Example 2)

G02 Xx1 Zz1 Ii1 Kk1 Rr1 Ff1 ;	R specification circular on X-Z plane (When the R specification and I, K specification are contained in the same block, the R specification has priority in processing.)
-------------------------------	---

#### 6.5 Plane selection; G17, G18, G19



##### Function and purpose

These commands are used to select the control plane and the plane on which the circular exists. If the 3 basic axes and the parallel axes corresponding to these basic axes are entered as parameters, the commands can select the plane composed of any 2 axes which are not parallel axes. If a rotation axis is entered as a parallel axis, the commands can select the plane containing the rotation axis.

These commands are used to select:

- The plane for circular interpolation
- The plane for nose R compensation

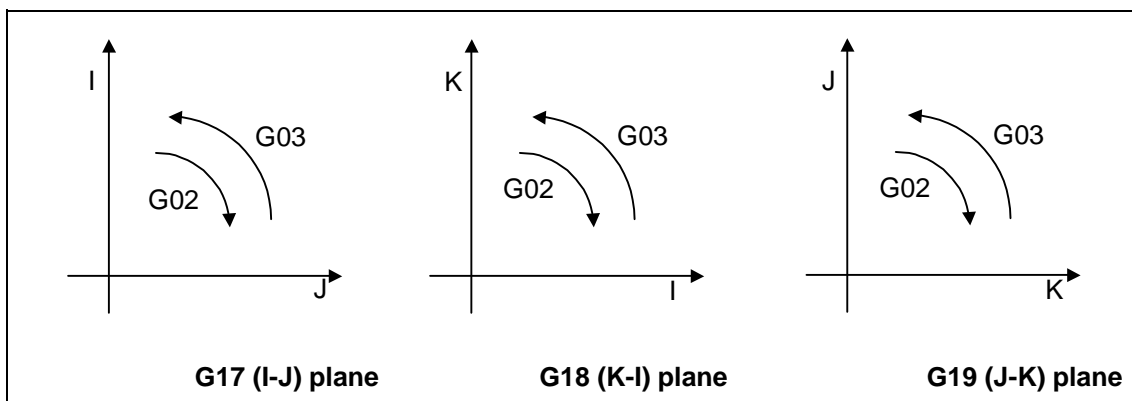


##### Command format

<b>G17;</b>	<b>(I-J plane selection)</b>
<b>G18;</b>	<b>(K-I plane selection)</b>
<b>G19;</b>	<b>(J-K plane selection)</b>

I, J and K indicate each basic axis or parallel axis.

When the power is turned ON or when the system is reset, the plane set by the parameters "#1025 I\_plane" is selected.







### Parameter entry

	#1026~1028 base_I, J, K	#1029~1031 aux_I, J, K
I	X	Y
J	Y	
K	Z	

**Fig. 1 Examples of plane selection parameter entry**

Basic axes and parallel axes can be entered in the parameters. The same axis name can be entered in duplication, but when it is assigned in duplication, the plane is determined by plane selection system (4).

It is not possible to set axes, which have not been entered, as control axes.

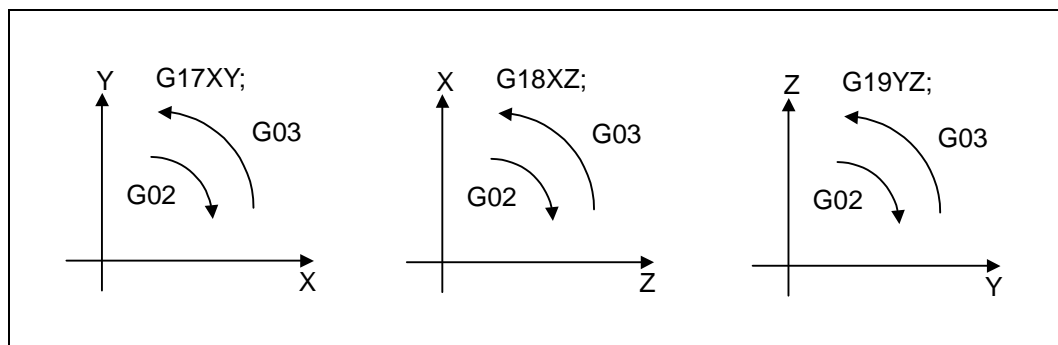


### Plane selection system

This section describes the plane selection for the parameter entry samples shown in Fig. 1.

- (1) Axis addresses assigned in the same block as the plane selection (G17, G18, G19) command determine which of the basic axes or parallel axes are to be in the actual plane selected.

**(Example)**



- (2) Plane selection is not performed with blocks in which the plane selection G code (G17, G18, G19) is not assigned.

G18 X\_ Z\_ ;     Z-X plane  
 Y\_ Z\_ ;     Z-X plane (no plane change)

- (3) When the axis addresses are omitted in the block containing the plane selection G codes (G17, G18, G19), it is assumed that the axis addresses of the 3 basic axes have been assigned.

G18 ; (Z-X plane = G18 XZ ;)

- (4) When the basic axes or their parallel axes are duplicated and assigned in the same block as the plane selection G code (G17, G18, G19), the plane is determined in the order of basic axes, and then parallel axes.

G18 XYZ ;     The Z-X plane is selected.  
 Therefore, the Y movement is unrelated to the selected plane.

- (Note 1)** When the "2" in the parameter "#1025 I\_plane" is kept ON, the G18 plane is selected when the power is turned ON or when the system is reset.

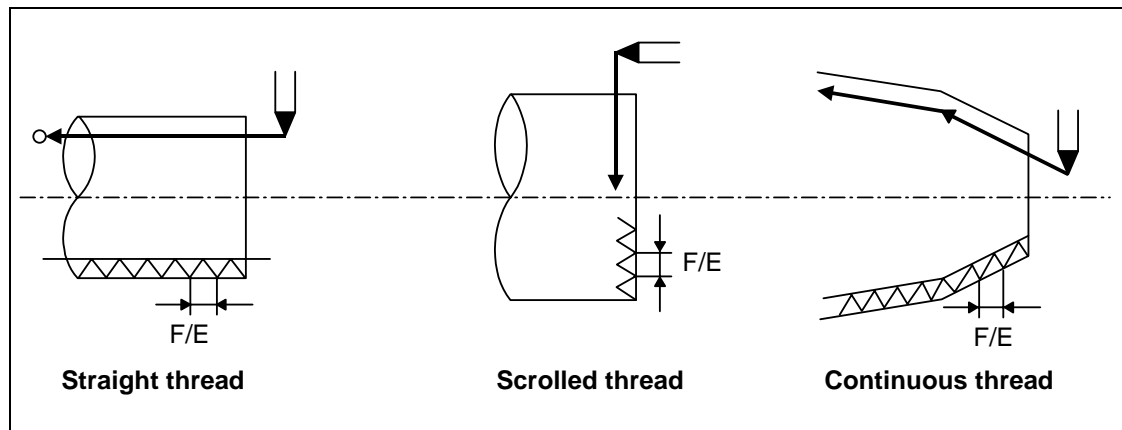
#### 6.6 Thread cutting

##### 6.6.1 Constant lead thread cutting; G33



#### Function and purpose

The G33 command exercises feed control over the tool which is synchronized with the spindle rotation and so this makes it possible to conduct constant-lead straight thread-cutting, tapered thread-cutting, and continuous thread-cutting.



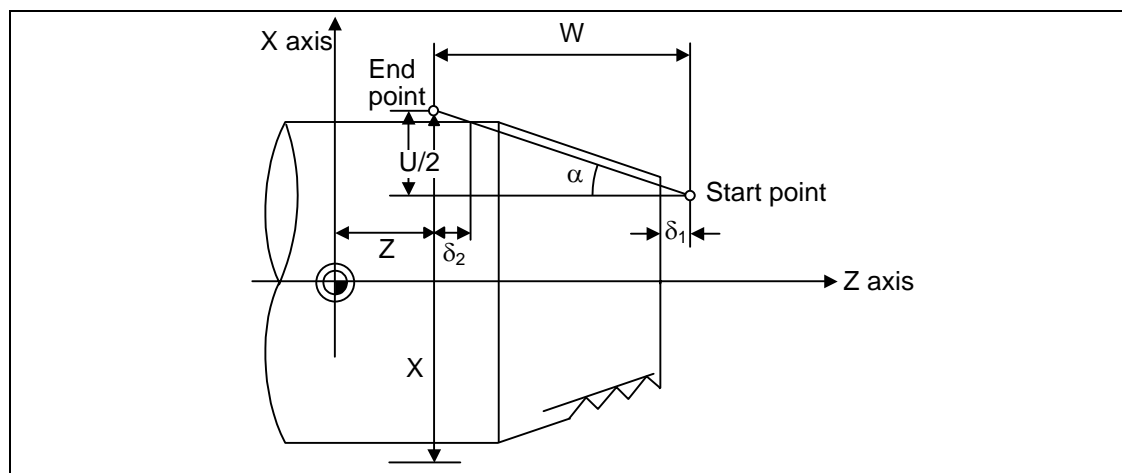
#### Command format

##### G33 Zz/Ww Xx/Uu Ff Qq ; (Normal lead thread cutting commands)

Zz, Ww, Xx, Uu Thread end point addresses and coordinates  
 Ff Lead of long axis (axis which moves most) direction  
 Qq Thread cutting start shift angle, 0.001 to 360.000°

##### G33 Zz/Ww Xx/Uu Ee Qq ; (Precision lead thread cutting commands)

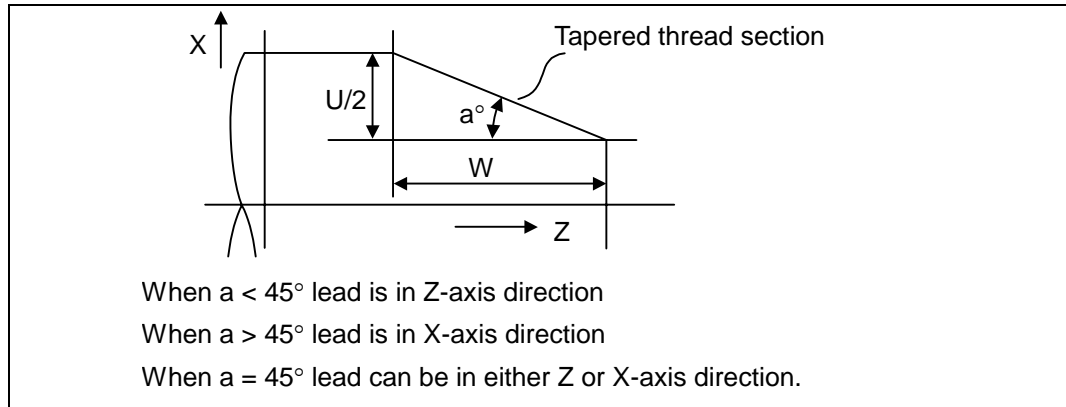
Zz, Ww, Xx, Uu Thread end point addresses and coordinates  
 Ee Lead of long axis (axis which moves most) direction  
 Qq Thread cutting start shift angle, 0.001 to 360.000°





### Detailed description

- (1) The E command is also used for the number of ridges in inch thread cutting, and whether the ridge number or precision lead is to be designated can be selected by parameter setting. (Parameter "#1229 set 01/bit" is set to "1" for precision lead designation).
- (2) The lead in the long axis direction is commanded for the taper thread lead.



- (a) When using E60

Input unit system	Thread cutting metric input			Thread cutting inch input		
	0.001mm			0.0001inch		
Command address	F (mm/rev)	E (mm/rev)	E (threads/ inch)	F (inch/rev)	E (inch/rev)	E (threads/ inch)
Minimum command unit	1 (= 0.0001), (1.=1.0000)	1 (= 0.00001), (1.=1.00000)	1 (= 1.00), (1.=1.00)	1(=0.000001), (1.=1.000000)	1(=0.0000001), (1.=1.0000000)	1 (= 1), (1.=1.0000)
Command range	0.0001 to 999.9999	0.00001 to 999.99999	0.03 to 999.99	0.000001 to 99.999999	0.000010 to 9.9999999	0.0101 to 9999.9999

- (b) When using E68

Input unit system	Thread cutting Metric input			Thread cutting Inch input		
	B (0.001mm)			C (0.0001inch)		
Command address	F (mm/rev)	E (mm/rev)	E (threads /inch)	F (mm/rev)	E (mm/rev)	E (threads / inch)
Minimum command unit	1(=0.0001) (1.=1.0000)	1(=0.0001) (1.=1.00000)	1(=1) (1.=1.00)	1(=0.00001) (1.=1.00000)	1(=0.000001) (1.=1.000000)	1(=1) (1.=1.0000)
Command range	0.0001 to 999.9999	0.00001 to 999.99999	0.03 to 999.99	0.00001 to 99.99999	0.000001 to 99.999999	0.255 to 9999.999

Input unit system	Thread cutting Metric input			Thread cutting Inch input		
	B (0.001mm)			C (0.0001inch)		
Command address	F (mm/rev)	E (mm/rev)	E (threads /inch)	F (mm/rev)	E (mm/rev)	E (threads / inch)
Minimum command unit	1(=0.00001) (1.=1.000000)	1(=0.000001) (1.=1.0000000)	1(=1) (1.=1.0000)	1(=0.0000001) (1.=1.0000000)	1(=0.00000001) (1.=1.00000000)	1(=1) (1.=1.00000)
Command range	0.000001 to 99.999999	0.000010 to 9.9999999	0.0101 to 9999.9999	0.0000001 to 9.9999999	0.00000001 to 0.99999999	0.10001 to 999.99999

**(Note 1)** It is not possible to assign a lead where the feedrate as converted into per-minute feed exceeds the maximum cutting feedrate.

- (3) The constant surface speed control function should not be used for taper thread cutting commands or scrolled thread cutting commands.
- (4) The spindle rotation speed should be kept constant throughout from the rough cutting until the finishing.
- (5) If the feed hold function is employed during thread cutting to stop the feed, the thread ridges will lose their shape. For this reason, feed hold does not function during thread cutting. Note that this is valid from the time the thread cutting command is executed to the time the axis moves.  
If the feed hold switch is pressed during thread cutting, block stop will result at the end point of the block following the block in which thread cutting is completed (no longer G33 mode).
- (6) The converted cutting feedrate is compared with the cutting feed clamp rate when thread cutting starts, and if it is found to exceed the clamp rate, an operation error will result.
- (7) In order to protect the lead during thread cutting, a cutting feedrate which has been converted may sometimes exceed the cutting feed clamp rate.
- (8) An illegal lead is normally produced at the start of the thread and at the end of the cutting because of servo system delay and other such factors.  
Therefore, it is recommended to program a thread length which is determined by adding the illegal lead to the required thread length.  

$$\text{Illegal lead} = \frac{\text{Maximum feedrate}}{\text{Thread lead}}$$
- (9) The spindle rotation speed is subject to the following restriction:  

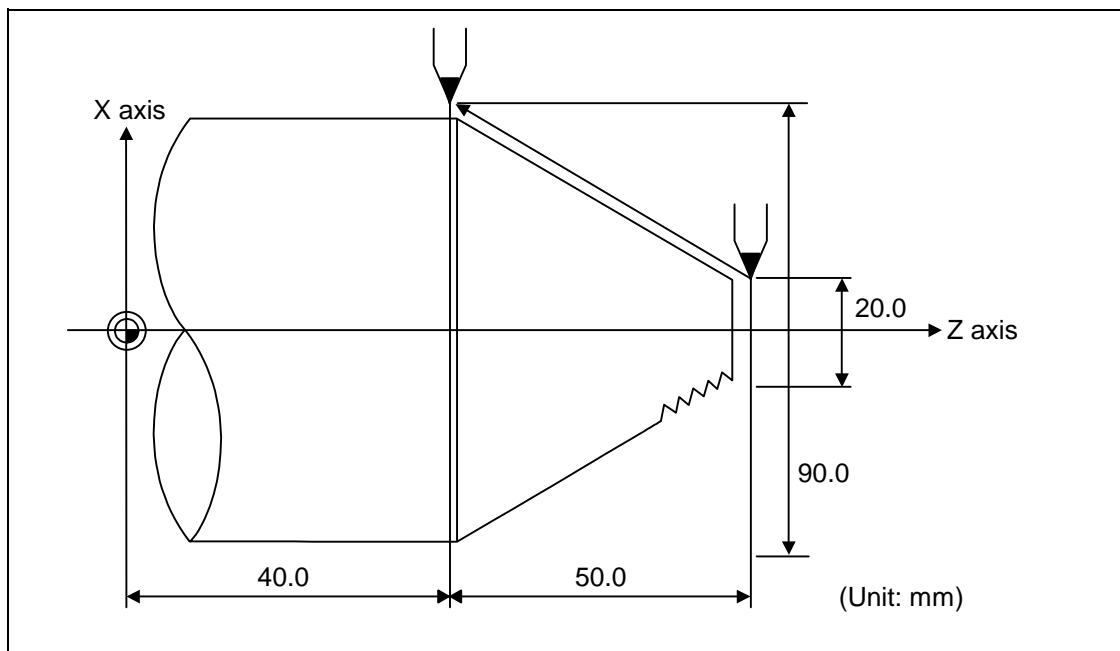
$$1 \leq R \leq$$

Where  $R \leq$  Tolerable speed of encoder (r/min)  
 $R =$  Spindle rotation speed (r/min)  
 Thread lead = mm or inches  
 Maximum feedrate = mm/min or inch/mm  
 (this is subject to the restrictions imposed by the machine specifications).
- (10) A program error (P97) may occur when the result of the expression (9) is  $R < 1$  because the thread lead is very large to the highest cutting feedrate.
- (11) Dry run is valid for thread cutting but the feedrate based on dry run is not synchronized with the spindle rotation.  
The dry run signal is checked at the start of thread cutting and any switching during thread cutting is ignored.
- (12) Synchronous feed applies for the thread cutting commands even with an asynchronous feed command (G94).
- (13) Spindle override or cutting feed override is invalid and 100% fixed during thread cutting.
- (14) When a thread cutting command is programmed during nose R compensation, the compensation is temporarily canceled and the thread cutting is executed.
- (15) When the mode is switched to another automatic mode while G33 is executed, the following block which does not contain a thread cutting command is first executed and then the automatic operation stops.
- (16) When the mode is switched to the manual mode while G33 is executed, the following block which does not contain a thread cutting command is first executed and then the automatic operation stops. In the case of a single block, the following block which does not contain a thread cutting command (When G33 mode is cancelled) is first executed and then the automatic operation stops. Note that automatic operation is stopped until the G33 command axis starts moving.

- (17) The thread cutting command waits for the single rotation sync signal of the rotary encoder and starts movement.
- (18) The automatic handle interrupt/interruption is valid during thread cutting.
- (19) The thread cutting start shift angle is not a modal. If there is no Q command with G33, this will be handled as "Q0".
- (20) If a value exceeding 360.000 is commanded with G33 Q, a program error (P35) will occur.
- (21) G33 cuts one row with one cycle. To cut two rows, change the Q value, and issue the same command.



#### Example of program



G33 X90.0 Z40.0 E12.34567 ;	Absolute value command
G33 U70.0 W-50.0 E12.34567 ;	Incremental value command

#### 6.6.2 Inch thread cutting; G33



#### Function and purpose

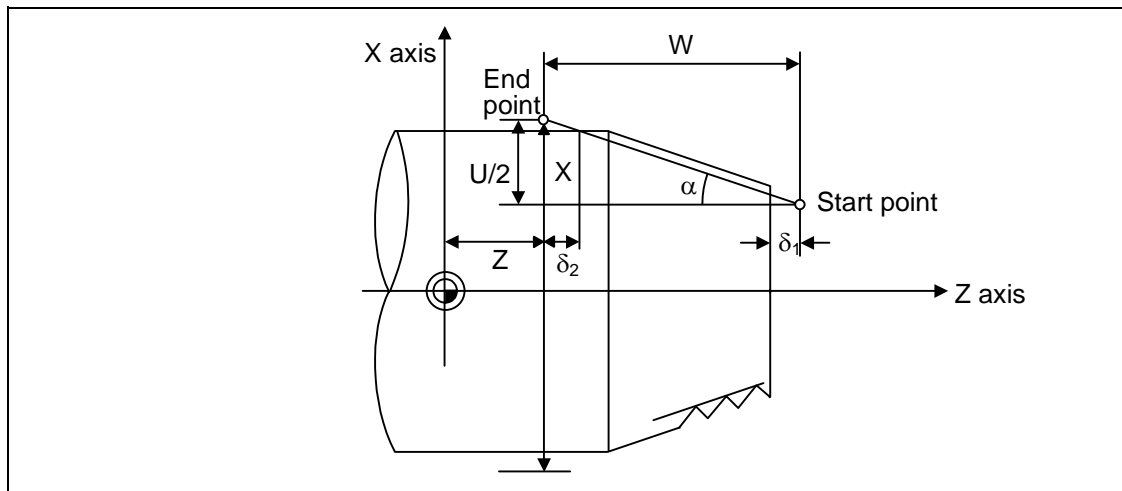
If the number of ridges per inch in the long axis direction is assigned in the G33 command, the feed of the tool synchronized with the spindle rotation will be controlled, which means that constant-lead straight thread-cutting, tapered thread-cutting, and continuous thread-cutting can be performed.



#### Command format

##### G33 Zz/Ww Xx/Uu Ee Qq;

Zz,Ww,Xx,Uu	Thread end point addresses and coordinates
Ee	Number of ridges per inch in direction of long axis (axis which moves most) (decimal point command can also be assigned)
Qq	Thread cutting start shift angle, 0.001 to 360.000°

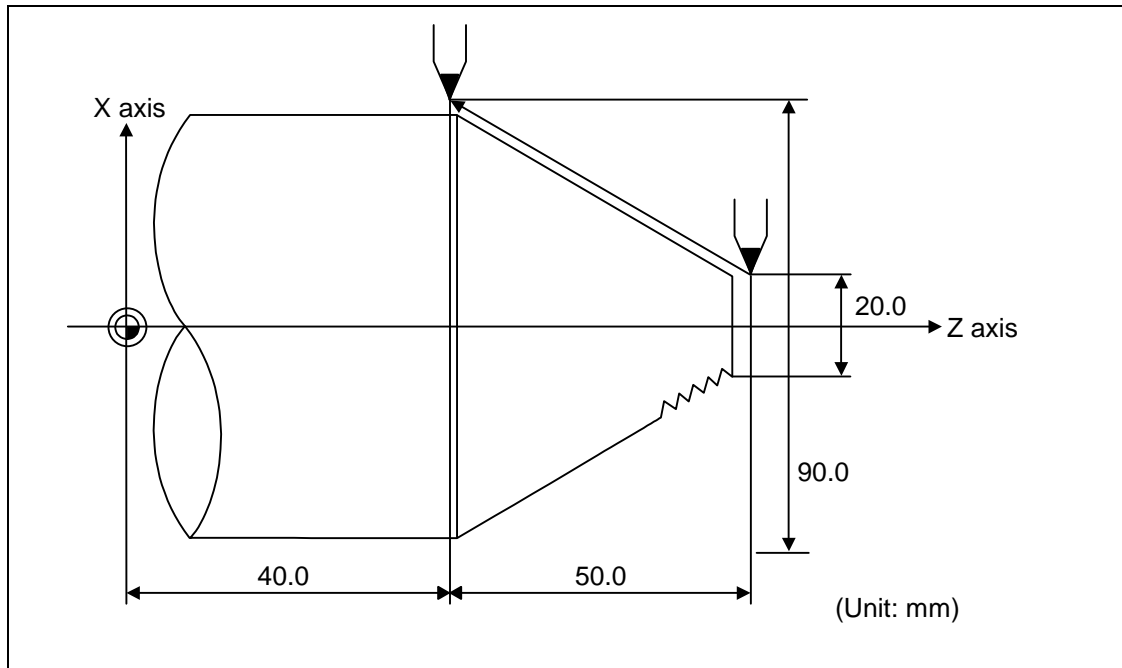


#### Detailed description

- (1) The number of ridges in the long axis direction is assigned as the number of ridges per inch.
- (2) The E code is also used to assign the precision lead length, and whether the ridge number or precision lead length is to be designated can be selected by parameter setting. (The number of ridges is designated by setting parameter "#1229 set 01/bit 1" to "0".)
- (3) The E command value should be set within the lead value range when the lead is converted.
- (4) See Section "6.6.1 Constant lead thread cutting" for other details.



### Example of program



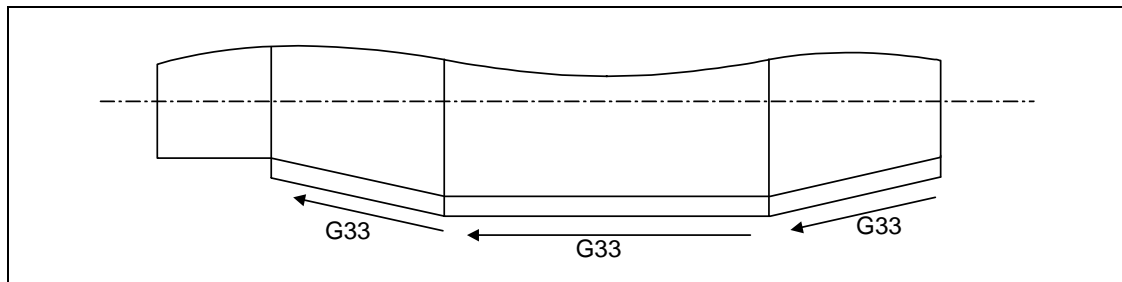
G33 X90.0 Z40.0 E12.0 ;	Absolute value command
G33 U70.0 W-50.0 E12.0 ;	Incremental value command

### 6.6.3 Continuous thread cutting

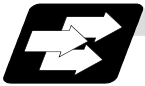


#### Function and purpose

Continuous thread cutting is possible by assigning thread cutting commands continuously. In this way, it is possible to cut special threads whose lead or shape changes.



#### 6.6.4 Variable Lead Thread Cutting; G34



##### Function and purpose

Variable lead thread cutting is enabled by a command specifying a lead increment or decrement amount per turn of the screw.



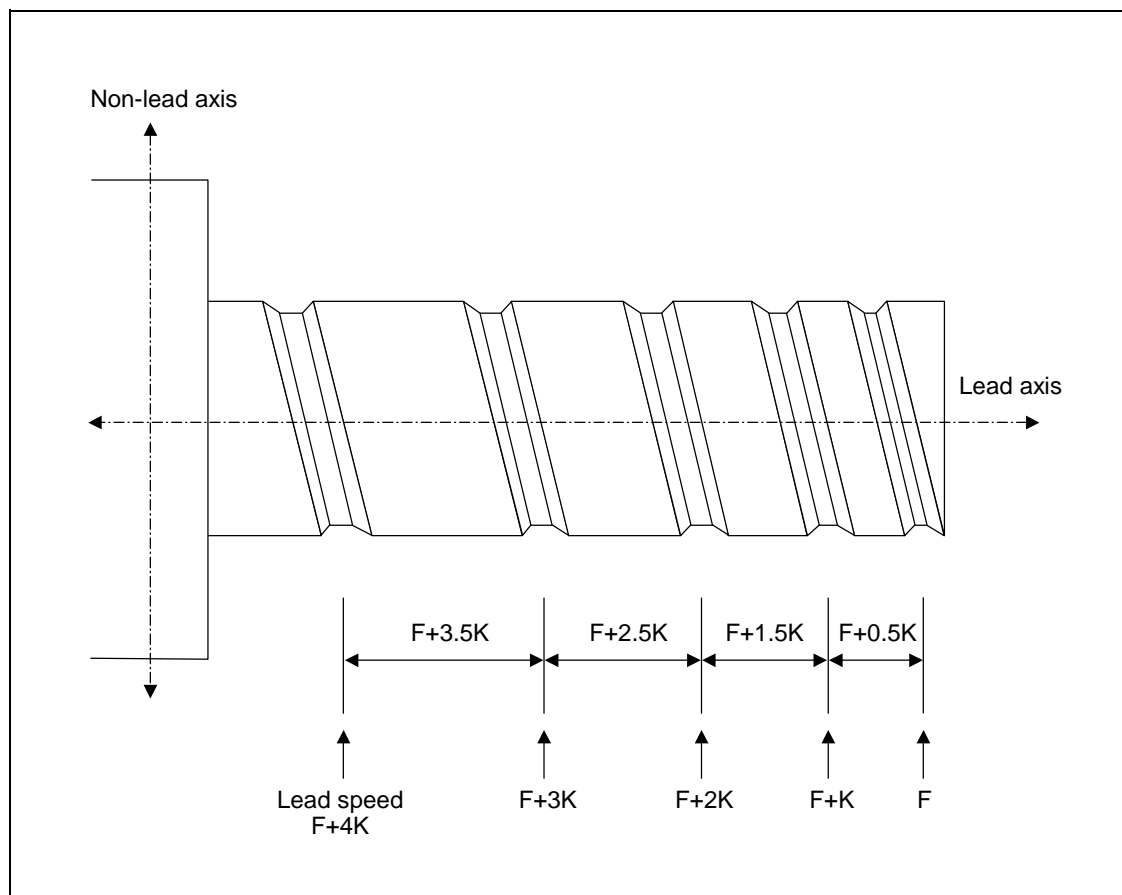
##### Command format

**G34 Xx/Uu Zz/Ww Ff/Ee Kk ;**

Xx/Uu Zz/Ww Thread end point addresses and coordinates

Ff/Ee Standard screw lead

Kk Lead increment or decrement amount per turn of the screw







### Detailed description

(1) The command range is as shown below.

(a) When using E60

Input unit system	Thread cutting metric input		Thread cutting inch input		B
	B (0.001mm)		B (0.0001inch)		
Command address	F (inch/rev)	E (inch/rev)	F (inch/rev)	E (inch/rev)	K (n * mm/rev) n: Number of pitches Same as F or E (signed)
Minimum command unit	1(=0.0001) (1.=1.0000)	1(=0.00001) (1.=1.00000)	1(=0.000001) (1.=1.000000)	1(=0.0000001) (1.=1.0000000)	
Command range	0.0001 to 999.9999	0.00001 to 999.99999	0.000001 to 99.999999	0.0000001 to 9.9999999	

(b) When using E68

#### Thread cutting metric input

Input unit system	B (0.001mm)		C (0.0001mm)		B/C
	Command address	F (mm/rev)	E (mm/rev)	F (mm/rev)	
Minimum command unit	1 (=0.0001) (1.=1.0000)	1 (=0.00001) (1.=1.00000)	1 (=0.00001) (1.=1.00000)	1 (=0.000001) (1.=1.000000)	
Command range	0.0001 to 999.9999	0.00001 to 999.99999	0.00001 to 99.99999	0.000001 to 99.999999	

#### Thread cutting inch input

Input unit system	B (0.0001inch)		C (0.00001inch)		B/C
	Command address	F (inch/rev)	E (inch/rev)	F (inch/rev)	
Minimum command unit	1 (=0.000001) (1.=1.000000)	1 (=0.0000001) (1.=1.0000000)	1 (=0.0000001) (1.=1.0000000)	1 (=0.00000001) (1.=1.00000000)	
Command range	0.000001 to 99.999999	0.0000001 to 9.9999999	0.0000001 to 9.9999999	0.00000001 to 0.99999999	

(2) A positive value of K indicates incremental pitches.

Movement amount of one block (n pitches) = (F + K) + (F + 2K) + (F + 3K) + ... + (F + nK)

(3) A negative value of K indicates decremental pitches.

Movement amount of one block (n pitches) = (F - K) + (F - 2K) + (F - 3K) + ... + (F - nK)

(4) A program error will occur if the thread lead is not set correctly.

Error number	Meaning	Remedy
P93	<b>Illegal pitch value</b> 1) An invalid value is specified for F/E or K in a thread cutting command. 2) The last lead goes outside of the F/E command range.	Specify valid values for F/E and K. <b>(Reference 1)</b>

**Reference 1)** Last lead =  $\sqrt{F^2 + 2KZ}$

Number of pitches =  $(-F + \text{last lead})/K$

Z: Length of lead axis

(5) The other matters are the same as G33.

Refer to section "6.6.1 Constant lead thread cutting; G33".



#### Precautions

The maximum thread length in the variable lead thread cutting function follows the program format. The following table shows the maximum thread lengths. A program error (P60) will occur if these values are exceeded.

<b>Command unit (mm)</b>	<b>Metric command (mm)</b>	<b>Inch command (inch)</b>
0.001	10737.377	10737.377
0.0001 (For only E68)	1073.7377	1073.7377

#### 6.7 Helical Interpolation; G17, G18, G19, and G02, G03

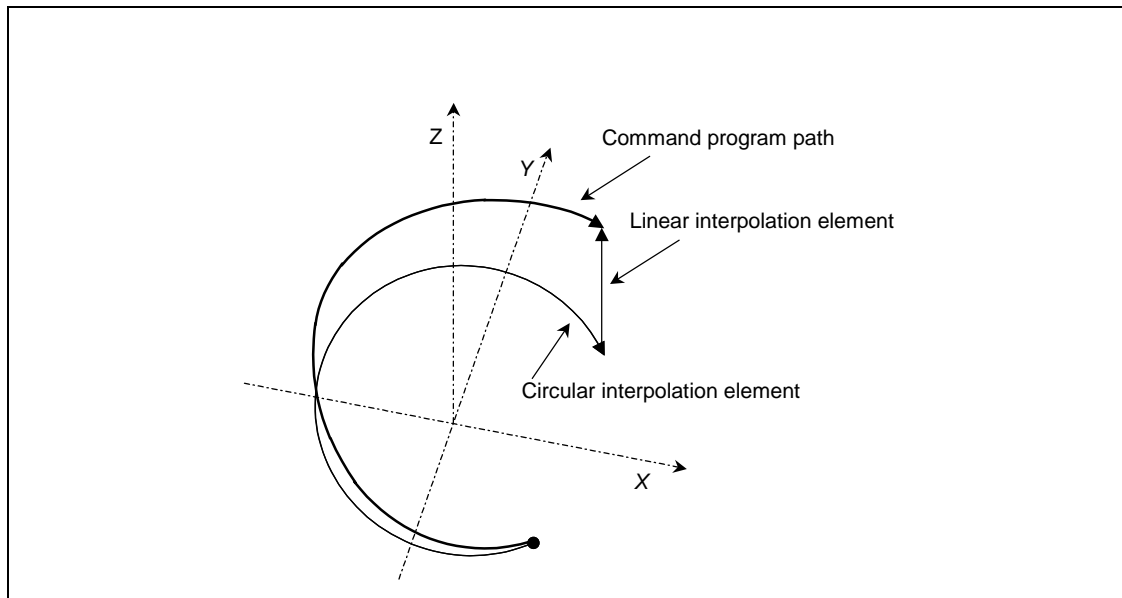
[E68]



#### Function and purpose

This function is for circularly interpolating 2 axes on the selected plane and simultaneously interpolating the other axis linearly in synchronization with the circular motion.

When this interpolation is performed with 3 orthogonal axes, the tool will travel helically.



#### Command format

**G17 G02 (G03) Xx/Uu Yy/Vv Zz/Ww Ii Jj Ff ;**

**G17 G02 (G03) Xx/Uu Yy/Vv Zz/Ww Rr Ff ;**

G17	Arc plane (G17: X-Y plane, G18: Z-X plane, G19: Y-Z plane)
G02 (G03)	Arc rotation direction (G02: clockwise, G03: counterclockwise)
Xx/Uu, Yy/Vv	Arc end point coordinates
Zz/Ww	Linear axis end point coordinates
Ii, Jj	Arc center coordinates
Rr	Arc radius
Ff	Feedrate

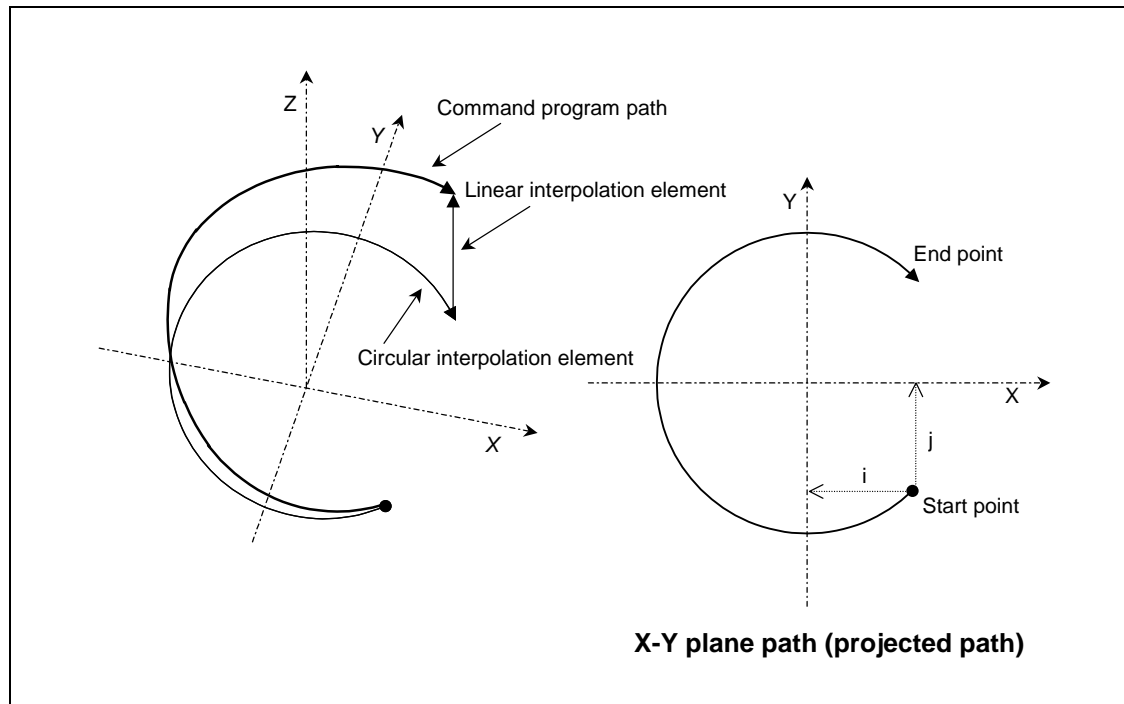
**(Note 1)** In this manual, the following setting descriptions are used: I axis: X, J axis: Y, K axis: Z



#### Detailed description

The following type of movement will take place when the following type of command is issued.

`G17 G02 Xx Yy Zz Ii Jj Ff ;`



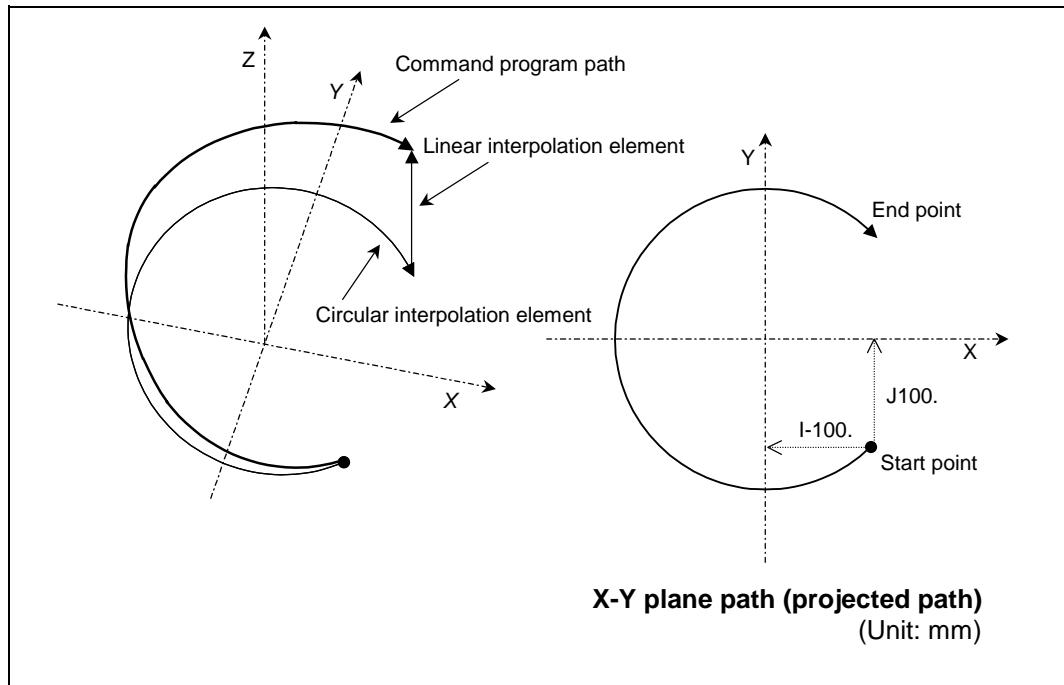
The left drawing shows the process as an exploded view, and the right drawing shows the arc plane from directly above.



## Example of program

(Example)

```
G17 G02 X100. Y100. Z100. I-100. J100. F120 ;
```



The left drawing shows the process as an exploded view, and the right drawing shows the arc plane from directly above.

At the start of the block, the axis centers at the point -100mm in the X axis direction and 100mm in the Y axis direction from the workpiece coordinates (start point), and starts cutting at the feedrate 120mm/min while rotating.



## Precautions and restrictions

- (1) When executing helical interpolation, command another linear axis (several axis can be commanded) that does not contain the circular interpolation command and arc axis.
- (2) Up to the number of simultaneous contouring control axes can be commanded simultaneously.
- (3) A command exceeding one rotation cannot be issued. (The circular interpolation command specifications are followed.)
- (4) Command the feedrate as the composite speed for each axis.
- (5) With helical interpolation, the axis that configures the plane is the circular interpolation axis, and the other axes are the linear interpolation axes.
- (6) Drawing is not performed in the graphic check. (Accurate drawing is not possible.)
- (7) The corner chamfering or corner R commands that are issued before or after the helical interpolation command block are effective only on the axes contained in the selected plane.
- (8) For the parameters and error messages, refer to the materials concerning the circular interpolation (G02,G03)

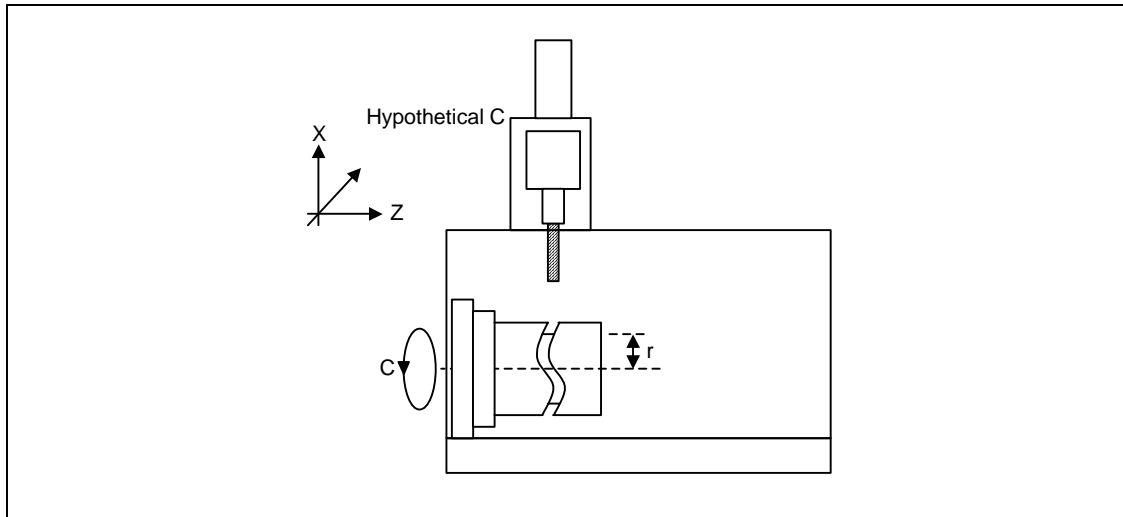
### 6.8 Cylindrical Interpolation; G07.1 (6 and 7 Only in G Code System)

[E68]

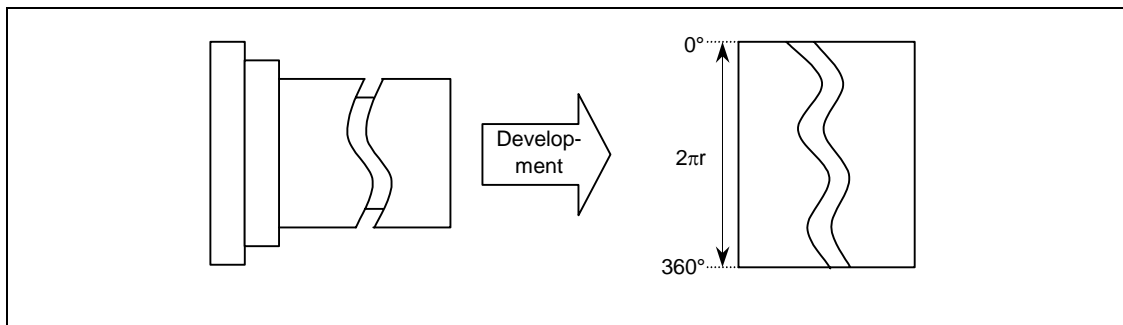


#### Function and purpose

This function develops a shape with a cylindrical side (shape in cylindrical coordinate system) into a plane. When the developed shape is programmed as the plane coordinates, that is converted into the linear axis and rotation axis movement in the cylindrical coordinates and the contour is controlled during machining.



As programming can be carried out with a shape with which the side on the cylinder is developed, this is effective for machining cylindrical cams, etc. When programmed with the rotation axis and its orthogonal axis, slits, etc., can be machined on the cylinder side.





## Command format

<b>G07.1 Cc ;</b>	<b>(Cylindrical interpolation mode start/cancel)</b>
Cc	Cylinder radius value
	<ul style="list-style-type: none"> <li>• Radius value <math>\neq</math> 0: Cylindrical interpolation mode start</li> <li>• Radius value = 0: Cylindrical interpolation mode cancel</li> </ul>

**(Note)** The above format applies when the name of the rotation axis is "C". If the name is not "C", command the name of the rotation axis being used instead of "C".

- (1) The coordinates commanded in the interval from the start to cancellation of the cylindrical interpolation mode will be the cylindrical coordinate system.
 

G07.1 C	Cylinder radius value;	Cylindrical interpolation mode start
:		(Cylindrical interpolation will start)
:		(The coordinate commands in this interval will be the cylindrical coordinate system)
:		
G07.1 C0 ;		Cylindrical interpolation mode cancel
		(Cylindrical interpolation will be canceled)
- (2) G107 can be used instead of G07.1.
- (3) Command G07.1 in an independent block. A program error (P33) will occur if this command is issued in the same block as another G code.
- (4) Program the rotation axis with an angle degree.
- (5) Linear interpolation or circular interpolation can be commanded during the cylindrical interpolation mode. Note that the plane selection command must be issued just before the G07.1 block.
- (6) The coordinates can be commanded with either an absolute command or incremental command.
- (7) Tool radius compensation can be applied on the program command. Cylindrical interpolation will be executed on the path after tool radius compensation.
- (8) Command the segment feed in the cylinder development with F. The F unit is mm/min or inch/min.



## Detailed description

**(1) Cylindrical interpolation accuracy**

In the cylindrical interpolation mode, the movement amount of the rotation axis commanded with an angle is converted on the circle periphery, and after operating the linear and circular interpolation between the other axes, the amount is converted into an angle again.

Thus, the actual movement amount may differ from the commanded value such as when the cylinder radius is small.

Note that the error generated at this time is not cumulated.

#### (2) Related parameters

#	Item	Details	Setting range
1516	mill_ax Milling axis name	Set the name of the rotation axis for pole coordinate interpolation, cylindrical interpolation. Only one of the rotation axes can be set.	A to Z
8111	Milling Radius	Select the diameter and radius of the linear axis for pole coordinate interpolation, cylindrical interpolation. 0: Radius command for all axes 1: Each axis setting (follows #1019 dia diameter designation axis)	0 / 1
1267 (PR)	ext03 (bit0) G code type	The type of high-speed high-accuracy G code is changed. 0: G61.1 type 1: G8 type	0 / 1
1270 (PR)	ext06 (bit7) Handle C axis coordinate during cylindrical interpolation	Specify whether the rotary axis coordinate before the cylindrical interpolation start command is issued is kept during the cylindrical interpolation or not. 0: Do not keep 1: keep	0 / 1

#### (3) Plane selection

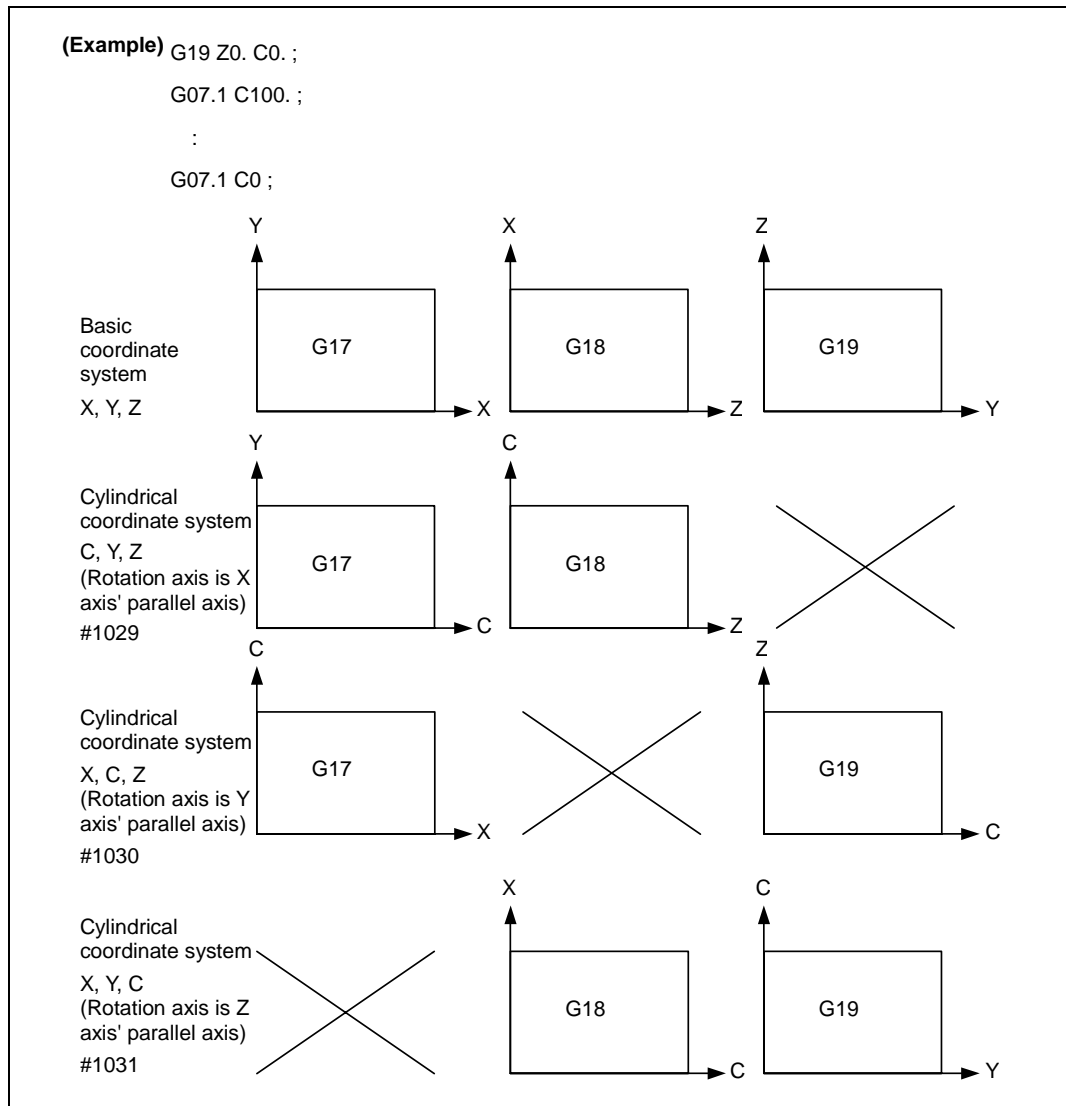
The axis used for cylindrical interpolation must be set with the plane selection command.

##### (Note)

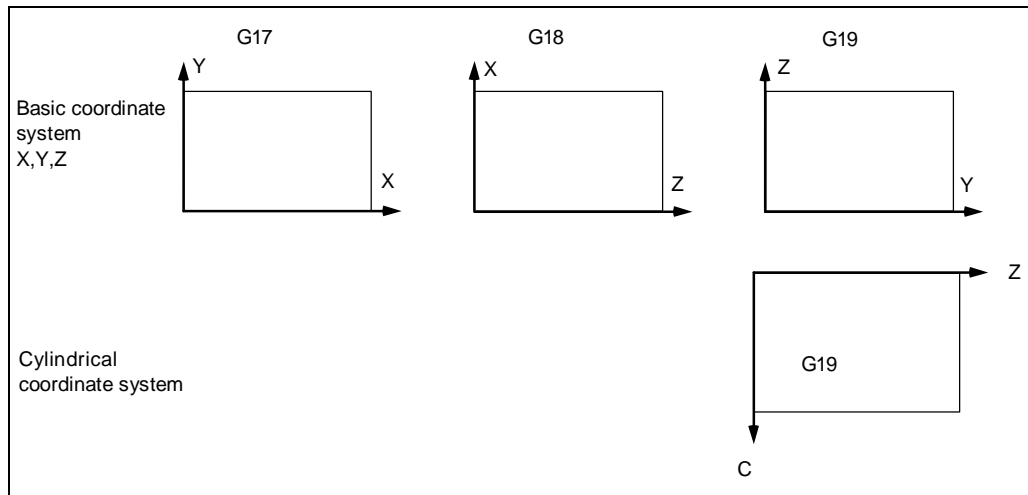
The correspondence of the rotation axis to an axis' parallel axis is set with the parameters (#1029, #1030, #1031).

The circular interpolation and tool radius compensation, etc., can be designated on that plane. The plane selection command is set immediately before or after the G07.1 command. If not set and a movement command is issued, a program error (P485) will occur.





**(Note)** Depending on the model or version, the Z-C plane (Y-Z cylinder plane) will be automatically selected with G07.1 and G19. The circular interpolation and tool radius compensation, etc., can be designated on that plane.





## Relation with other functions

**(1) Circular interpolation**

- (a) Circular interpolation between the rotation axis and linear axis is possible during the cylindrical interpolation mode.
- (b) An R specification command can be issued with circular interpolation. (I, J and K cannot be designated.)

**(2) Tool radius compensation**

The tool radius can be compensated during the cylindrical interpolation mode.

- (a) Command the plane selection in the same manner as circular interpolation. When using tool radius compensation, start up and cancel the compensation within the cylindrical interpolation mode.
- (b) A program error (P485) will occur if G07.1 is commanded during tool radius compensation.
- (c) If the G07.1 command is issued with no movement command given after the tool radius compensation is canceled, the position of the axis in the G07.1 command block is interpreted as the position applied after the tool radius compensation is canceled and the following operations are performed.

**(3) Cutting asynchronous feed**

- (a) The asynchronous mode is forcibly set when the cylindrical interpolation mode is started.
- (b) When the cylindrical interpolation mode is canceled, the synchronization mode will return to the state before the cylindrical interpolation mode was started.
- (c) A program error (P485) will occur if G07.1 is commanded in the constant surface speed control mode (G96).

**(4) Miscellaneous functions**

- (a) The miscellaneous function (M) and 2nd miscellaneous function can be issued even in the cylindrical interpolation mode.
- (b) The S command in the cylindrical interpolation mode issues the rotary tool's rotation speed instead of the spindle rotation speed.
- (c) Issue the T command before cylindrical interpolation is started. A program error (P485) will occur if the T command is issued in the cylindrical interpolation mode.

⋮

T1212 ;                    ... T command before cylindrical interpolation → Valid

T0 X100. Z0 ;

G19 Z C ;

G07.1 C100. ;

⋮

T1200 ;                    ... T command in cylindrical interpolation mode →Program error

⋮

G07.1 C0 ;

- (d) Complete the tool compensation movement (movement of tool length and wear compensation amount) before executing the cylindrical interpolation. If the tool compensation movement is not completed when the cylindrical interpolation start command has been issued, the followings will be resulted:
- Machine coordinate is not changed even if G07.1 is executed.
  - The workpiece coordinate is changed to that of the post tool length compensation when G07.1 is executed. (Even if canceling the cylindrical interpolation, this workpiece coordinate will not be canceled.)

#### (5) F command during cylindrical interpolation

As for the F command in the cylindrical interpolation mode, whether the previous F command is used or not depends on that the mode just before G07.1 is the feed per minute command (G94) or feed per rotation command (G95).

- (a) When G94 is commanded just before G07.1  
 If there is no F command in the cylindrical interpolation, the previous F command feedrate will be used.  
 The feedrate after the cylindrical interpolation mode is canceled will remain the F command feedrate issued when the cylindrical interpolation mode was started or the final F command feedrate set during cylindrical interpolation.
- (b) When G95 is commanded just before G07.1  
 The previous F command feedrate cannot be used during cylindrical interpolation, thus a new F command must be issued.  
 The feedrate after the cylindrical interpolation mode is canceled will return to that applied before the cylindrical interpolation mode was started.

#### When there is no F command in G07.1

Previous mode	No F command	After G07.1 is canceled
G94	Previous F is used	←
G95	Program error (P62)	F just before G07.1 is used

#### When F is commanded in G07.1

Previous mode	No F command	After G07.1 is canceled
G94	Commanded F is used	←
G95	Commanded F is used *1	F just before G07.1 is used

\*1) Moves with the feed per minute command during G07.1.



## Restrictions and precautions

- (1) The following G code commands can be used during the cylindrical interpolation mode.

G code	Details
G00	Positioning
G01	Linear interpolation
G02	Circular interpolation (CW)
G03	Circular interpolation (CWW)
G04	Dwell
G09	Exact stop check
G22/23	Chuck barrier ON/OFF
G40-42	Tool nose R compensation
G61	Exact stop check mode
G64	Cutting mode
G65	Macro call (simple call)
G66	Macro modal call A (modal call)
G66.1	Macro modal call B (block call per macro)
G67	Macro modal call cancel (modal call cancel)
G80-89	Hole drilling fixed cycle
G90/91	Absolute/incremental value command
G94	Asynchronous feed
G98	Hole drilling cycle initial return
G99	Hole drilling cycle R point return

A program error (P481) may occur if a G code other than those listed above is commanded during cylindrical interpolation.

- (2) The cylindrical interpolation mode is canceled when the power is turned ON or reset.
- (3) A program error (P484) will occur if any axis commanded for cylindrical interpolation has not completed zero point return.
- (4) Tool radius compensation must be canceled before the cylindrical interpolation mode can be canceled.
- (5) When the cylindrical interpolation mode is canceled, the mode will change to the cutting mode, and the plane will return to that selected before cylindrical interpolation.
- (6) The program of the block during the cylindrical interpolation cannot be restarted (program restart).
- (7) A program error (P486) will occur if the cylindrical interpolation command is issued during the mirror image.
- (8) When the cylindrical interpolation mode is started or canceled, the deceleration check is performed.
- (9) A program error (P481) will occur if the cylindrical interpolation or the pole coordinate interpolation is commanded during the cylindrical interpolation mode.
- (10) The G84 or G88 synchronous tapping cycles cannot be used during cylindrical interpolation mode. The asynchronous tap can be used during cylindrical interpolation mode; however, the synchronous tap must not be commanded.



### Example of program

<Program>

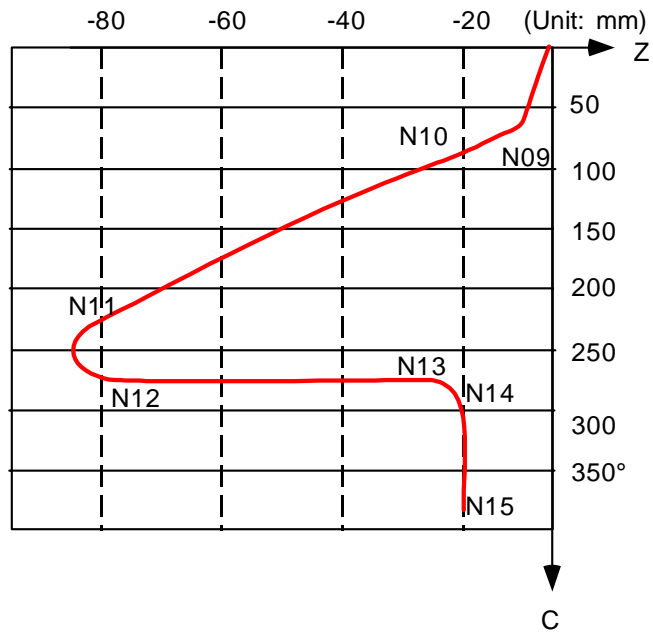
```

N01 G28XZC;
N02 T0202;
N03 G97S100M23;
N04 G00X50.Z0.;
N05 G94G01X40.F100.;
N06 G19C0Z0;
N07 G07.1C20.;
N08 G41;
N09 G01Z-10.C80.F150;
N10 Z-25.C90.;
N11 Z-80.C225;
N12 G03Z-75.C270.R55.;
N13 G01Z-25;
N14 G02Z-20.C280.R80.;
N15 G01C360.
N16 G40;
N17 G07.1C0;
N18 G01X50.;
N19 G0X100.Z100.;
N20 M25;
N21 M30;
    
```

<Parameter>	
#1029 aux_I	
#1030 aux_J	C
#1031 aux_K	

← Command of plane selection for cylindrical interpolation and command of two interpolation axes  
 ← Cylindrical interpolation start

← Cylindrical interpolation cancel



## 6. Interpolation Functions

### 6.9 Pole Coordinate Interpolation

#### 6.9 Pole Coordinate Interpolation; G12.1 (Only 6, 7 in G Code System)

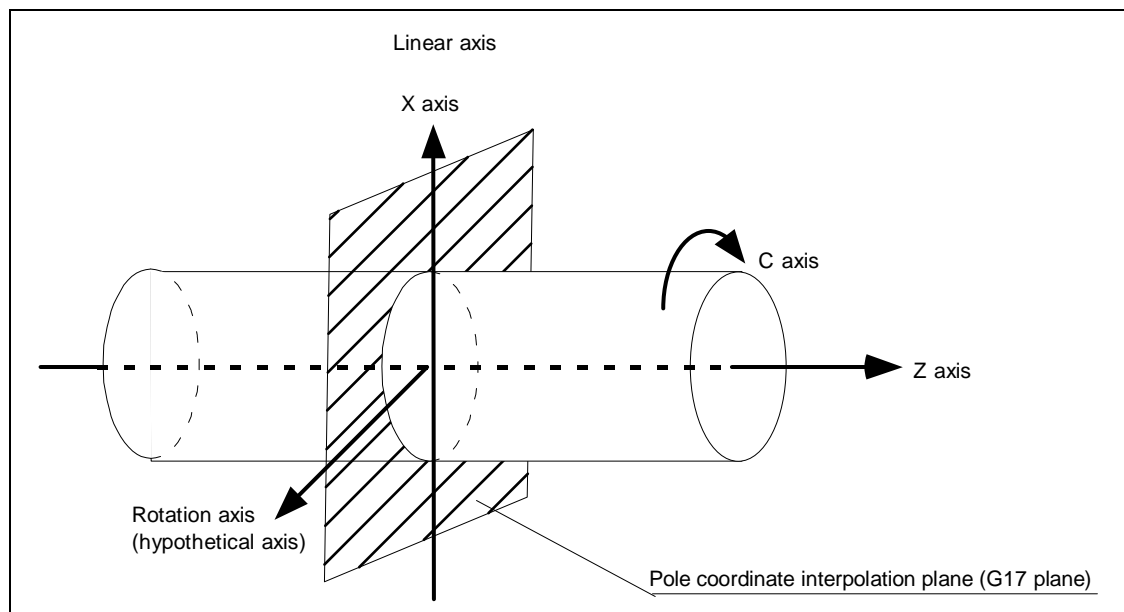
[E68]



#### Function and purpose

This function converts the commands programmed with the orthogonal coordinate axis into linear axis movement (tool movement) and rotation axis movement (workpiece rotation), and controls the contour.

The plane that uses the linear axis as the plane's 1st orthogonal axis, and the intersecting hypothetical axis as the plane's 2nd axis (hereafter "pole coordinate interpolation plane") is selected. Pole coordinate interpolation is carried out on this plane. The workpiece coordinate system zero point is used as the coordinate system zero point during pole coordinate interpolation.



This is effective for cutting a notch section on a linear section of the workpiece diameter, and for cutting cam shafts, etc.



#### Command format

**G12.1 ; Pole coordinate interpolation mode start**

**G13.1 ; Pole coordinate interpolation mode cancel**

- (1) The coordinates commanded in the interval from the start to cancellation of the pole coordinate interpolation mode will be the pole coordinate interpolation.
 

G12.1 ;	Pole coordinate interpolation mode start	
:		(Pole coordinate interpolation will start)
:		(The coordinate commands in this interval will be the pole coordinate interpolation)
G13.1 ;	Pole coordinate interpolation mode cancel	
		(Pole coordinate interpolation is canceled)
- (2) G112 and G113 can be used instead of G12.1 and G13.1.
- (3) Command G12.1 and G13.1 in an independent block. A program error (P33) will occur if this command is issued in the same block as another G code.
- (4) Linear interpolation or circular interpolation can be commanded during the pole coordinate interpolation mode.
- (5) The coordinates can be commanded with either an absolute command or incremental command.
- (6) Tool radius compensation can be applied on the program command. Pole coordinate interpolation will be executed on the path after tool radius compensation.
- (7) Command the segment feed in the pole coordinate interpolation plane (orthogonal coordinate system) with F. The F unit is mm/min or inch/min.
- (8) When the G12.1 and G13.1 commands are issued, the deceleration check is executed.



#### Detailed description

##### (1) Plane selection

The linear axis and rotation axis used for pole coordinate interpolation must be set beforehand with the parameters.

- (a) Determine the deemed plane for carrying out pole coordinate interpolation with the parameter (#1533) for the linear axis used for pole coordinate interpolation.

#1533 setting value	Deemed plane
X	G17 (XY plane)
Y	G19 (YZ plane)
Z	G18 (ZX plane)
Blank (no setting)	G17 (XY plane)

- (b) A program error (P485) will occur if the plane selection command (G16 to G19) is issued during the pole coordinate interpolation mode.

**(Note)** Depending on the model or version, parameter (#1533) may not be provided. In this case, the operation will be the same as if the parameter (#1533) is blank (no setting).

#### (2) Related parameters

#	Item	Details	Setting range
1516	mill_ax	Milling axis name Set the name of the rotation axis for pole coordinate interpolation, cylindrical interpolation. Only one of the rotation axes can be set.	A to Z
1517	mill_c	Milling interpolation hypothetical axis name Select the hypothetical axis command name for pole coordinate interpolation, cylindrical interpolation. 0: Y axis command 1: Command rotation axis name	0 / 1
8111	Milling Radius	Select the diameter and radius of the linear axis for polar coordinate interpolation and cylindrical interpolation. 0: Radius command for all axes 1: Each axis setting (follows #1019 dia diameter designation axis)	0 / 1



#### Relation with other functions

#### (1) Program commands during pole coordinate interpolation

- (a) The program commands in the pole coordinate interpolation mode are commanded with the orthogonal coordinate value of the linear axis and rotation axis (hypothetical axis) on the pole coordinate interpolation plane.  
The axis address of the rotation axis (C) is commanded as the axis address for the plane's 2nd axis (hypothetical axis) command.  
The command unit is not deg (degree), and instead is the same unit (mm or inch) as the command issued with the axis address for the plane's 1st axis (linear axis).
- (b) The hypothetical axis coordinate value will be set to "0" when G12.1 is commanded. In other words, the position where G12.1 is commanded will be interpreted as angle = 0, and the pole coordinate interpolation will start.

#### (2) Circular interpolation on pole coordinate plane

The arc radius address for carrying out circular interpolation during the pole coordinate interpolation mode is determined with the linear axis parameter (#1533).

#1533 setting value	Center designation command
X	I, J (pole coordinate plane is interpreted as XY plane)
Y	J, K (pole coordinate plane is interpreted as YZ plane)
Z	K, I (pole coordinate plane is interpreted as ZX plane)
Blank (no setting)	I, J (pole coordinate plane is interpreted as XY plane)

The arc radius can also be designated with the R command.

**(Note)** Depending on the model or version, parameter (#1533) may not be provided. In this case, the operation will be the same as if the parameter (#1533) is blank (no setting).

#### (3) Tool radius compensation

The tool radius can be compensated during the pole coordinate interpolation mode.

- (a) Command the plane selection in the same manner as pole coordinate interpolation.  
When using tool radius compensation, it must be started up and canceled within the pole coordinate interpolation mode.
- (b) A program error (P485) will occur if pole coordinate interpolation is executed during tool radius compensation.
- (c) If the G12.1 and G13.1 commands are issued with no movement command given after the tool radius compensation is canceled, the position of the axis in the G12.1 and G13.1 commands block is interpreted as the position applied after the tool radius compensation is canceled and the following operations are performed.



#### (4) Cutting asynchronous feed

- (a) The asynchronous mode is forcibly set when the pole coordinate interpolation mode is started.
- (b) When the pole coordinate interpolation mode is canceled, the synchronization mode will return to the state before the pole coordinate interpolation mode was started.
- (c) A program error (P485) will occur if G12.1 is commanded in the constant surface speed control mode (G96).

#### (5) Miscellaneous functions

- (a) The miscellaneous function (M) and 2nd miscellaneous function can be issued even in the pole coordinate interpolation mode.
- (b) The S command in the pole coordinate interpolation mode issues the rotary tool's rotation speed instead of the spindle rotation speed.
- (c) Issue the T command before pole coordinate interpolation is started. A program error (P485) will occur if the T command is issued in the pole coordinate interpolation mode.

```

:
:
T1212 ;      ... T command before pole coordinate interpolation → Valid
G0 X100. Z0 ;
G12.1 ;
:
T1200 ;      ... T command in pole coordinate interpolation mode → Program error
:
G13.1 ;

```

- (d) Complete the tool compensation operation (movement of tool length and wear compensation amount) before executing the polar coordinate interpolation. If the tool compensation operation is not completed when the polar coordinate interpolation start command has been issued, the followings will be resulted:
  - Machine coordinate is not changed even if G12.1 is executed.
  - The workpiece coordinate is changed to that of the post tool length compensation when G12.1 is executed. (Even if canceling the polar coordinate interpolation, this workpiece coordinate will not be canceled.)

#### (Example)

Workpiece coordinate offset (X axis)=20.

Tool compensation amount of T0101 (X axis)=100.

Setting compensation operation with movement command after T command

<Machining program>	<Workpiece coordinate>		<Machine coordinate>		
:	[X axis]	[C axis]	[X axis]	[C axis]	
G00 X200. C0.;	200.	0.	220.	0.	
T0101;	200.	0.	220.	0.	
G12.1;	100.	0.	220.	0.	<-The workpiece coordinate system is shifted (without moving the axis)
G01 X50. F1000;	50.	0.	170.	0.	
:					

#### (6) F command during pole coordinate interpolation

As for the F command in the pole coordinate interpolation mode, whether the previous F command is used or not depends on that the mode just before G12.1 is the feed per minute command (G94) or feed per rotation command (G95).

##### (a) When G94 is commanded just before G12.1

If there is no F command in the pole coordinate interpolation, the previous F command feedrate will be used.

The feedrate after the pole coordinate interpolation mode is canceled will remain the F command feedrate issued when the pole coordinate interpolation mode was started or the final F command feedrate set during pole coordinate interpolation.

The previous F command feedrate cannot be used during pole coordinate interpolation.

##### (b) When G95 is commanded just before G12.1

The previous F command feedrate cannot be used during pole coordinate interpolation. A new F command must be issued.

The feedrate after the pole coordinate interpolation mode is canceled will return to that applied before the pole coordinate interpolation mode was started.

#### When there is no F command in G12.1

Previous mode	No F command	After G13.1 is canceled
G94	Previous F is used	←
G95	Program error (P62)	F just before G12.1 is used

#### When F is commanded in G12.1

Previous mode	No F command	After G07.1 is canceled
G94	Commanded F is used	←
G95	Commanded F is used *1	F just before G12.1 is used

\*1) Moves with the feed per minute command during G12.1.

#### (7) Hole drilling axis in the hole drilling fixed cycle command during the pole coordinate interpolation

Hole drilling axis in the hole drilling fixed cycle command during the pole coordinate interpolation is determined with the linear axis parameter (# 1533).

#1533 setting value	Hole drilling axis
X	Z (pole coordinate plane is interpreted as XY plane)
Y	X (pole coordinate plane is interpreted as YZ plane)
Z	Y (pole coordinate plane is interpreted as ZX plane)
Blank (no setting)	Z (pole coordinate plane is interpreted as XY plane)



#### Restrictions and precautions

- (1) The following G code commands can be used during the pole coordinate interpolation mode.

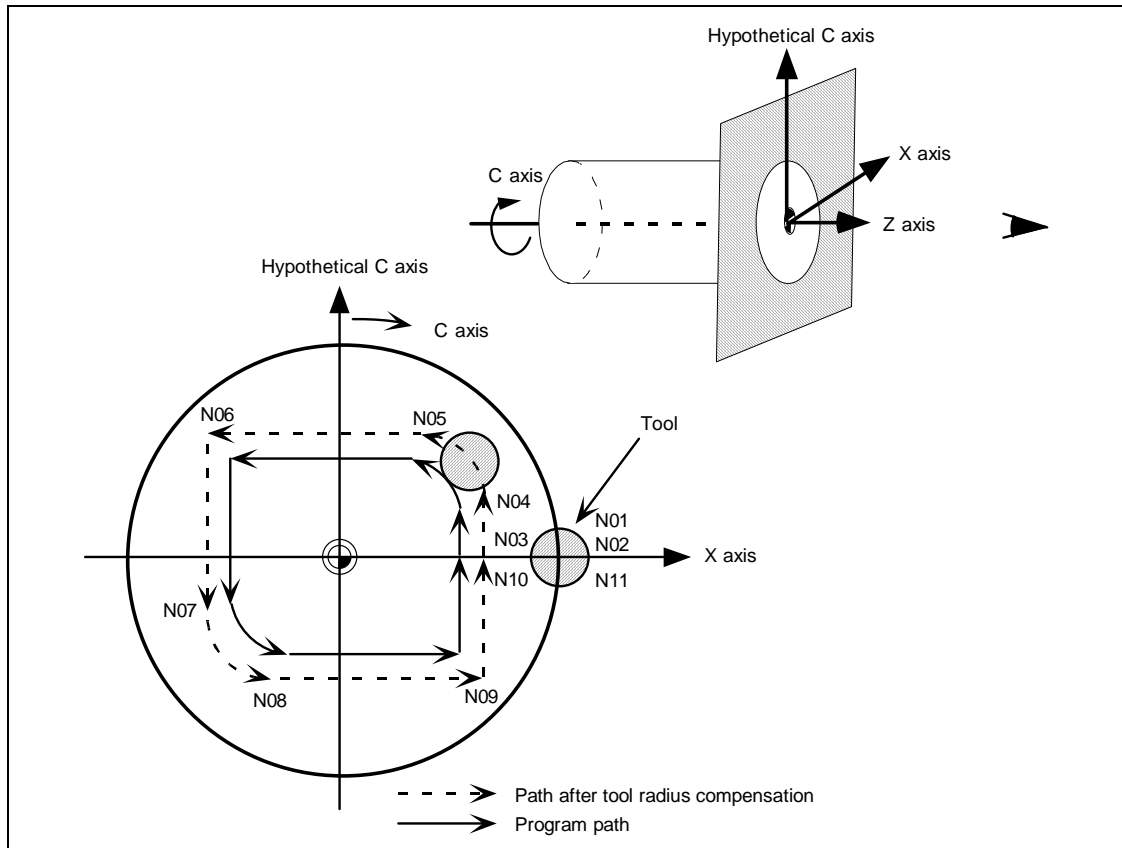
G code	Details
G00	Positioning
G01	Linear interpolation
G02	Circular interpolation (CW)
G03	Circular interpolation (CWW)
G04	Dwell
G09	Exact stop check
G22/23	Chuck barrier ON/OFF
G40-42	Tool nose R compensation
G61	Exact stop check mode
G64	Cutting mode
G65	Macro call (simple call)
G66	Macro modal call A (modal call)
G66.1	Macro modal call B (block call per macro)
G67	Macro modal call cancel (modal call cancel)
G80-89	Hole drilling fixed cycle
G90/91	Absolute/incremental value command
G94	Asynchronous feed
G98	Hole drilling cycle initial return
G99	Hole drilling cycle R point return

A program error (P481) may occur if a G code other than those listed above is commanded during pole coordinate interpolation.

- (2) The program cannot be restarted (resumed) for a block in pole coordinate interpolation.
- (3) Before commanding pole coordinate interpolation, set the workpiece coordinate system so that the center of the rotation axis is at the coordinate system zero point. Do not change the coordinate system during the pole coordinate interpolation mode. (G50, G52, G53, relative coordinate reset, G54 to G59, etc.)
- (4) The feedrate during pole coordinate interpolation will be the interpolation speed on the pole coordinate interpolation plane (orthogonal coordinate system).  
(The relative speed with the tool will be converted with pole coordinate conversion.)  
When passing near the center of the rotation axis on the pole coordinate interpolation plane (orthogonal coordinate system), the rotation axis side feedrate after pole coordinate interpolation will be very high.
- (5) The axis movement command outside of the plane during pole coordinate interpolation will move unrelated to the pole coordinate interpolation.
- (6) The current position displays during pole coordinate interpolation will all indicate the actual coordinate value. However, the "remaining movement amount" will be the movement amount on the pole coordinate input plane.
- (7) The pole coordinate interpolation mode will be canceled when the power is turned ON or reset.
- (8) A program error (P484) will occur if any axis commanded for pole coordinate interpolation has not completed zero point return.
- (9) Tool radius compensation must be canceled before the pole coordinate interpolation mode can be canceled.
- (10) When the pole coordinate interpolation mode is canceled, the mode will change to the cutting mode, and the plane will return to that selected before pole coordinate interpolation.
- (11) A program error (P486) will occur if the pole coordinate interpolation command is issued during the mirror image.
- (12) A program error (P481) will occur if the cylindrical interpolation or the pole coordinate interpolation is commanded during the pole coordinate interpolation mode.
- (13) The G84 or G88 synchronous tapping cycles cannot be used during pole coordinate interpolation mode. The asynchronous tap can be used during pole coordinate interpolation mode; however, the synchronous tap must not be commanded.



### Example of program



### <Program>

```

:
:
N00 T0101;
:
:
N01 G17 G90 G0 X40.0 C0 Z0;   Setting of start position
N02 G12.1;                   Pole coordinate interpolation mode: Start
N03 G1 G42 X20.0 F2000;      Actual machining start
N04 C10.0;
N05 G3 X10.0 C20.0 R10.0;
N06 G1 X-20.0;               Shape program
N07 C-10.0;
N08 G3 X-10.0 C-20.0 I10.0 J0; (Follows orthogonal coordinate values on X-C
N09 G1 X20.0;                hypothetical axis plane.)
N10 C0;
N11 G40 X40.0;
N12 G13.1;                   Pole coordinate interpolation mode: Cancel
:
:
M30 ;
    
```

## 7. Feed Functions

## 7.1 Rapid traverse rate



## Function and purpose

The rapid traverse rate can be set independently for each axis. The speed range that can be set is 1mm/min to 240,000 mm/min (E60) or 1mm/min to 1,000,000mm/min (E68) when the input setting unit is 1 $\mu$ m. Note that the upper limit speed is limited by the machine specification. Refer to the specifications manual of the machine for the rapid traverse rate settings.

Two paths are available during positioning: the interpolation type where the area from the start point to the end point is linearly interpolated or the non-interpolation type where movement proceeds at the maximum speed of each axis. The type is selected by parameter "#1086 G0Intp". The positioning time is the same in each case.

**(Note)** Rapid traverse override

Override can be applied by an external input signal for both manual and automatic rapid traverse. There are 2 types which are determined by the PLC specifications.

Type 1: Override in 4 steps: 1%, 25%, 50% and 100%

Type 2: Override in 1% steps from 0% to 100%.

## 7.2 Cutting feedrate



## Function and purpose

The cutting feedrate is assigned with address F and 8 digits.

The F8 digits are assigned with a decimal point for a 5-digit integer and a 3-digit fraction. The cutting feedrate is valid for the G01, G02, G03 and G33 commands.

**Examples (asynchronous feed)**

				Feedrate	
G1	X100.	Z100.	F200 ;	200.0mm/min	F200. or F200.000 gives the same rate.
F1	X100.	Z100.	F123.4 ;	123.4mm/min	
F1	X100.	Z100.	F56.789 ;	56.789mm/min	

Speed range that can be commanded

(1) When using E60

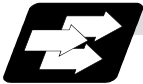
Command mode	F command range	Feed rate command range	Remarks
mm/min	0.001 ~ 240000.000	0.001 ~ 240000 mm/min	
inch/min	0.0001 ~ 9448.8188	0.0001 ~ 9448.8188 inch/min	
°/min	0.001 ~240000.000	0.001 ~ 240000 °/min	

(2) When using E68 (when input setting unit is 1 $\mu$ m)

Command mode	F command range	Feed rate command range	Remarks
mm/min	0.001 ~ 1000000.000	0.001 ~ 1000000.000 mm/min	
inch/min	0.0001 ~ 39370.0787	0.0001 ~ 39370.0787 inch/min	
°/min	0.001 ~ 1000000.000	0.001 ~ 1000000.000 °/min	

**(Note 1)** A program error (P62) results when there is no F command in the first cutting command (G01, G02, G03, G33) after the power has been turned ON.

#### 7.3 F1-digit feed



##### Function and purpose

By setting the F1-digit feed parameter, the feedrate which has been set to correspond to the 1-digit number following the F address serves as the command value.

When F0 is assigned, the rapid traverse rate is established and the speed is the same as for G00. (G modal does not change.)

When F1 to F5 is assigned, the feedrate set to correspond to the command serves as the command value.

The command greater than F6 digits is considered to be the normal cutting feedrate.

The F1-digit command is valid only in a G01, G02 or G03 modal.

The F1-digit command can also be used for fixed cycle.



##### Detailed description

- (1) Use of both the F1-digit command and normal cutting feedrate command is possible when the F1-digit is valid.

**(Example 1)**

F0	Rapid traverse rate
F1 to F5	F1-digit
F6 or more	Normal cutting feedrate command

- (2) F1 to F5 are invalid in the G00 mode and the rapid traverse rate is established instead.
- (3) If F0 is used in the G02 or G03 mode, a program error (P121) will result.
- (4) When F1. to F5. (with decimal point) are assigned, the 1mm/min to 5mm/min direct commands are established instead of the F1-digit command.
- (5) When the commands are used with the millimeter or degree units, the feedrate set to correspond to F1 to F5 serves as the assigned speed mm (°)/min.
- (6) When the commands are used with inch units, one-tenth of the feedrate set correspond to F1 to F5 serves at the assigned speed inch/min.
- (7) During a F1-digit command, the F1-digit number and F1-digit command signal are output as the PLC signals.
- (8) F1-digit and G commands
  - (a) 01 group G command in same block as F1-digit commands

	Executed feedrate	Modal display rate	G modal
G0F0 F0G0	Rapid traverse rate	0	G0
G0F1 F1G0	Rapid traverse rate	1	G0
G1F0 F0G1	Rapid traverse rate	0	G1
G1F1 F1G1	F1 contents	1	G1

- (b) F1-digit and unmodal commands may be assigned in the same block. In this case, the unmodal command is executed and at the same time the F1-digit modal command is updated.

## 7. Feed Functions

### 7.4 Synchronous/asynchronous feed

#### 7.4 Synchronous/asynchronous feed; G94, G95



#### Function and purpose

Using the G95 command, it is possible to assign the feed amount per rotation with an F code. When this command is used, the rotary encoder must be attached to the spindle.



#### Command format

<b>G94;</b>	
<b>G95;</b>	
G94	Per-minute feed (mm/min) (asynchronous feed)
G95	Per-rotation feed (mm/rev) (synchronous feed)

The G95 command is a modal command and so it is valid until the G94 command (per-minute feed) is next assigned.

- (1) The F code command range is as follows.  
The movement amount per spindle rotation with synchronous feed (per-rotation feed) is assigned by the F code and the command range is as shown in the table below.

- (a) When using E60

Input command unit system	Metric input		Inch input	
	B (0.001mm)		B (0.0001inch)	
Command mode	Per-minute feed	Per-rotation feed	Per-minute feed	Per-rotation feed
Command address	F (mm/min)	F (mm/rev)	F (inch/min)	F (inch/rev)
Minimum command unit	1 (= 1.000), (1. = 1.000)	1 (= 0.0001), (1. = 1.0000)	1 (= 0.01), (1. = 1.0000)	1 (= 0.000001), (1. = 1.000000)
Command range	0.001 ~1000000.000	0.0001 ~999.9999	0.0001 ~ 39370.0787	0.000001 ~ 99.999999

- (b) When using E68

Input command unit system	Metric input		C (0.0001mm)	
	B (0.001mm)		C (0.0001mm)	
Command mode	Per-minute feed	Per-rotation feed	Per-minute feed	Per-rotation feed
Command address	F (mm/min)	F (mm/rev)	F (mm/min)	F (mm/rev)
Minimum command unit	1 (= 1.000), (1. = 1.000)	1 (= 0.0001), (1. = 1.0000)	1 (= 1.0000), (1. = 1.0000)	1 (= 0.00001), (1. = 1.00000)
Command range	0.001 ~1000000.000	0.0001 ~999.9999	0.0001 ~100000.0000	0.00001 ~99.99999

Input command unit system	B (0.0001inch)		C (0.00001inch)	
	Command mode	Per-minute feed	Per-rotation feed	Feed per minute
Command address	F (inch/min)	E (inch/rev)	F (inch/min)	E (inch/rev)
Minimum command unit	1 (= 0.01), (1. = 1.0000)	1 (= 0.000001), (1. = 1.000000)	1 (= 0.01), (1. = 1.00000)	1 (= 0.0000001), (1. = 1.0000000)
Command range	0.0001 ~ 39370.0787	0.000001 ~ 99.999999	0.00001 ~ 3937.00787	0.0000001 ~ 9.9999999

## 7. Feed Functions

### 7.4 Synchronous/asynchronous feed

- (2) The effective speed (actual movement speed of machine) under per-rotation feed conditions is given in the following formula (Formula 1).

$$FC = F \times N \times OVR \dots\dots\dots \text{(Formula 1)}$$

Where FC	= Effective rate (mm/min, inch/min)
F	= Commanded speed (mm/rev, inch/rev)
N	= Spindle rotation speed (r/min)
OVR	= Cutting feed override

When a multiple number of axes have been commanded at the same time, the effective rate FC in formula 1 applies in the vector direction of the command.

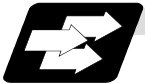
- (Note 1)** The effective rate (mm/min or inch/min), which is produced by converting the commanded speed, the spindle rotation speed and the cutting feed override into the per-minute speed, appears as the FC on the monitor 1. screen of the setting and display unit.
- (Note 2)** When the above effective rate exceeds the cutting feed clamp rate, it is clamped at that clamp rate.
- (Note 3)** If the spindle rotation speed is zero when synchronous feed is executed, operation alarm "105" results.
- (Note 4)** During machine lock high-speed processing, the rate will be 240,000 mm/min (or 9448 inch/min, 240,000 °/min) when using E60 or 1,000,000 mm/min (or 39,370 inch/min, 1,000,000 °/min) when using E68 regardless of the commanded speed and spindle speed.
- (Note 5)** Under dry run conditions, asynchronous speed applies and movement results at the externally set speed (mm/min or inch/min).
- (Note 6)** Whether asynchronous feed (G94) or synchronous feed (G95) is to be established when the power is turned ON or when M02 or M30 is executed can be selected by setting parameter "#1074 I\_Sync".



## 7. Feed Functions

### 7.5 Feedrate designation and effects on control axes

#### 7.5 Feedrate designation and effects on control axes



##### Function and purpose

It has already been mentioned that a machine has a number of control axes. These control axes can be divided into linear axes which control linear movement and rotation axes which control rotary movement. The feedrate is designed to assign the displacement speed of these axes, and the effect exerted on the tool movement speed which poses problems during cutting differs according to when control is exercised over the linear axes or when it is exercised over the rotation axes.

The displacement amount for each axis is assigned separately for each axis by a value corresponding to the respective axis. The feedrate is not assigned for each axis but assigned as a single value. Therefore, when two or more axes are to be controlled simultaneously, it is necessary to understand how this will work for each of the axes involved.

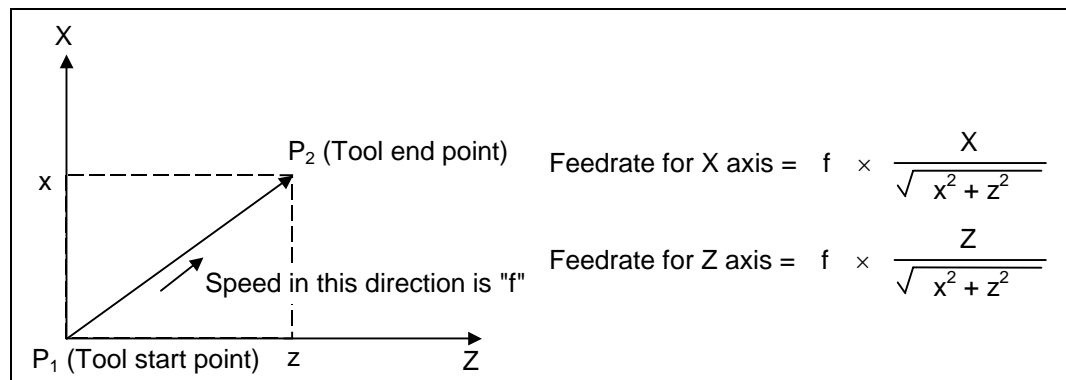
The assignment of the feedrate is described with the following related items.



##### When controlling linear axes

Even when only one machine axis is to be controlled or there are two or more axes to be controlled simultaneously, the feedrate which is assigned by the F code functions as a linear speed in the tool advance direction.

**(Example)** When the feedrate is designated as "f" and linear axes (X and Z) are to be controlled

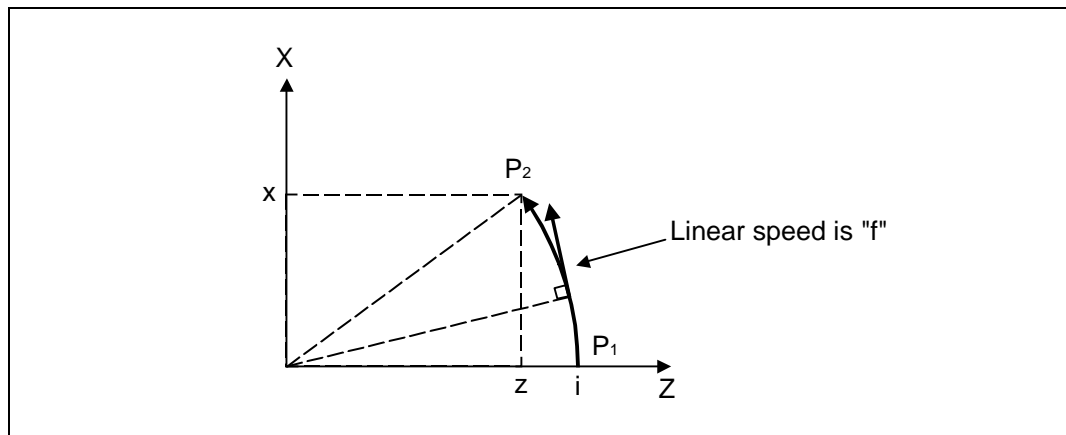


When only linear axes are to be controlled, it is sufficient to designate the cutting feed in the program. The feedrate for each axis is such that the designated rate is broken down into the components corresponding to the movement amounts.

## 7. Feed Functions

### 7.5 Feedrate designation and effects on control axes

**(Note)** When the circular interpolation function is used and the tool is moved along the circumference of an arc by the linear control axis, the rate in the tool advance direction, or in other words the tangential direction, will be the feedrate designated in the program.



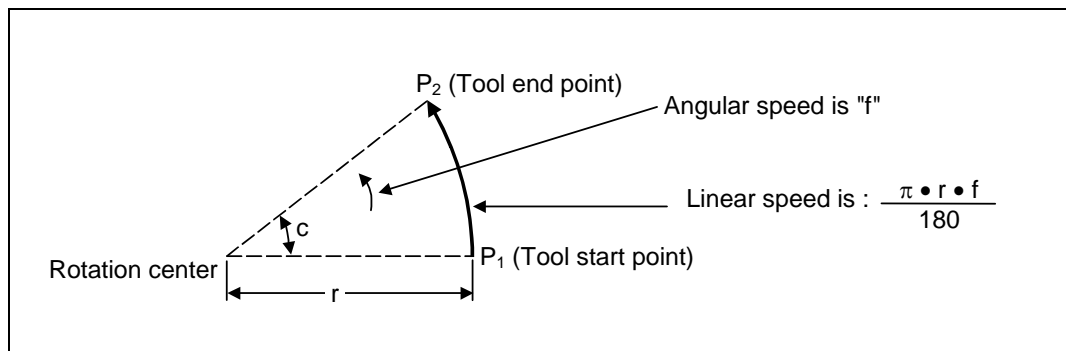
**(Example)** When the feedrate is designated as "f" and the linear axes (X and Z) are to be controlled using the circular interpolation function. In this case, the feedrate of the X and Z axes will change along with the tool movement. However, the combined speed will always be maintained at the constant value "f".



#### When controlling rotation axes

When rotation axes are to be controlled, the designated feedrate functions as the rotation speed of the rotation axes or, in other words, as an angular speed. Consequently, the cutting feed in the tool advance direction, or in other words the linear speed, varies according to the distance between the center of rotation and the tool. This distance must be borne in mind when designating the feedrate in the program.

**(Example)** When the feedrate is designated as "f" and rotation axis (C) is to be controlled ("f" units = °/min)



In this case, in order to make the cutting feed (linear speed) in the tool advance direction "fc":

$$fc = f \times$$

Therefore, the rate to be designated in the program must be:

$$f = fc \times$$

$$\frac{180}{\pi \cdot r}$$

## 7. Feed Functions

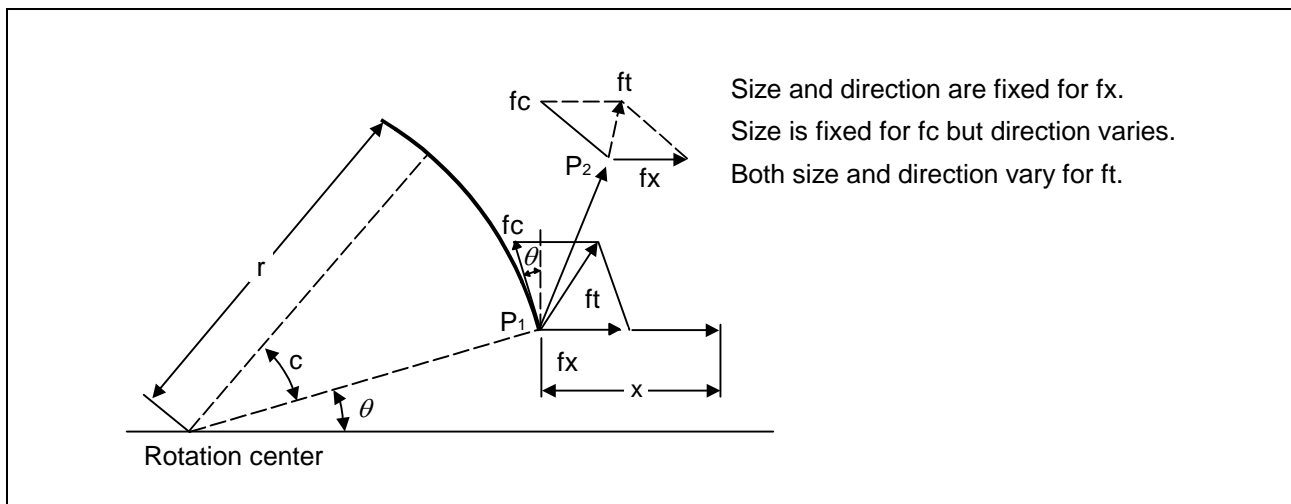
### 7.5 Feedrate designation and effects on control axes



When linear and rotation axes are to be controlled at the same time

The NC unit proceeds in exactly the same way whether linear or rotation axes are to be controlled. When a rotation axis is to be controlled, the numerical value assigned by the coordinate words (C, H) is the angle and the numerical values assigned by the feedrate (F) are all handled as linear speeds. In other words,  $1^\circ$  of the rotation axis is treated as being equivalent to 1mm of the linear axis. Consequently, when both linear and rotation axes are to be controlled simultaneously, the components for each axis of the numerical values assigned by F will be the same as for section (1) above (applying when linear axes are to be controlled). However, although in this case both the size and direction of the speed components based on linear axis control do not vary, the direction of the speed components based on rotation axis control will change along with the tool movement (their size will not change). This means, as a result, that the combined tool advance direction feedrate will vary along with the tool movement.

**(Example)** When the feedrate is designated as "f" and linear (X) and rotation (C) axes are to be controlled simultaneously  
If the X-axis incremental command value is "x" and the C-axis incremental command value is "c":



## 7. Feed Functions

### 7.5 Feedrate designation and effects on control axes

X-axis feedrate (linear speed) "fx" and C-axis feedrate (angular speed) "ω" are expressed as:

$$f_x = f \times \frac{X}{\sqrt{X^2 + C^2}} \quad \dots\dots (1) \qquad \omega = f \times \frac{C}{\sqrt{X^2 + C^2}} \quad \dots\dots (2)$$

Linear speed "fc" based on C-axis control is expressed as:

$$f_c = \omega \times \frac{\pi \times r}{180} \quad \dots\dots (3)$$

If the speed in the tool advance direction at start point P<sub>1</sub> is "ft" and the component speeds in the X-axis and Y-axis directions are "ftx" and "fty," respectively. Then these can be expressed as:

$$f_{tx} = -r \sin \left( \frac{\pi}{180} \theta \right) \times \frac{\pi}{180} \omega + f_x \quad \dots\dots (4)$$

$$f_{ty} = -r \cos \left( \frac{\pi}{180} \theta \right) \times \frac{\pi}{180} \omega \quad \dots\dots (5)$$

Where r is the distance between center of rotation and tool (in mm units), and θ is the angle between the P<sub>1</sub> point and the X axis at the center of rotation (in units °). The combined speed "ft" according to formulae (1), (2), (3), (4) and (5) is:

$$f_t = \sqrt{f_{tx}^2 + f_{ty}^2} \\ = f \times \frac{\sqrt{X^2 - X \times C \times r \sin \left( \frac{\pi}{180} \theta \right) \frac{\pi}{90} + \left( \frac{\pi \times r \times C}{180} \right)^2}}{X^2 + C^2} \quad \dots\dots (6)$$

Consequently, feedrate "f" designated by the program must be as follows:

$$f = f_t \times \frac{X^2 + C^2}{\sqrt{X^2 - X \times C \times r \sin \left( \frac{\pi}{180} \theta \right) \frac{\pi}{90} + \left( \frac{\pi \times r \times C}{180} \right)^2}} \quad \dots\dots (7)$$

"ft" in formula (6) is the speed at the P<sub>1</sub> point and the value of θ changes as the C axis rotates, which means that the value of "ft" will also change. Consequently, in order to keep the cutting feed "ft" as constant as possible, the angle of rotation which is designated in one block must be reduced to as low as possible and the extent of the change in the θ value must be minimized.

### 7.6 Thread cutting mode



#### Function and purpose

F7-digit or E8-digit commands for thread leads can be issued for the thread cutting mode (G33, G34, G76, G78 commands).

The thread lead command range is 0.00001 to 999.9999mm/rev (F7 digits) or 0.0001 to 999.99999 mm/rev (E8 digits) (with input unit of microns).

(1) When using E60

#### Thread cutting metric input

#### Thread cutting inch input

Input unit system	B (0.001mm)			B (0.0001inch)		
Command address	F (mm/rev)	E (mm/rev)	E (ridges/inch)	F (inch/rev)	E (inch/rev)	E (ridges/inch)
Minimum command unit	1 (= 0.0001), (1.=1.0000)	1 (= 0.00001), (1.=1.00000)	1 (= 1.00), (1.=1.00)	1(=0.000001), (1.=1.000000)	1(=0.0000001), (1.=1.0000000)	1 (= 1.0000), (1.=1.0000)
Command range	0.0001 ~ 999.9999	0.00001 ~ 999.99999	0.03 ~ 999.99	0.0000001 ~ 99.999999	0.000010 ~ 9.9999999	0.0101 ~ 9999.9999

(2) When using E68

#### Thread cutting metric input

Input unit system	B (0.001mm)			C (0.0001mm)		
Command address	F (mm/rev)	E (mm/rev)	E (ridges/inch)	F (mm/rev)	E (mm/rev)	E (ridges/inch)
Minimum command unit	1 (= 0.0001), (1.=1.0000)	1 (= 0.00001), (1.=1.00000)	1 (= 1), (1.=1.00)	1 (= 0.00001), (1.=1.00000)	1(=0.000001), (1.=1.000000)	1 (= 1), (1.=1.000)
Command range	0.0001 ~ 999.9999	0.00001 ~ 999.99999	0.03 ~ 999.99	0.00001 ~ 99.99999	0.000001 ~ 99.999999	0.255 ~9999.999

#### Thread cutting inch input

Input unit system	B (0.0001inch)			C (0.00001inch)		
Command address	F (inch/rev)	E (inch/rev)	E (ridges/inch)	F (inch/rev)	E (inch/rev)	E (ridges/inch)
Minimum command unit	1(=0.000001), (1.=1.000000)	1(=0.0000001), (1.=1.0000000)	1 (= 1.0000), (1.=1.0000)	1(=0.0000001), (1.=1.0000000)	1(=0.00000001), (1.=1.00000000)	1(=1.00000), (1.=1.00000)
Command range	0.000001 ~ 99.999999	0.000010 ~ 9.9999999	0.0101 ~ 9999.9999	0.0000001 ~ 9.9999999	0.0000010 ~ 0.99999999	0.10001 ~999.99999

## 7. Feed Functions

### 7.7 Automatic acceleration/deceleration

#### 7.7 Automatic acceleration/deceleration

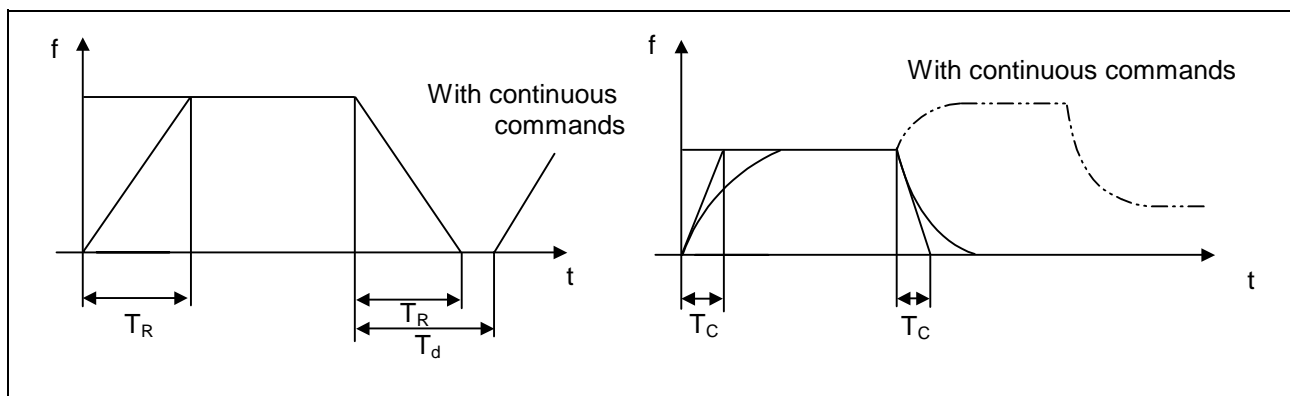


##### Function and purpose

The rapid traverse and manual feed acceleration/deceleration pattern is linear acceleration and linear deceleration.

Time constant  $T_R$  can be set independently for each axis using parameters in 1ms steps from 1 to 500ms.

The cutting feed (not manual feed) acceleration/deceleration pattern is exponential acceleration/deceleration. Time constant  $T_c$  can be set independently for each axis using parameters in 1ms steps across a range from 1 to 500ms. (Normally, the same time constant is set for all axes.)



Rapid traverse acceleration/deceleration Pattern  
( $T_R$  = Rapid traverse time constant)  
( $T_d$  = Deceleration check time)

Cutting feed acceleration/deceleration pattern  
( $T_c$  = Cutting feed time constant)

With rapid traverse and manual feed, the following block is executed after the command pulse of the present block has become "0" and the tracking error of the acceleration/deceleration circuit has become "0". However, with cutting feed, the following block is executed as soon as the command pulse of the present block becomes "0" although an external signal (error detect) can detect that the tracking error of the acceleration/deceleration circuit has reached "0" and the following block can be executed. When the in-position check has been made valid (selected by parameter "#1193 inpos") during the deceleration check, it is first confirmed that the tracking error of the acceleration/deceleration circuit has reached "0", then it is checked that the position deviation is less than the parameter setting value "#2204 SV024", and finally the following block is executed. It depends on the machine as to whether the error detect function can be activated by a switch or M function and so reference should be made to the instructions issued by the machine tool builder.

#### 7.8 Speed clamp



##### Function and purpose

This function exercises control over the actual cutting feedrate in which override has been applied to the cutting feedrate command so that the speed clamp value which has been preset independently for each axis is not exceeded.

**(Note)** Speed clamping is not applied to synchronous feed and thread cutting.

### 7.9 Exact stop check; G09



#### Function and purpose

In order for roundness to be prevented during corner cutting and for machine shock to be alleviated when the tool feedrate changes suddenly, there are times when it is desirable to start the commands in the following block once the in-position state after the machine has decelerated and stopped or the elapsing of the deceleration check time has been checked. The exact stop check function is designed to accomplish this purpose.

Whether to control with the deceleration check time or with the in-position state can be selected with the parameter(#1193 Inpos). (Refer to the section "7.11 Deceleration check")

The in-position width is set into parameter "#2224 SV024" or "#2077 G0inps", "#2078 G1inps" on the servo parameter screen.



#### Command format

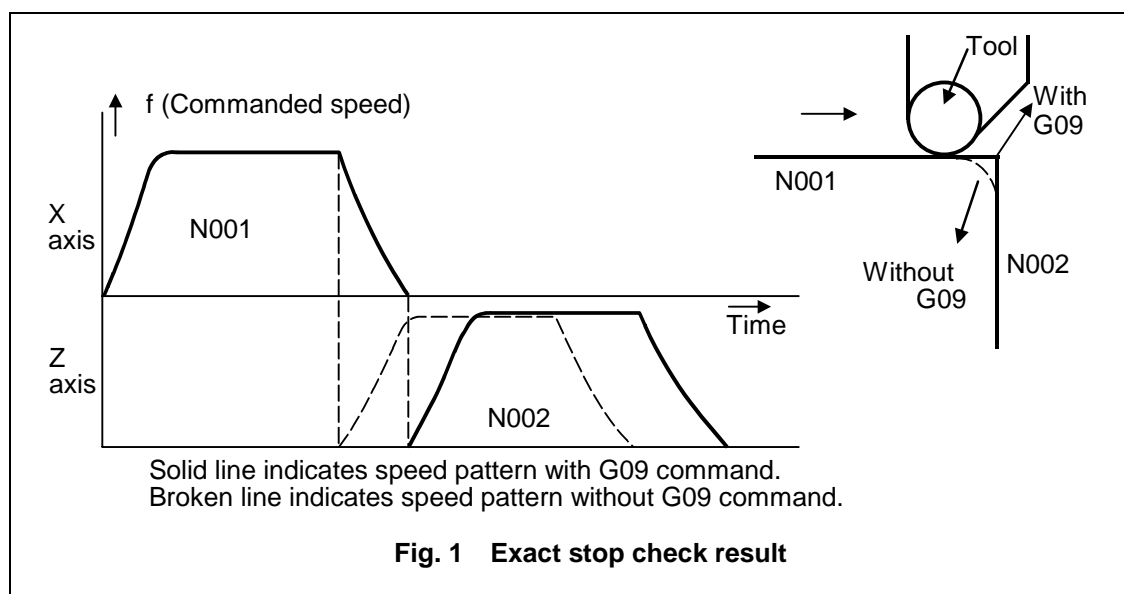
**G09 G01 (G02, G03) ;**

The exact stop check command G09 has an effect only with the cutting command (G01 ~ G03) in its particular block.



#### Example of program

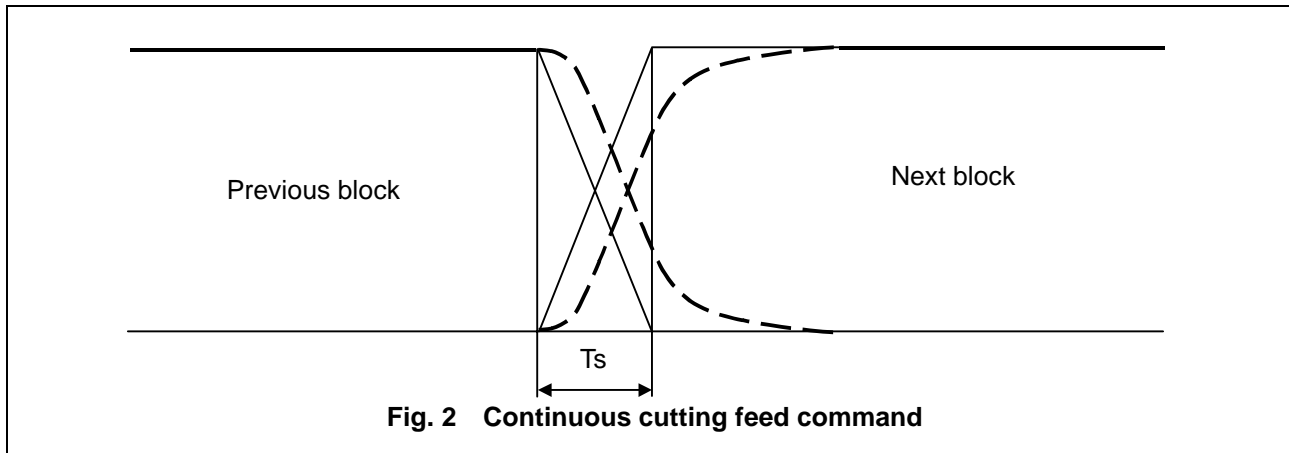
N001 G09 G01 X100.000 F150 ;	The commands in the following block are started once the deceleration check time or in-position state has been checked after the machine has decelerated and stopped.
N002 Z100.000 ;	



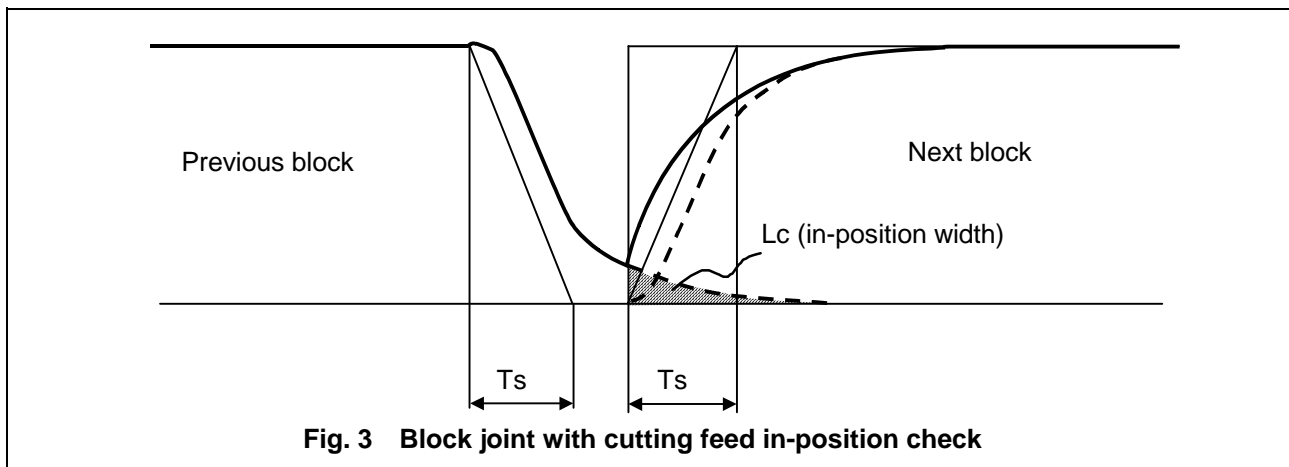


### Detailed description

#### (1) With continuous cutting feed



#### (2) With cutting feed in-position check

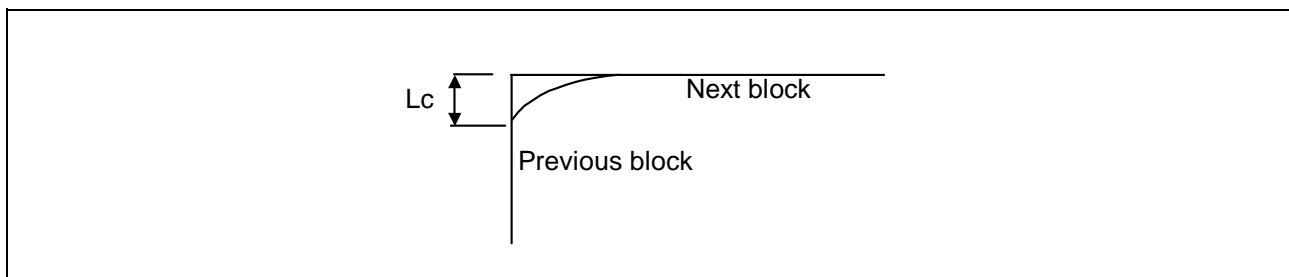


In Figs. 2 and 3:

$T_s$  = Cutting feed acceleration/deceleration time constant  
 $L_c$  = In-position width

As shown in Fig. 3, the in-position width " $L_c$ " can be set into the servo parameter "#2224 SV024" as the remaining distance (shaded area in Fig. 3) of the previous block when the next block is started.

The in-position width is designed to reduce the roundness at the workpiece corners to below the constant value.

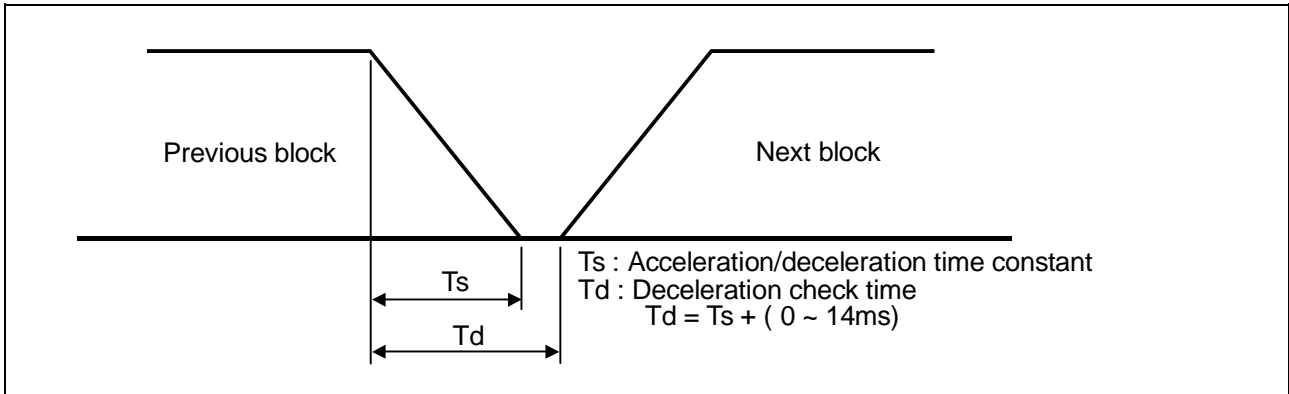


To eliminate corner roundness, set the value as small as possible to servo parameter "#2224 SV024" and perform an in-position check or assign the dwell command (G04) between blocks.

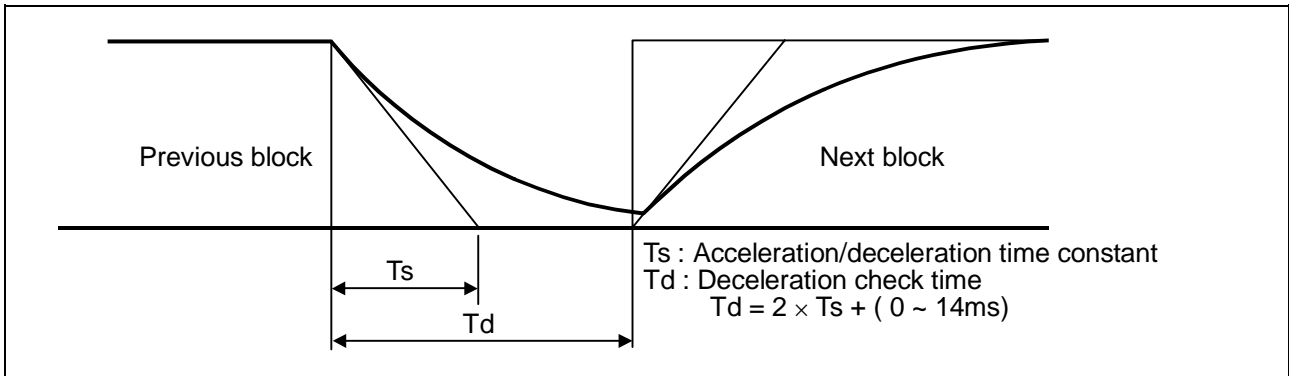


### (3) With deceleration check

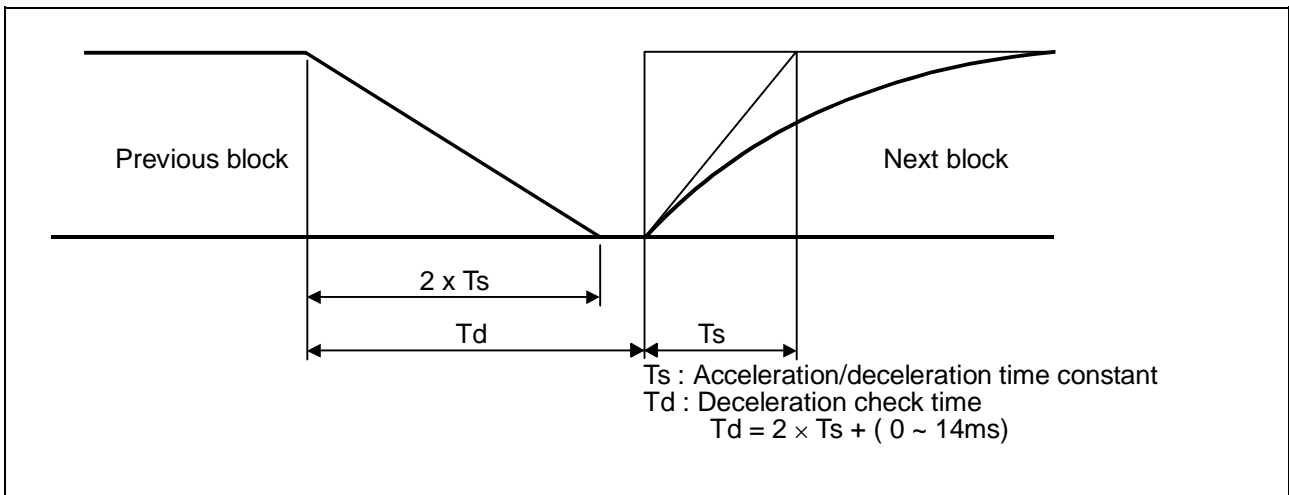
#### (a) With linear acceleration/deceleration



#### (b) With exponential acceleration/deceleration

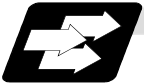


#### (c) With exponential acceleration/linear deceleration



The time required for the deceleration check during cutting feed is the longest among the cutting feed deceleration check times of each axis determined by the cutting feed acceleration/deceleration time constants and by the cutting feed acceleration/deceleration mode of the axes commanded simultaneously.

#### 7.10 Exact stop check mode ; G61



##### Function and purpose

Whereas the G09 exact stop check command checks the in-position status only for the block in which the command has been assigned, the G61 command functions as a modal. This means that deceleration will apply at the end points of each block to all the cutting commands (G01 to G03) subsequent to G61 and that the in-position status will be checked. G61 is released by automatic corner override (G62), tapping mode (G63), or cutting mode (G64).



##### Command format

G61 ;

In-position check is executed in the G61 block, and thereafter, the in-position check is executed at the end of the cutting command block is executed until the check mode is canceled.

### 7.11 Deceleration check



#### Function and purpose

The deceleration check is a function that determines the method of the check at the completion of the axis movement block's movement.

The deceleration check includes the in-position check and commanded speed check method.

The G0 and G1 deceleration check method combination can be selected.

(Refer to section "Deceleration check combination".)

With this function, the deceleration check in the reverse direction of G1 → G0 or G1 → G1 can be changed by changing the parameter setting.

#### (1) Types of deceleration check

<p><b>Commanded speed check</b>          With the commanded speed check, the completion of deceleration is judged when the command to the motor is completed.</p>	
<p><b>In-position check</b>          With the in-position check, the completion of deceleration is judged when the motor moves to the in-position width designated with the parameter.</p>	

#### (2) Designating deceleration check

The deceleration check by designating a parameter includes "deceleration check specification type 1" and "deceleration check specification type 2". The setting is selected with the parameter "#1306 InpsTyp".

##### (a) Deceleration check specification type 1 ("#1306 InpsTyp" = 0)

The G0 and G1 deceleration check method can be selected with the base specification parameter deceleration check method 1 (#1193 inpos) and "deceleration check method 2" (#1223 aux07/bit1).

Parameter	Rapid traverse command	Parameter	Other than rapid traverse command (G1 : other than G0 command)	
<b>Inpos (#1193)</b>	G0→XX (G0+G9→XX)	<b>AUX07/BIT-1 (#1223/BIT-1)</b>	G1+G9→XX	G0→XX
<b>0</b>	Command deceleration check	<b>0</b>	Command deceleration check	No deceleration check
<b>1</b>	In-position check	<b>1</b>	In-position check	

**(Note)** XX expresses all commands.

##### (b) Deceleration check specification type 2 ("#1306 InpsTyp" = 1)

Rapid traverse and cutting in-position are designated with the "inpos" parameter.

Parameter #1193 Inpos	Command block		
	G0	G1+G9	G1
<b>0</b>	Command deceleration check	Command deceleration check	No deceleration check
<b>1</b>	In-position check	In-position check	No deceleration check

**(Note)** "#1193 inpos" is the parameter per system.

### 7.11.1 G1 → G0 deceleration check



#### Detailed description

- (1) In G1 → G0 continuous blocks, the parameter "#1502 G0lpfg" can be changed to change the deceleration check in the reverse direction.

	Same direction	Reverse direction
G0lpfg: 0		
G0lpfg: 1		



#### Example of program

When there is a deceleration check in the movement of several axes:

- |     |   |  |
|-----|---|--|
| (1) | G91 G1 X100. Z100. F4000 ;<br>G0 X-100. Z120. ; | A deceleration check is carried out, because the X axis moves in the reverse direction in the program above.   |
| (2) | G91 G1 X100. Z-100. F4000 ;<br>G0 X80. Z100. ;  | A deceleration check is carried out, because the Z axis moves in the reverse direction in the program above.   |
| (3) | G90 G1 X100. Z100. F4000 ;<br>G0 X80. Z120. ;   | (When the program start position is X0 Z0)<br>A deceleration check is carried out, because the X axis moves in the reverse direction in the program above. |
| (4) | G91 G1 X100. Z100. F4000 ;<br>G0 X100. Z100. ;  | A deceleration check is not carried out, because both the X axis and the Z axis move in the same direction in the program above.                           |
| (5) | G91 G1 X100. Z80. F4000 ;<br>G0 X80. ;          | A deceleration check is not carried out, because the X axis moves in the same direction, and there is no Z axis movement command in the program above.     |

### 7.11.2 G1 → G1 deceleration check



#### Detailed description

- (1) In G1 → G1 continuous blocks, the parameter "#1503 G1lpfg" can be changed to change the deceleration check of the reverse direction.

	Same direction	Reverse direction
G1lpfg: 0		
G1lpfg: 1		



#### Example of program

When there is a deceleration check in the movement of several axes:

(1)	G91 G1 X100. Z100. F4000 ; G1 X-100. Z120. ;	A deceleration check is carried out, because the X axis moves in the reverse direction in the program above.
(2)	G91 G1 X100. Z-100. F4000 ; G1 X80. Z100. ;	A deceleration check is carried out, because the Z axis moves in the reverse direction in the program above.
(3)	G90 G1 X100. Z100. F4000 ; G1 X80. Z120. ;	(When the program start position is X0 Z0) A deceleration check is carried out, because the X axis moves in the reverse direction in the program above.
(4)	G91 G1 X100. Z100. F4000 ; G1 X100. Z100. ;	A deceleration check is not carried out, because both the X axis and the Z axis move in the same direction in the program above.
(5)	G91 G1 X100. Z80. F4000 ; G1 X80. ;	A deceleration check is not carried out, because the X axis moves in the same direction, and there is no Z axis movement command in the program above.

### 7.12 Automatic corner override ; G62



#### Function and purpose

With this command, when cutting with tool radius compensation, an override is automatically applied to the feedrate to reduce the load during inside corner cutting or during inside cutting of automatic corner R.

Automatic corner override is valid until the tool radius compensation cancel (G40), exact stop check mode (G61), tapping mode (G63), or cutting mode (G64) command is issued.



#### Command format

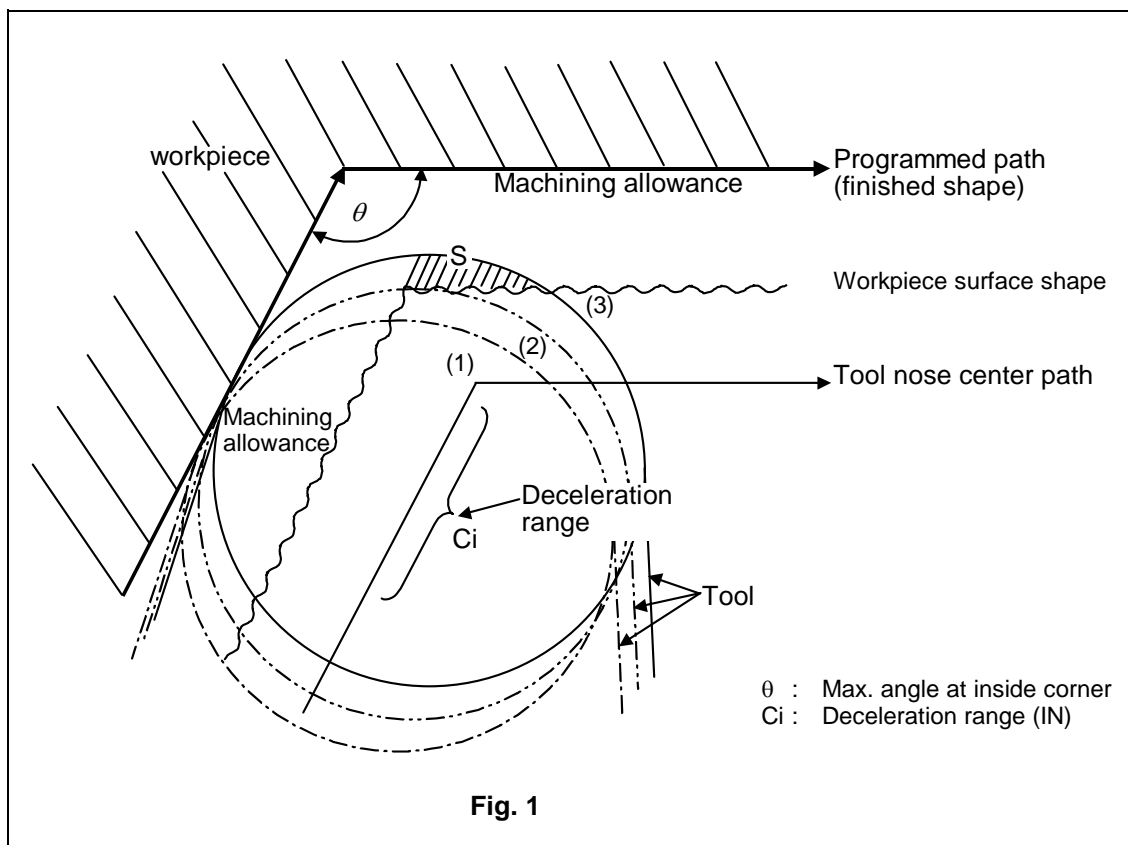
G62 ;



#### Machining inside corners

When cutting an inside corner as in Fig. 1, the machining allowance amount increases and a greater load is applied to the tool. To remedy this, override is applied automatically within the corner set range, the feedrate is reduced, the increase in the load is reduced and cutting is performed effectively.

However, this function is valid only when finished shapes are programmed.



#### (1) Operation

(a) When automatic corner override is not to be applied :

When the tool moves in the order of (1) → (2) → (3) in Fig. 1, the machining allowance at (3) increases by an amount equivalent to the area of shaded section S and so the tool load increases.

(b) When automatic corner override is to be applied :

When the inside corner angle  $\theta$  in Fig. 1 is less than the angle set in the parameter, the override set into the parameter is automatically applied in the deceleration range  $C_i$ .

#### (2) Parameter setting

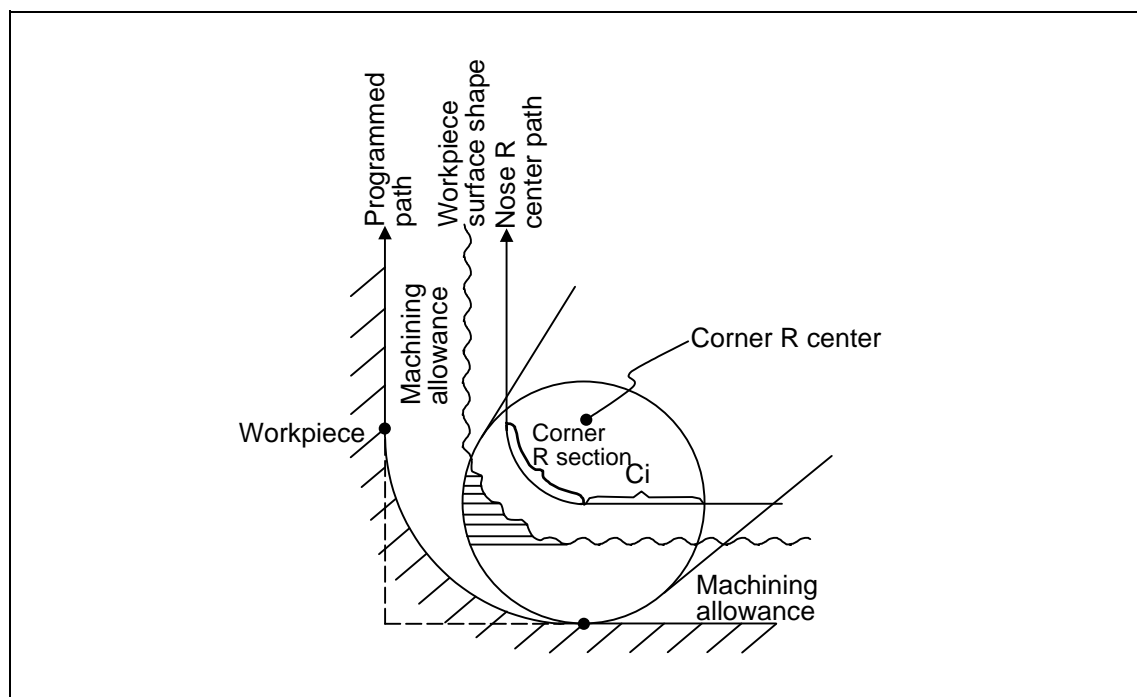
The following parameters are set into the machining parameters :

#	Parameter	Parameter setting
#8007	OVERRIDE	0 to 100%
#8008	MAX. ANGLE	0 to 180°
#8009	DSC. ZONE	0 to 99999.999mm or 0 to 3937.000 inches

Refer to the Operation Manual for details on the setting method.



#### Automatic corner R



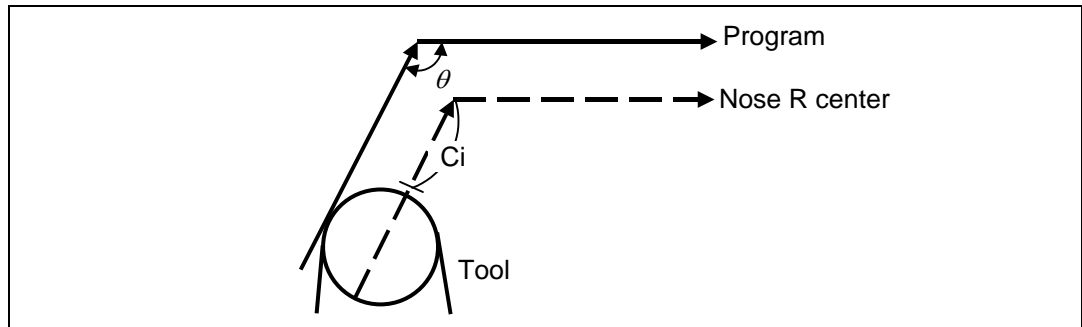
(1) The override set in the parameter is automatically applied at the deceleration range  $C_i$  and corner R section for inside offset with automatic corner R. (There is no angle check.)





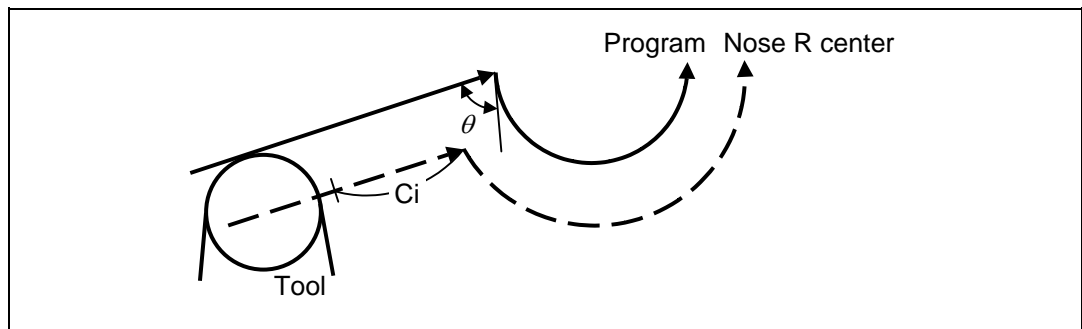
### Application example

#### (1) Linear – linear corner



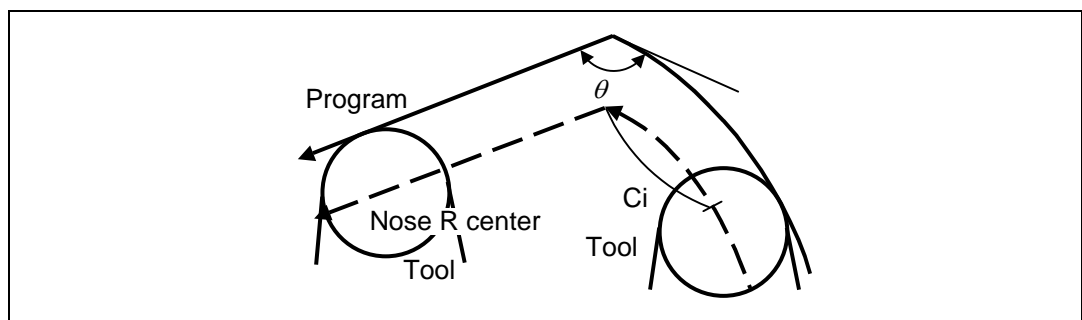
The override set in the parameter is applied at Ci.

#### (2) Linear – arc (outside offset) corner



The override set in the parameter is applied at Ci.

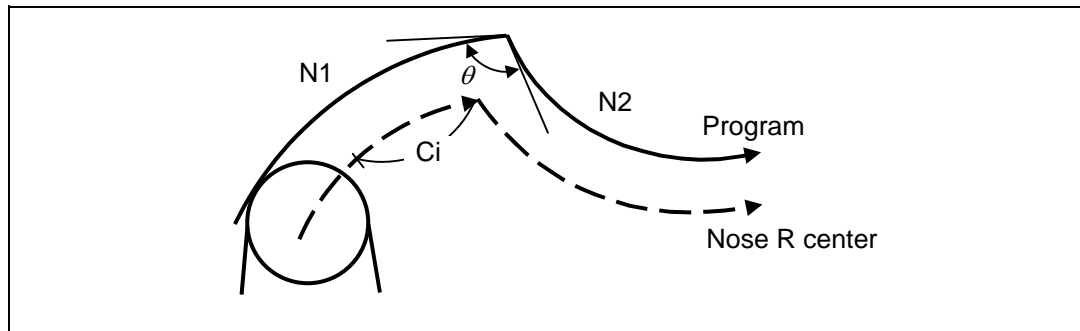
#### (3) Arc (inside offset) – linear corner



The override set in the parameter is applied at Ci.

**(Note)** The deceleration range Ci where the override is applied is the length of the arc with an arc command.

#### (4) Arc (inside offset) – arc (outside offset) corner



The override set in the parameter is applied at Ci.



#### Relation with other functions

Function	Override at corner
Cutting feed override	Automatic corner override is applied after cutting feed override has been applied.
Override cancel	Automatic corner override is not canceled by override cancel.
Speed clamp	Valid after automatic corner override
Dry run	Automatic corner override is invalid.
Synchronous feed	Automatic corner override is applied to the synchronous feedrate.
Thread cutting	Automatic corner override is invalid.
G31 skip	Program error results with G31 command during nose R compensation.
Machine lock	Valid
Machine lock high speed	Automatic corner override is invalid.
G00	Invalid
G01	Valid
G02, G03	Valid



#### Precautions

- (1) Automatic corner override is valid only in the G01, G02, and G03 modes; it is not effective in the G00 mode. When switching from the G00 mode to the G01 (or G02 or G03) mode at a corner (or vice versa), automatic corner override will not be applied at that corner in the G00 block.
- (2) Even if the automatic corner override mode is entered, the automatic corner override will not be applied until the nose R compensation mode is entered.
- (3) Automatic corner override will not be applied on a corner where the nose R compensation is started or canceled.
- (4) Automatic corner override will not be applied on a corner where the nose R compensation I, K vector command is issued.
- (5) Automatic corner override will not be applied when intersection calculation cannot be executed. Intersection calculation cannot be executed in the following case.
  - (a) When the movement command block does not continue for four or more times.
- (6) The deceleration range with an arc command is the length of the arc.
- (7) The inside corner angle, as set by parameter, is the angle on the programmed path.
- (8) Automatic corner override will not be applied when the maximum angle in the parameter is set to 0 or 180.
- (9) Automatic corner override will not be applied when the override in the parameter is set to 0 or 100.

#### 7.13 Tapping mode ; G63



##### Function and purpose

The G63 command allows the control mode best suited for tapping to be entered, as indicated below :

1. Cutting override is fixed at 100%.
2. Deceleration commands at joints between blocks are invalid.
3. Feed hold is invalid.
4. Single block is invalid.
5. In-tapping mode signal is output.

G63 is released by the exact stop check mode (G61), automatic corner override (G62), or cutting mode (G64) command.



##### Command format

G63 ;

#### 7.14 Cutting mode ; G64



##### Function and purpose

The G64 command allows the cutting mode in which smooth cutting surfaces are obtained to be established. Unlike the exact stop check mode (G61), the next block is executed continuously with the machine not decelerating and stopping between cutting feed blocks in this mode.

G64 is released by the exact stop check mode (G61), automatic corner override (G62), or tapping mode (G63) command.

This cutting mode is established in the initialized status.



##### Command format

G64 ;

## 8. Dwell

The G04 command can delay the start of the next block.

### 8.1 Per-second Dwell ; G04



#### Function and purpose

The machine movement is temporarily stopped by the program command to make the waiting time state. Therefore, the start of the next block can be delayed. The waiting time state can be canceled by inputting the skip signal.



#### Command format

**G04 X/U\_\_ ; or G04 P\_\_ ;**  
**X, P, U Dwell time**

For E68, in addition to the address P and X, the address U (actually, the address corresponding to the X-axis designated with the #1014 incax) can be used. Note that this is invalid when the #1076 AbsInc is set to 0.



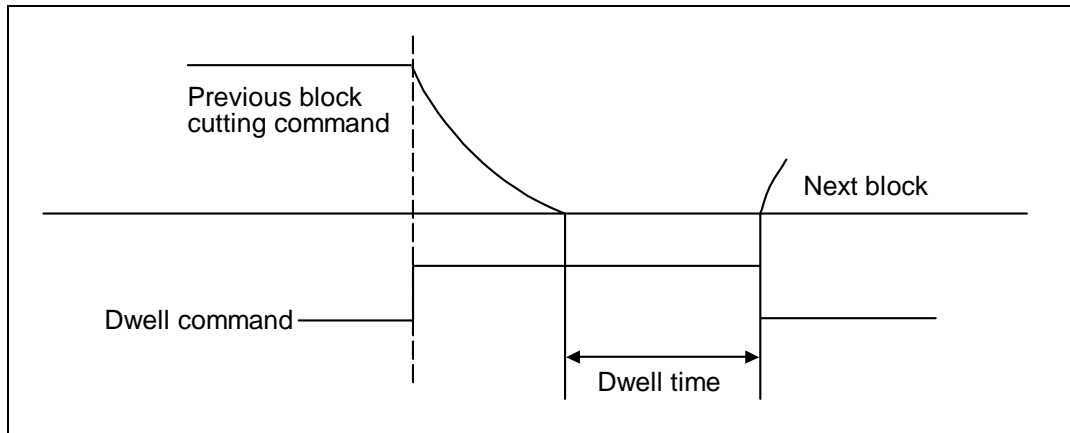
#### Detailed description

- (1) When designating the dwell time with X or U, the decimal point command is valid.
- (2) When designating the dwell time with P, the availability of the decimal point command can be selected with the parameter (#8112). When the decimal point command is invalid in the parameter setting, the command below the decimal point issued with P is ignored.
- (3) When the decimal point command is valid or invalid, the dwell time command range is as follows.

Command range when the decimal point command is valid	Command range when the decimal point command is invalid
0.001 ~ 99999.999 (s)	1 ~ 99999999 (ms)

- (4) The dwell time setting unit applied when there is no decimal point can be made 1s by setting 1 in the parameter #1078 Decpt2. This is effect for X, U and P. However, P can be used only when the decimal point command is validated (#8112=1). Refer to the "Example of program" for details.
- (5) When a cutting command is in the previous block, the dwell command starts calculating the dwell time after the machine has decelerated and stopped. When it is commanded in the same block as an M, S, T or B command, the calculation starts simultaneously.
- (6) The dwell is valid during the interlock.
- (7) The dwell is valid even for the machine lock.

- (8) The dwell can be canceled by setting the parameter #1173 dwlskp beforehand. If the set skip signal is input during the dwell time, the remaining time is discarded, and the following block will be executed.





## Example of program

(1) Address X,U

Command	Dwell time [sec]	
	#1078 Decpt2 = 0	#1078 Decpt2 = 1
G04 X500 ;	0.5	500
G04 X5000 ;	5	5000
G04 X5. ;	5	5
G04 X#100 ;	1000	1000
G04 U500 ;	0.5	500
G04 U5000 ;	5	5000
G04 U5. ;	5	5
G04 U#100 ;	1000	1000

(2) Address P

Command	Dwell time (s)			
	#1078 Decpt2 = 0		#1078 Decpt2 = 1	
	DECIMAL PNT-N	DECIMAL PNT-P	DECIMAL PNT-N	DECIMAL PNT-P
G04 P5000 ;	5		5	5000
G04 P12.345 ;	0.012	12.345	0.012	12.345
G04 P#100 ;	1	1000	1	1000

**(Note 1)** The above examples are the results under the following conditions.  
 • #100 = 1000 ;

**(Note 2)** "G04P decimal point valid/invalid (DECIMAL PNT-P)" depends on the control parameter (#8112) setting.

**(Note 3)** If the input setting unit is "C" (0.0001mm), the X before G04 will be multiplied by 10. For example for "X5. G04 ;", the dwell time will be 50 sec.



## Precautions and restrictions

- (1) Address U can be used in the E68 system.
- (2) When using this function, command X or U after G04 in order to make sure that the dwell is based on X or U.

## 9. Miscellaneous Functions

### 9.1 Miscellaneous functions (M8-digits BCD)

## 9. Miscellaneous Functions

### 9.1 Miscellaneous functions (M8-digits BCD)



#### Function and purpose

The miscellaneous (M) functions are also known as M functions, and they include such numerically controlled machine functions as spindle forward and reverse rotation, operation stop and coolant ON/OFF. These functions are designated by an 8-digit number (0 to 99999999) following the address M with this controller, and up to 4 groups can be commanded in a single block.

**(Example)** G00 Xx Mm1 Mm2 Mm3 Mm4 ;

When five or more commands are issued, only the last four will be valid.

The output signal is an 8-digit BCD code and start signal.

The six commands of M00, M01, M02, M30, M96, M97, M98 and M99 are used as auxiliary commands for specific objectives and so they cannot be used as general auxiliary commands. This therefore leaves 92 miscellaneous functions which are usable as such commands. Reference should be made to the instructions issued by the machine tool builder for the actual correspondence between the functions and numerical values.

When the M00, M01, M02, and M30 functions are used, the next block is not read into the pre-read buffer due to pre-read inhibiting.

When an M function can be specified together with other commands in the same there are two possible sequences in which the commands are executed, as follows. The machine specifications determine which sequence applies.

- (1) The M function is executed after the movement command.
  - (2) The M function is executed at the same time as the movement command.
- Which of these sequences actually applies depends on the machine specifications.

Processing and completion sequences are required in each case for all M commands except M96, M97, M98 and M99. The 8M functions used for specific purposes will now be described.



#### Program stop : M00

When the tape reader has read this function, it stops reading the next block. As far as the NC system's functions are concerned, only the tape reading is stopped. Whether such machine functions as the spindle rotation and coolant supply are stopped or not differs according to the machine in question.

Re-start is enabled by pressing the automatic start button on the machine operation board.

Whether resetting can be initiated by M00 depends on the machine specifications.



## 9. Miscellaneous Functions

### 9.1 Miscellaneous functions (M8-digits BCD)



#### Optional stop : M01

If the tape reader reads the M01 command when the optional stop switch on the machine operation board is ON, it will stop and the same effect as with the M00 function will apply. If the optional stop switch is OFF, the M01 command is ignored.

#### (Example)

<pre>      } N10 G00 X1000 ; N11 M01 ; N12 G01 X2000 Z3000 F600 ;       }</pre>	<p>Optional stop switch command is ignored.</p> <p>Stops at N11 when switch is ON Next command (N12) is executed without stopping at N11 when switch is OFF</p>
---	---



#### Program end : M02 or M30

This command is normally used in the final block for completing the machining, and so it is primarily used for tape rewinding. Whether the tape is actually rewound or not depends on the machine specifications. Depending on the machine specifications, the system is reset by the M02 or M30 command upon completion of tape rewinding and any other commands issued in the same block. (Although the contents of the command position display counter are not cleared by this reset action, the modal commands and compensation amounts are canceled.)

The next operation stops when the rewinding operation is completed (the in-automatic operation lamp goes off). To restart the unit, the automatic start button must be pressed or similar steps must be taken.

**(Note 1)** Independent signals are also output respectively for the M00, M01, M02 and M30 commands and these outputs are each reset by pressing the reset key.

**(Note 2)** M02 or M30 can be assigned by manual data input (MDI). At this time, commands can be issued simultaneously with other commands just as with the tape.



#### Macro interrupt : M96, M97

M96 and M97 are M codes for user macro interrupt control.

The M code for user macro interrupt control is processed internally, and is not output externally. To use M96 and M97 as a miscellaneous function, change the setting to another M code with the parameter (#1109 subs\_M and #1110 M96\_M, #1111 M97\_M).

## 9. Miscellaneous Functions

### 9.1 Miscellaneous functions (M8-digits BCD)



#### Subprogram call/completion : M98, M99

These commands are used as the return instructions from branch destination subprograms and branches to subprograms.

M98 and M99 are processed internally and so M code signals and strobe signals are not output.



#### Internal processing with M00/M01/M02/M30 commands

Internal processing suspends pre-reading when the M00, M01, M02 or M30 command has been read. Other tape rewinding operations and the initialization of modals by resetting differ according to the machine specifications.

## 9. Miscellaneous Functions

### 9.2 2nd miscellaneous functions (A8-digits, B8-digits or C8-digits)

#### 9.2 2nd miscellaneous functions (A8-digits, B8-digits or C8-digits)



##### Function and purpose

These serve to assign the indexing table positioning and other such functions. In this controller, they are assigned by an 8-digit number from 0 to 99999999 following address A, B or C. The machine tool builder determines which codes correspond to which positions.

If the A, B or C function is designated in the same block as a movement command, the commands may be executed in either of the following two orders. The machine specifications determine which sequence applies.

- (1) The A, B or C function is executed after the movement command.
- (2) The A, B or C function is executed simultaneously with the movement command.

Processing and completion sequences are required for all 2nd miscellaneous functions.

The table below given the various address combinations. It is not possible to use an address which is the same for the axis name of an additional axis and 2nd miscellaneous function.

2nd miscellaneous function \ Additional axis name	A	B	C
A	x	○	○
B	○	x	○
C	○	○	x

**(Note)** When "A" has been assigned as the 2nd miscellaneous function address, the following commands cannot be used.

- (1) Linear angle commands
- (2) Geometric I commands
- (3) Deep hole drilling cycle 2 commands

### 9.3 Index Table Indexing

[E68]



#### Function and purpose

Index table indexing can be carried out by setting the index axis. The indexing command only issues the indexing angle to the axis set for indexing. It is not necessary to command special M codes for table clamping and unclamping, thus simplifying the program.



#### Detailed description

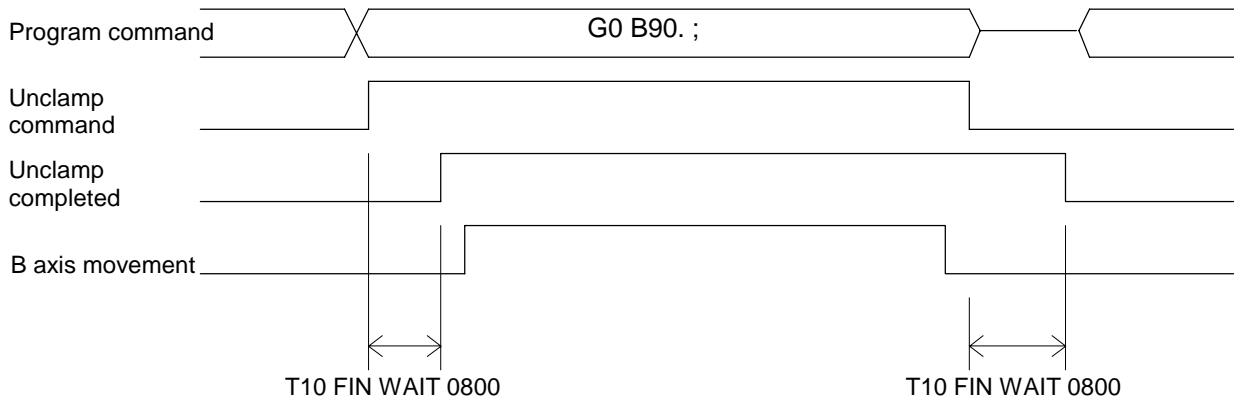
The index table index function carries out operations as follows.

**(Example)** G00 B90 ;

↳ The axis that was designated as the index axis with parameter "#2076 index\_x".

- (1) Set the "index\_x" parameter (#2076) for the axis in which index table indexing will be carried out to "1".
- (2) The movement command (either absolute or incremental) for the selected axis is executed with the program command.
- (3) An unclamp process are carried out before the axis movement.
- (4) The commanded axis movement starts after the unclamp process completes.
- (5) The clamp process is carried out after the movement is completed.
- (6) The next block is processed after the unclamp process completes.

#### <Operation time chart>





#### Precautions

- (1) Several axes can be set as index table indexing axes.
- (2) The movement speed of index table indexing axes follows the feedrate of the modal (G0/G1) at that time.
- (3) The unclamp process for the indexing axes is also issued when the index table indexing axes are commanded in the same block as other axes. Thus, the movement of other axes commanded in the same block is not carried out until the unclamp process completes.  
Note that the movement of other axes commanded in the same block is carried out for non-interpolation commands.
- (4) Index table indexing axes are used as normal rotation axes, but this function performs an unclamp process even for linear axes.
- (5) If some error that makes unclamp command OFF occurs during indexing axis movement in automatic operation, the unclamp state will be remained, and the indexing axis will execute a deceleration stop.  
Other axes commanded in the same block will also execute a deceleration stop, except for non-interpolation commands.
- (6) If the axis movement is interrupted by an interlock, etc., during indexing axis movement, the unclamp state will be remained.
- (7) The clamp and unclamp process are not executed when the movement commands of the index table indexing axis are continuous.  
Note that the clamp and unclamp process are executed even when the movement commands are continued during single block operation.
- (8) Make sure that the command position is at a position where clamping is possible.

### 10. Spindle Functions

#### 10.1 Spindle functions (S function)



##### Function and purpose

The spindle functions are also known simply as S functions and they assign the spindle rotation speed. If the S function is designated in the same block as a movement command, the commands may be executed in either of the following two orders.

- (1) The S function is executed after the movement command.
- (2) The S function is executed simultaneously with the movement command.

#### 10.1.1 Spindle functions (S6-digits Analog)



##### Function and purpose

When the S6-digits function is added, a 6-digit value (0 to 999999) can be designated after the S code. Other commands conform to the S2-digits function. Always select S command binary output when using this function.

In this function, the appropriate gear signal, voltage corresponding to the commanded spindle rotation speed, and start signal are output by the numeric command of the six digits following the S code.

Processing and completion sequences are required for all S commands.

The analog signal specifications are given below.

- |                           |                      |
|---------------------------|----------------------|
| (1) Output voltage.....   | 0 to 10V             |
| (2) Resolution.....       | 1/4096 ( $2^{-12}$ ) |
| (3) Load conditions.....  | 10k $\Omega$         |
| (4) Output impedance..... | 220 $\Omega$         |

If the parameters for up to 4 gear stages are set in advance, the gear stage corresponding to the S command will be selected and the gear signal will be output. The analog voltage is calculated in accordance with the input gear signal.

- |   |   |
|---|---|
| (1) Parameters corresponding to individual gears..... | Limit rotation speed, maximum rotation speed, shift rotation speed and tapping rotation speed |
| (2) Parameters corresponding to all gears.....        | Orientation rotation speed, minimum rotation speed  |

#### 10.1.2 Spindle functions (S8-digits)



##### Function and purpose

These functions are assigned with an 8-digit (0 ~ 99999999) number following the address S, and one group can be assigned in one block.

The output signal is a 32-bit binary data with sign and start signal. Processing and completion sequences are required for all S commands.

### 10.2 Constant surface speed control; G96, G97



#### Function and purpose

These commands automatically control the spindle rotation speed in line with the changes in the radius coordinate values as cutting proceeds in the diametrical direction, and they serve to keep the cutting point speed constant during the cutting.



#### Command format

##### G96 Ss Pp; Constant surface speed ON

Ss	Surface speed
Pp	Designation of constant surface speed control axis

##### G97; Constant surface speed cancel



#### Detailed description

- The constant surface speed control axis is set by parameter (#1181 G96\_ax).  
 0 : Fixed at 1st axis (P command invalid)  
 1 : 1st axis  
 2 : 2nd axis  
 3 : 3rd axis
- When the above-mentioned parameter is not zero, the constant surface speed control axis can be designated by address P.  
**(Example)** When G96\_ax = 1

Program	Constant surface speed control axis
G96 S100 ;	1st axis
G96 S100 P3 ;	3rd axis

- Example of selection program and operation

G90 G96 G01 X50. Z100. S200 ;	}	The spindle rotation speed is controlled so that the surface speed is 200m/min.
}		
G97 G01 X50. Z100. F300 S500 ;	}	The spindle rotation speed is controlled to 500 r/min.
}		
M02 ;		The modal returns to the initial value.

## 10. Spindle Functions

### 10.3 Spindle clamp speed setting

#### 10.3 Spindle clamp speed setting; G92



##### Function and purpose

The maximum clamp rotation speed of the spindle can be assigned by address S following G92 and the minimum clamp rotation speed by address Q.



##### Command format

**G92 Ss Qq ;**

Ss                      Maximum clamp rotation speed

Qq                      Minimum clamp rotation speed

Besides this command, parameters can be used to set the rotation speed range up to 4 stages in 1 r/min units to accommodate gear selection between the spindle and spindle motor.

The lowest upper limit and highest lower limit are valid among the rotation speed ranges based on the parameters and based on "G92 Ss Qq ;".

Set in the parameter (#1146 Sclamp, #1227 aux11/bit5) whether to carry out rotation speed clamp only in the constant surface speed mode or even when the constant surface speed is canceled.

**(Note)** G92S command and rotation speed clamp operation

		Sclamp = 0		Sclamp = 1	
		aux11/bit5 = 0	aux11/bit5 = 1	aux11/bit5 = 0	aux11/bit5 = 1
<b>Command</b>	<b>In G96</b>	ROTATION SPEED CLAMP COMMAND		ROTATION SPEED CLAMP COMMAND	ROTATION SPEED CLAMP COMMAND
	<b>In G97</b>	SPINDLE ROTATION SPEED COMMAND		ROTATION SPEED CLAMP COMMAND	ROTATION SPEED CLAMP COMMAND
<b>Operation</b>	<b>In G96</b>	ROTATION SPEED CLAMP EXECUTION		ROTATION SPEED CLAMP EXECUTION	ROTATION SPEED CLAMP EXECUTION
	<b>In G97</b>	NO ROTATION SPEED CLAMP		ROTATION SPEED CLAMP EXECUTION	NO ROTATION SPEED CLAMP



### 10.4 Spindle/C Axis Control

[E68]



#### Function and purpose

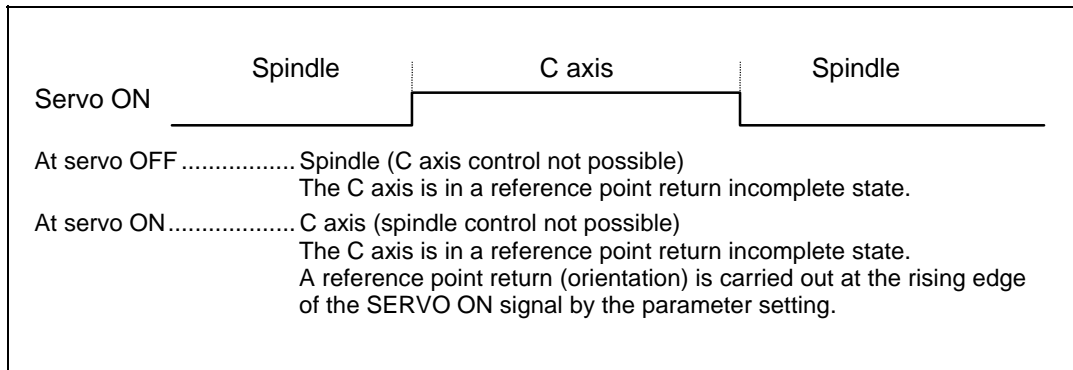
This function enables one spindle to also be used as a C axis (rotation axis) by an external signal.



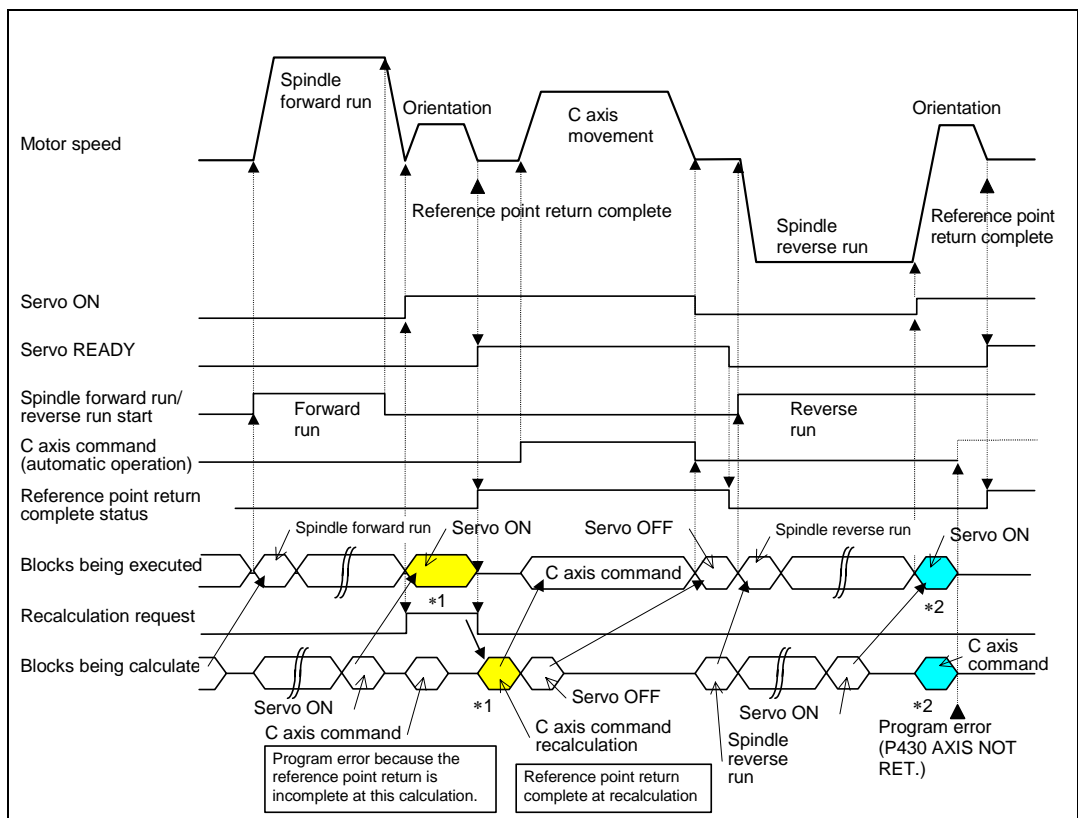
#### Detailed description

##### (1) Spindle/C axis changeover

Changeover between the spindle and C axis is done by the C axis SERVO ON signal.



##### (2) Changeover timing chart example

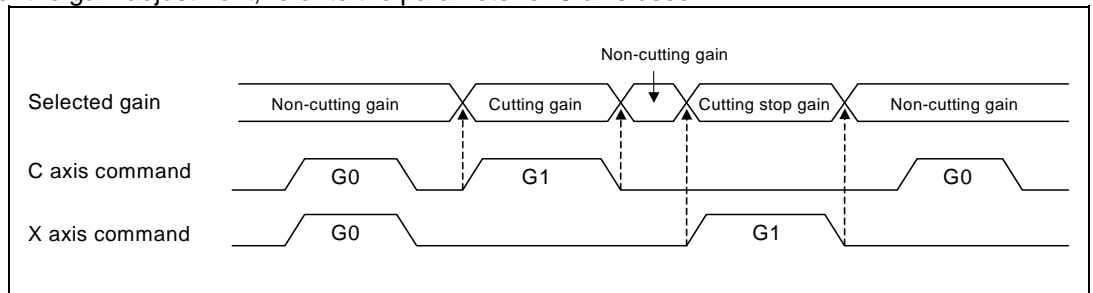


**(Note)** For axis commands, the reference point return complete is checked at calculation. Thus, when the C axis servo ON command and C axis command are continuous, the program error (P430) will occur as shown above in \*2. In response to this kind of situation, the following two processes must be carried out on user PC, as shown above in \*1.

- Input the recalculation request signal with a servo ON command.
- Wait for the completion of the servo ON command until the C axis enters a servo READY state.

### (3) C axis gain

The C axis gain is changed over (the optimum gain is selected) by the C axis cutting condition. During C axis cutting feed, cutting gain is applied. During other axis' cutting feed (C axis face turning), non-cutting stop gain is applied. Non-cutting gain is applied in all other cases. For the gain adjustment, refer to the parameter of C axis used.



**(Note 1)** There are 1st to 3rd cutting gains, which are selected with the ladder.

### (4) Deceleration check in movement including spindle/C-axis

The deceleration check in a movement command including the spindle/C-axis is as the table described below when the following condition is fulfilled.

When the different values are set for the position loop gain in non-cutting mode (spindle parameter #3203 PGCO) and the position loop gain in cutting mode (spindle parameter #3330 PGC1 to #3333 PGC4).

That is because a vibration and so on occurs in the machine when the gain is changed during the axis movement.

Parameter	Rapid traverse command	Parameter	Other than rapid traverse command (G1 : other than G0 command)	
<b>Inpos (#1193)</b>	G0→XX (G0+G9→XX)	<b>AUX07/BIT-1 (#1223/BIT-1)</b>	G1+G9→XX (G1+G9→XX)	G1→G1
<b>0</b>	Command deceleration check	<b>0</b>	In-position check (Applicable only to SV024)	No deceleration check
<b>1</b>	In-position check	<b>1</b>		

**(Note 1)** When G1 command is issued, the in-position check is performed regardless of the deceleration check parameter.

**(Note 2)** XX expresses all commands.

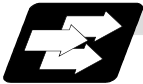


#### Precautions and Restrictions

- (1) A reference point return cannot be executed by the orientation when there is no Z phase in the detector (PLG, ENC, other).  
Replace the detector with one having a Z phase, or if using the detector as it is, set the position control changeover to "After deceleration stop" in the parameters (Spindle parameters "#3329 SP129 bitE: 1"), and set the axis to "Axis without zero point" (Zero point return parameters "#2031 noref: 1").
- (2) The program error (P430) will occur if a C axis command is issued during servo OFF or during orientation.
- (3) Do not execute a servo OFF during a C axis command.  
The remaining C axis commands will be cleared at servo ON.  
(If servo OFF is executed during C axis control, the feed will stop and spindle control will occur.)
- (4) If servo ON is executed during spindle rotation, the rotation will stop and C axis control will occur.
- (5) Dog-type reference point return is not possible for the C axis.  
Set the reference point return to the orientation method in the parameters (Spindle parameters "#3329 SP129 bitE: 0"), or set the axis to "Axis without zero point" (Zero point return parameters "#2031 noref: 1").

### 10.5 Spindle Synchronization Control

[E68]



#### Function and purpose

In a machine having two spindles, this function controls the rotation speed and phase of one spindle (basic spindle) in synchronization with the rotation of the other spindle (synchronous spindle).

The function is used when the rotation speed of the two spindles must be matched, for example, if a workpiece grasped by the 1st spindle is to be grasped by a 2nd spindle, or if the spindle rotation speed has to be changed when one workpiece is grasped by both the 1st and 2nd spindles.

There are two types of spindle synchronization: Spindle synchronization I and Spindle synchronization II.

The spindle synchronization control I	The designation of the synchronous spindle and start/stop of the synchronization are executed by commanding G codes in the machining program.
The spindle synchronization function II	The selections of the synchronized spindle and synchronization start, etc., are all designated from the PLC. Refer to the instruction manual issued by the machine tool builder for details.



#### Common setting for the spindle synchronization control I and II

When the spindle synchronization control is carried out, the followings must be set.

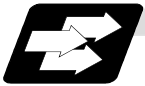
- Chuck close

- Error temporary cancel

- Multi-speed acceleration/deceleration

For details, refer to the "10.5.3 Precautions for Using Spindle Synchronization Control".

### 10.5.1 Spindle Synchronization Control I



#### Function and purpose

With the spindle synchronization control I, the designation of the synchronous spindle and start/stop of the synchronization are executed by commanding G codes in the machining program.



#### Command format

#### (1) Spindle synchronization control ON (G114.1)

This command designates the basic spindle and synchronous spindle, and synchronizes the two designated spindles. By commanding the synchronous spindle phase shift amount, the phases of the basic spindle and synchronous spindle can be aligned.

<b>G114.1 H_ D_ R_ A_ ;</b>	
H_	Basic spindle selection
D_	Synchronous spindle selection
R_	Synchronous spindle phase shift amount
A_	Spindle synchronization acceleration/deceleration time constant

#### (2) Spindle synchronization control cancel (G113)

This command cancels the synchronous state of the two spindles rotating in synchronization with the spindle synchronization command.

**G113 ;**

Address	Meaning of address	Command range (unit)	Remarks
H	Basic spindle selection  Select the No. of the spindle to be used as the basic spindle from the two spindles.	1 to 2  1: 1st spindle 2: 2nd spindle	<ul style="list-style-type: none"> <li>• A program error (P35) will occur if a value exceeding the command range or spindle No. without specifications is commanded.</li> <li>• A program error (P33) will occur if there is no command.</li> <li>• A program error (P700) will occur if a spindle not serially connected is commanded.</li> </ul>
D	Synchronous spindle selection  Select the No. of the spindle to be synchronized with the basic spindle from the two spindles.	1 to 2 or -1 to -2  1: 1st spindle 2: 2nd spindle	<ul style="list-style-type: none"> <li>• A program error (P35) will occur if a value exceeding the command range is commanded.</li> <li>• A program error (P33) will occur if there is no command.</li> <li>• A program error (P33) will occur if the same spindle as that commanded for the basic spindle selection is designated.</li> <li>• The rotation direction of the synchronous spindle in respect to the basic spindle is commanded with the D sign.</li> <li>• A program error (P700) will occur if a spindle not serially connected is commanded.</li> </ul>

Address	Meaning of address	Command range (unit)	Remarks
R	Synchronous spindle phase shift amount  Command the shift amount from the reference point (one rotation signal) of the synchronous spindle.	0 to 359.999 (°) or 0 to 35999 (° × 10 <sup>-3</sup> )	<ul style="list-style-type: none"> <li>• A program error (P35) will occur if a value exceeding the command range is commanded.</li> <li>• The commanded shift amount is effective in the clockwise direction of the basic spindle.</li> <li>• The commanded shift amount's minimum resolution is as follows: For semi-closed (Only gear ratio 1:1) 360/4096 (°) For full closed (360/4096) * K (°) K: Spindle and encoder gear ratio</li> <li>• If there is no R command, the phases will not be aligned.</li> </ul>
A	Spindle synchronization acceleration/deceleration time constant  Command the acceleration/deceleration time constant for when the spindle synchronization command rotation speed changes. (Command this to accelerate or decelerate at a speed slower than the time constant set in the parameters.)	0.001 to 9.999 (s) or 1 to 9999 (ms)	<ul style="list-style-type: none"> <li>• A program error (P35) will occur if a value exceeding the command range is commanded.</li> <li>• If the commanded value is smaller than the acceleration/deceleration time constant set with the parameters, the value set in the parameters will be applied.</li> </ul>



#### Rotation and rotation direction

- (1) The rotation speed and rotation direction of the basic spindle and synchronous spindle during spindle synchronization control are the rotation speed and rotation direction commanded for the basic spindle. Note that the rotation direction of the synchronous spindle can be reversed from the basic spindle through the program.
- (2) The basic spindle's rotation speed and rotation direction can be changed during spindle synchronization control.
- (3) The synchronous spindle's rotation command is also valid during spindle synchronization control.

When spindle synchronization control is commanded, if neither a forward run command nor reverse run command is commanded for the synchronous spindle, the synchronization standby state will be entered without starting the synchronous spindle's rotation. If the forward run command or reverse run command is input in this state, the synchronous spindle will start rotation. The synchronous spindle's rotation direction will follow the direction commanded in the program.

If spindle stop is commanded for the synchronous spindle during spindle synchronization control (when both the forward run and reverse run commands are turned OFF), the synchronous spindle rotation will stop.

- (4) The rotation speed command (S command) and constant surface speed control are invalid for the synchronous spindle during spindle synchronization control. Note that the modal is updated, so these will be validated when the spindle synchronization is canceled.
- (5) The constant surface speed can be controlled by issuing a command to the basic spindle even during spindle synchronization control.

## 10. Spindle Functions

### 10.5 Spindle Synchronization Control

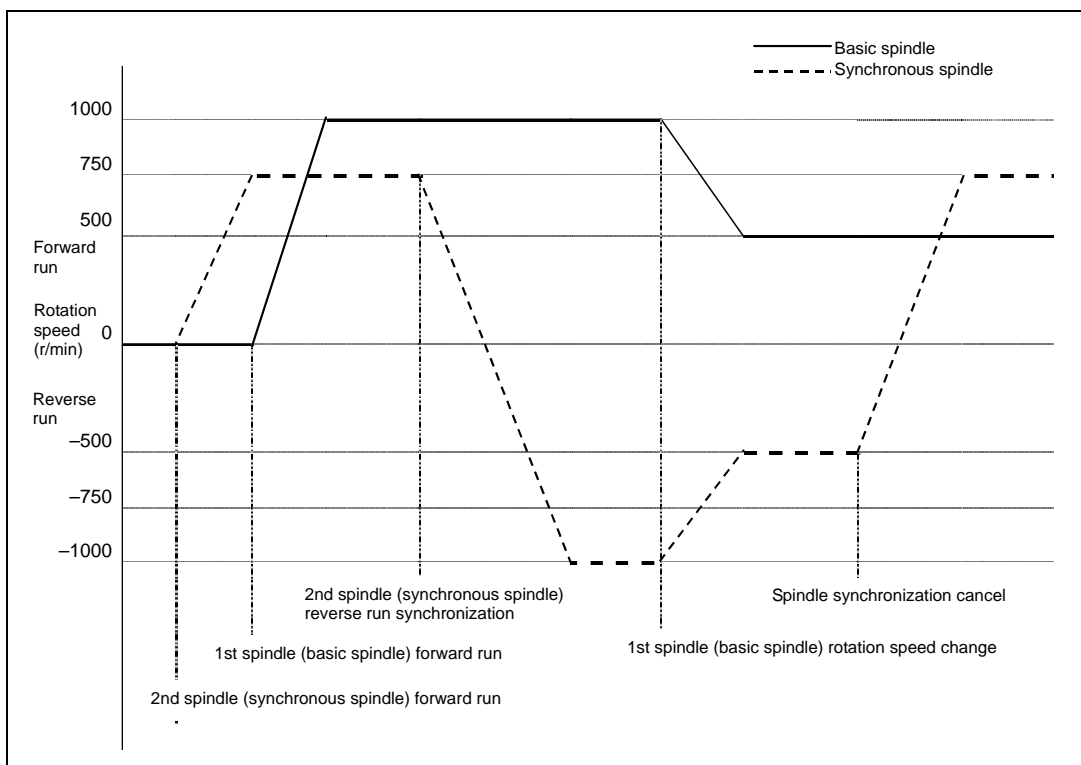


#### Rotation synchronization

- (1) When rotation synchronization control (command with no R address) is commanded with the G114.1 command, the synchronous spindle rotating at a random rotation speed will accelerate or decelerate to the rotation speed commanded beforehand for the basic spindle, and will enter the rotation synchronization state.
- (2) If the basic spindle's commanded rotation speed is changed during the rotation synchronization state, acceleration/deceleration will be carried out while maintaining the synchronization state following the spindle acceleration/deceleration time constants set in the parameters, and the commanded rotation speed will be achieved.
- (3) In the rotation synchronization state, the basic spindle can be controlled to the constant surface speed even when two spindles are grasping one workpiece.
- (4) Operation will take place in the following manner.

M23 S2=750 ;	...	Forward rotate 2nd spindle (synchronous spindle) at 750 r/min (speed command)
:		
M03 S1=1000 ;	...	Forward rotate 1st spindle (basic spindle) at 1000 r/min (speed command)
:		
G114.1 H1 D-2 ;	...	Synchronize 2nd spindle (synchronous spindle) to 1st spindle (basic spindle) with reverse run
:		
S1=500 ;	...	Change 1st spindle (basic spindle) rotation speed to 500 r/min
:		
G113 ;	...	Cancel spindle synchronization

#### <Operation>



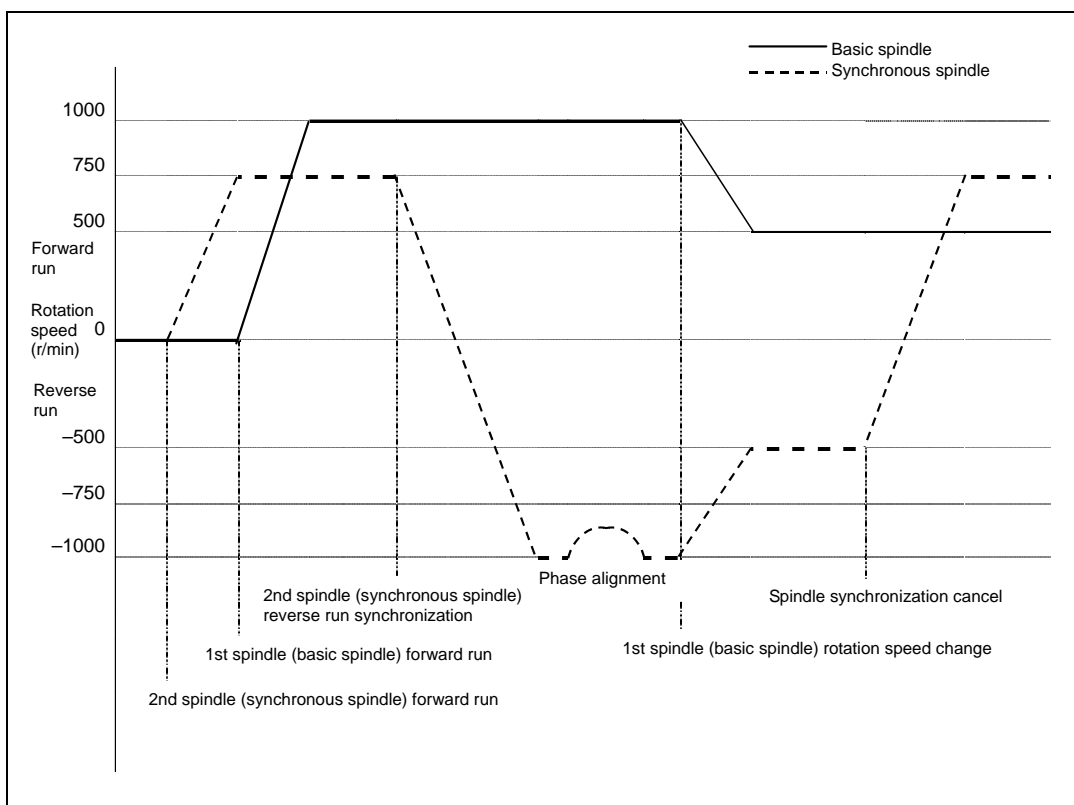


### Phase synchronization

- (1) When phase synchronization (command with R address) is commanded with the G114.1 command, the synchronous spindle rotating at a random rotation speed will accelerate or decelerate to the rotation speed commanded beforehand for the basic spindle, and will enter the rotation synchronization state. Then, the phase is aligned so that the rotation phase commanded with the R address is reached, and the phase synchronization state is entered.
- (2) If the basic spindle's commanded rotation speed is changed during the phase synchronization state, acceleration/deceleration will be carried out while maintaining the synchronization state following the spindle acceleration/deceleration time constants set in the parameters, and the commanded rotation speed will be achieved.
- (3) In the phase synchronization state, the basic spindle can be controlled to the constant surface speed even when two spindles are grasping one workpiece.
- (4) Operation will take place in the following manner.

M23 S2=750 ;	...	Forward rotate 2nd spindle (synchronous spindle) at 750 r/min (speed command)
:		
M03 S1=1000 ;	...	Forward rotate 1st spindle (basic spindle) at 1000 r/min (speed command)
:		
G114.1 H1 D-2 Rxx ;	...	Synchronize 2nd spindle (synchronous spindle) to 1st spindle (basic spindle) with reverse run
:		Shift phase of synchronous spindle by R command value
:		
S1=500 ;	...	Change 1st spindle (basic spindle) rotation speed to 500 r/min
:		
G113 ;	...	Cancel spindle synchronization

### <Operation>







### Cautions on programming

- (1) To enter the rotation synchronization mode while the basic spindle and synchronous spindle are chucking the same workpiece, turn the basic spindle/synchronous spindle rotation commands ON before turning the spindle synchronization control mode ON.

```

:
M6 ;          1st spindle chuck close
:
M5 S1=0 ;    1st spindle stops at S=0
M3 ;        1st spindle rotation command ON
:
M15;        2nd spindle chuck close
M25 S2=0;   2nd spindle stops at S=0
M24;        2nd spindle rotation command ON
:
G114.1 H1 D-2; Rotation synchronization mode ON
:
S1=1500 ;   Synchronous rotation at S=1500
:
S1=0 ;      Both spindles stop
G113 ;     Synchronization mode OFF
    
```


- (2) To chuck the same workpiece with the basic spindle and synchronous spindle in the phase synchronization mode, align the phases before chucking.

```

:
M6 ;          1st spindle chuck close
:
M3 S1=1500 ;  1st spindle rotation command ON
:
:
G114.1 H1 D-2 R0 ; Phase synchronization mode ON
:
M24 ;        2nd spindle rotation command ON
:
M15 ;        2nd spindle chuck close (Note 1)
    
```

- (Note 1) Close the chuck after confirming that the spindle phase synchronization complete signal (X30A) has turned ON (phase alignment complete).

### CAUTION

 Do not make the synchronous spindle rotation command OFF with one workpiece chucked by the basic spindle and synchronous spindle during the spindle synchronous control mode. Failure to observe this may cause the synchronous spindle stop, and hazardous situation.



## Precautions and restrictions

- (1) The spindle rotating with spindle synchronization control will stop when emergency stop is applied.
- (2) The rotation speed clamp during spindle synchronization control will follow the smaller clamp value set for the basic spindle or synchronous spindle.
- (3) Orientation of the basic spindle and synchronous spindle is not possible during the spindle synchronization control mode. To carry out orientation, cancel the spindle synchronization control mode first.
- (4) The rotation speed command (S command) is invalid for the synchronous spindle during the spindle synchronization control mode. Note that the modal will be updated, so this will be validated when spindle synchronization control is canceled.
- (5) The constant surface speed control is invalid for the synchronous spindle during the spindle synchronization control mode. Note that the modal will be updated, so this will be validated when spindle synchronization is canceled.
- (6) The rotation speed command (S command) and constant surface speed control for the synchronous spindle will be validated when spindle synchronization control is canceled. Thus, the synchronous spindle may carry out different operations when this control is canceled.
- (7) If the phase synchronization command (command with R address) is issued while the phase shift calculation request signal is ON, an operation error (1106) will occur.
- (8) If the phase shift calculation request signal is ON and the basic spindle or synchronous spindle is rotation while rotation synchronization is commanded, an operation error (1106) will occur.
- (9) If the phase synchronization command R0 (<Ex.> G114.1 H1 D-2 R0) is commanded while the phase offset request signal is ON, the basic spindle and synchronous spindle phases will be aligned to the phase error of the basic spindle and synchronous spindle saved in the NC memory.
- (10) If a value other than the phase synchronization command R0 (<Ex.> G114.1 H1 D-2 R000) is commanded while the phase offset request signal is ON, the phase error obtained by adding the value commanded with the R address command to the phase difference of the basic spindle and synchronous spindle saved in the NC memory will be used to align the basic spindle and synchronous spindle.
- (11) The phase offset request signal will be ignored when the phase shift calculation request signal is ON.
- (12) The phase error of the basic spindle and synchronous spindle saved in the NC is valid only when the phase shift calculation signal is ON and for the combination of the basic spindle selection (H\_) and synchronous spindle (D\_) commanded with the rotation synchronization command (no R address).  
For example, if the basic spindle and synchronous spindle phase error is saved as "G114.1 H1 D-2 ;", the saved phase error will be valid only when the phase offset request signal is ON and "G114.1 H1 D\_2 R\*\*\* ;" is commanded. If "G114.1 H2 D-1 R\*\*\* ;" is commanded in this case, the phase shift amount will not be calculated correctly.
- (13) When the spindle synchronization commands are being issued with the PLC I/F method (#1300 ext36/bit7 OFF), a program error (P610) will occur if the spindle synchronization control is commanded with G114.1/G113.
- (14) Always execute the "Chuck close". If the "Chuck close" is not executed, an excessive load may be applied on the machine or an alarm may occur.

### 10.5.2 Spindle Synchronization Control II



#### Function and purpose

With the spindle synchronous control II, selection of the synchronized spindle and synchronization start, etc., are all designated from the PLC.



#### Basic spindle and synchronous spindle selection

Select the basic spindle and synchronous spindle for synchronization control from the PLC.

Device No.	Signal name	Abbrev.	Explanation
R446	Basic spindle selection	-	Select a serially connected spindle to be controlled as the basic spindle. (0: 1st spindle), 1: 1st spindle, 2: 2nd spindle. <b>(Note 1)</b> Spindle synchronization control will not take place if a spindle not connected in serial is selected. <b>(Note 2)</b> If "0" is designated, the 1st spindle will be controlled as the basic spindle.
R447	Synchronous spindle selection	-	Select a serially connected spindle to be controlled as the synchronous spindle. (0: 2nd spindle), 1: 1st spindle, 2: 2nd spindle. <b>(Note 3)</b> Spindle synchronization control will not take place if a spindle not connected in serial is selected or if the same spindle as the basic spindle is selected. <b>(Note 4)</b> If "0" is designated, the 2nd spindle will be controlled as the synchronous spindle.



### Starting spindle synchronization

The spindle synchronization control mode is entered by inputting the spindle synchronization control signal (SPSYC). The synchronous spindle will be controlled in synchronization with the rotation speed commanded for the basic spindle during the spindle synchronization control mode. When the difference of the basic spindle and synchronous spindle rotation speeds reaches the spindle synchronization rotation speed reach level setting value (#3050 sprlv), the spindle rotation speed synchronization complete signal (FSPRV) will be output.

The synchronous spindle's rotation direction is designated with the spindle synchronization rotation direction designation as the same as the basic spindle or the reverse direction.

Device No.	Signal name	Abbrev.	Explanation
Y398	Spindle synchronization control signal	SPSYC	The spindle synchronization control mode is entered when this signal turns ON.
X308	In spindle synchronization control	SPSYN1	This notifies that the mode is the spindle synchronization control.
X309	Spindle rotation speed synchronization complete signal	FSPRV	This turns ON when the difference of the basic spindle and synchronous spindle rotation speeds reaches the spindle rotation speed reach level setting value during the spindle synchronization control mode. This signal turns OFF when the spindle synchronization control mode is canceled, or when an error exceeding the spindle rotation speed reach level setting value occurs during the spindle synchronization control mode.
Y39A	Spindle synchronization rotation direction designation	-	Designate the basic spindle and synchronous spindle rotation directions for spindle synchronization control. 0: The synchronous spindle rotates in the same direction as the basic spindle. 1: The synchronous spindle rotates in the reverse direction of the basic spindle.

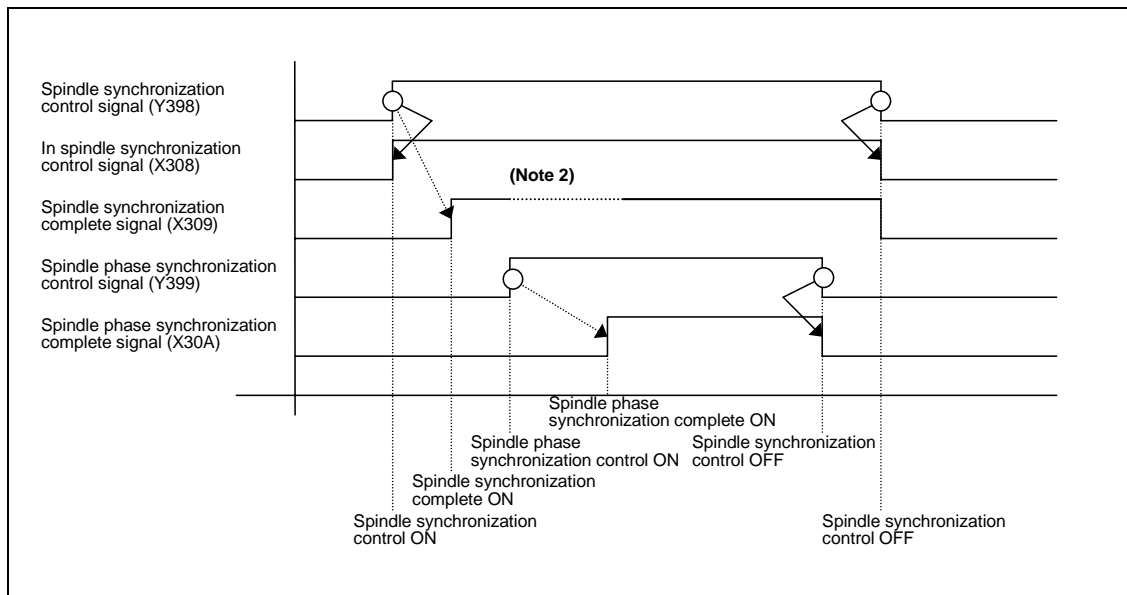


### Spindle phase alignment

Spindle phase synchronization starts when the spindle phase synchronization control signal (SPPHS) is input during the spindle synchronization control mode. The spindle phase synchronization complete signal is output when the spindle synchronization phase reach level setting value (#3051 spplv) is reached.

The synchronous spindle's phase shift amount can also be designated from the PLC.

Device No.	Signal name	Abbrev.	Explanation
Y399	Spindle phase synchronization control signal	SPPHS	Spindle phase synchronization starts when this signal is turned ON during the spindle synchronization control mode. <b>(Note 1)</b> If this signal is turned ON in a mode other than the spindle synchronization control mode, it will be ignored.
X30A	Spindle phase synchronization complete signal	FSPPH	This signal is output when the spindle synchronization phase reach level is reached after starting spindle phase synchronization.
R448	Phase shift amount setting	-	Designate the synchronous spindle's phase shift amount. Unit: 360°/4096



**(Note 2)** Turns OFF temporarily to change the rotation speed during phase synchronization.

## 10. Spindle Functions

### 10.5 Spindle Synchronization Control

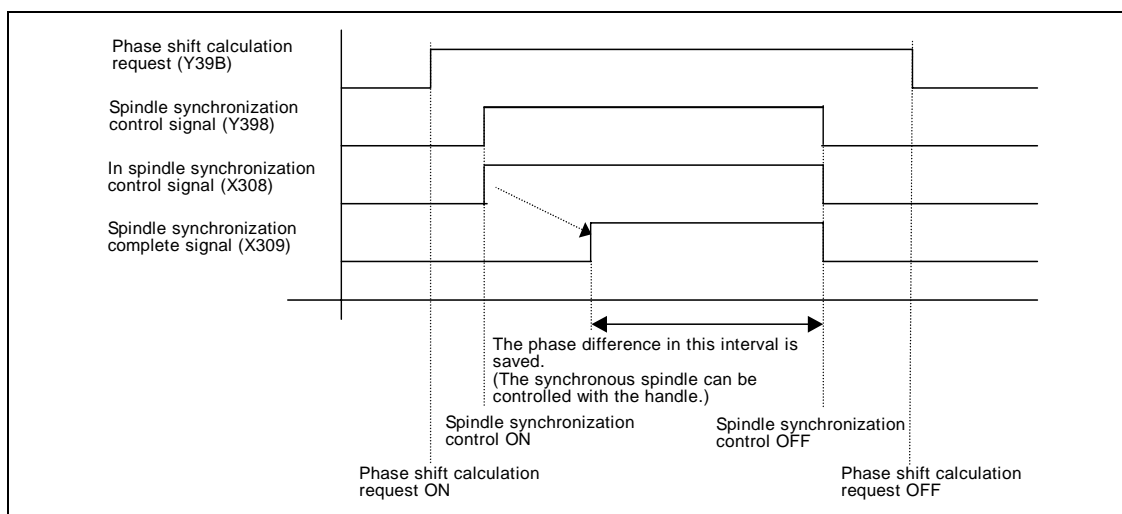


#### Calculating the spindle synchronization phase shift amount and requesting phase offset

The spindle phase shift amount calculation function obtains and saves the phase difference of the basic spindle and synchronous spindle by turning the PLC signal ON during spindle synchronization. When calculating the spindle phase shift, the synchronous spindle can be rotated with the handle, so the relation of the phases between the spindles can also be adjusted visually. If the spindle phase synchronization control signal is input while the phase offset request signal (SSPHF) is ON, the phases will be aligned using the position shifted by the saved phase shift amount as a reference.

This makes aligning of the phases easier when grasping the material that the shape of one end differ from another end.

Device No.	Signal name	Abbrev.	Explanation
Y39B	Phase shift calculation request	SSPHM	If spindle synchronization is carried out while this signal is ON, the phase difference of the basic spindle and synchronous spindle will be obtained and saved.
Y39C	Phase offset request	SSPHF	If spindle phase synchronization is carried out while this signal is ON, the phases will be aligned using the position shifted by the saved phase shift amount as a basic position.
R474	Phase difference output	-	The delay of the synchronous spindle in respect to the basic spindle is output. Unit: $360^\circ/4096$ <b>(Note 1)</b> If either the basic spindle or synchronous spindle has not passed through the Z phase, etc., and the phase cannot be calculated, -1 will be output. <b>(Note 2)</b> This data is output only while calculating the phase shift or during spindle phase synchronization.
R490	Phase offset data	-	The phase difference saved with phase shift calculation is output. Unit: $360^\circ/4096$ <b>(Note 3)</b> This data is output only during spindle synchronization control.



**(Note 4)** The phases cannot be aligned while calculating the phase shift.


**(Note 5)** The synchronous spindle cannot be rotated with the handle when the manual operation mode is set to the handle mode.



#### Precautions and restrictions

- (1) When carrying out spindle synchronization, a rotation command must be issued to both the basic spindle and synchronous spindle. The synchronous spindle's rotation direction will follow the basic spindle rotation direction and spindle synchronization rotation direction designation regardless of whether a forward or reverse run command is issued.
- (2) The spindle synchronization control mode will be entered even if the spindle synchronization control signal is turned ON while the spindle rotation speed command is ON. However, synchronous control will not actually take place. Synchronous control will start after the rotation command has been issued to the basic spindle, and then the spindle synchronization complete signal will be output.
- (3) The spindle rotating with spindle synchronization control will stop when emergency stop is applied.
- (4) An operation error will occur if the spindle synchronization control signal is turned ON while the basic spindle and synchronous spindle designations are illegal.
- (5) The rotation speed clamp during spindle synchronization control will follow the smaller clamp value set for the basic spindle or synchronous spindle.
- (6) Orientation of the basic spindle and synchronous spindle is not possible during the spindle synchronization. To carry out orientation, turn the spindle synchronization control signal OFF first.
- (7) The rotation speed command is invalid for the synchronous spindle during the spindle synchronization. The commanded rotation speed will be validated after spindle synchronization is canceled.
- (8) The constant surface speed control is invalid for the synchronous spindle during the spindle synchronization.
- (9) If the phase offset request signal is turned ON before the phase shift is calculated and then spindle phase synchronization is executed, the shift amount will not be calculated.
- (10) The spindle Z-phase encoder position parameters are invalid when phase offset is carried out.
- (11) If spindle phase synchronization is started while the phase shift calculation request signal is ON, the error "M01 operation error 1106" will occur.
- (12) Turn the phase shift calculation request signal ON when the basic spindle and synchronous spindle are both stopped. If the phase shift calculation request signal is ON while either of the spindles is rotating, the error "M01 operation error 1106" will occur.
- (13) The phase shift amount saved in the NC is held until the next phase shift is calculated. (This value is saved even when the power is turned OFF.)
- (14) Always execute the "Chuck close". If the "Chuck close" is not executed, an excessive load may be applied on the machine or an alarm may occur.

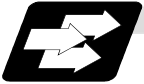
#### CAUTION

-  Do not make the synchronous spindle rotation command OFF with one workpiece chucked by the basic spindle and synchronous spindle during the spindle synchronous control mode. Failure to observe this may cause the synchronous spindle stop, and hazardous situation.

# 10. Spindle Functions

## 10.5 Spindle Synchronization Control

### 10.5.3 Precautions for Using Spindle Synchronization Control



#### Function and purpose

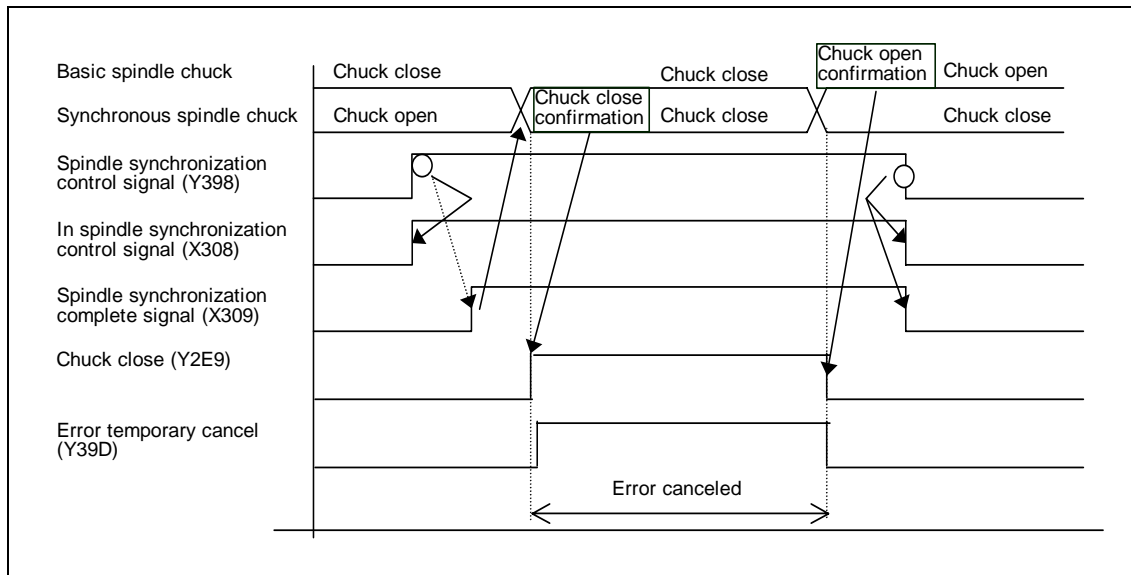
Some PLC signals must be set when spindle synchronization control I or II is used. If these signals are not set, an excessive load or an alarm may occur. Refer to the instruction manual issued by the machine tool builder for details. In this section, each function and the signal are explained.



#### Chuck close signal

The synchronous spindle side carries out droop compensation while the chuck is opened, and aligns itself with the basic spindle. However, when the chuck is closed, the droop compensation is added, and the synchronization error with the base increases. Droop compensation is prevented with the chuck close signal and the position where the chuck is grasped is maintained with position compensation.

Device No.	Signal name	Abbrev.	Explanation
Y2E9	Chuck close signal	-	This turns ON when the chuck is closed. When this signal turns ON, the compensation between the basic spindle and synchronous spindle will change from droop compensation to position compensation.
X30E	Chuck close confirmation signal	-	This turns ON when the chuck close signal is received during the spindle synchronization control mode.



**(Note 1)** Use the error temporary cancel only when there is still an error between the spindle and synchronization with the chuck close signal.



## 10. Spindle Functions

### 10.5 Spindle Synchronization Control



#### Error temporary cancel function

When spindle synchronization is carried out while grasping the workpiece with the basic spindle and rotating, if the chuck is closed to grasp the workpiece with the synchronous spindle, the speed will fluctuate due to external factors and an error will occur. If spindle synchronization is continued without compensating this error, the workpiece will twist.

This torsion can be prevented by temporarily canceling this error.

Device No.	Signal name	Abbrev.	Explanation
Y39D	Error temporary cancel signal	SPDRP0	The error is canceled when this signal is ON.



#### Phase error monitor

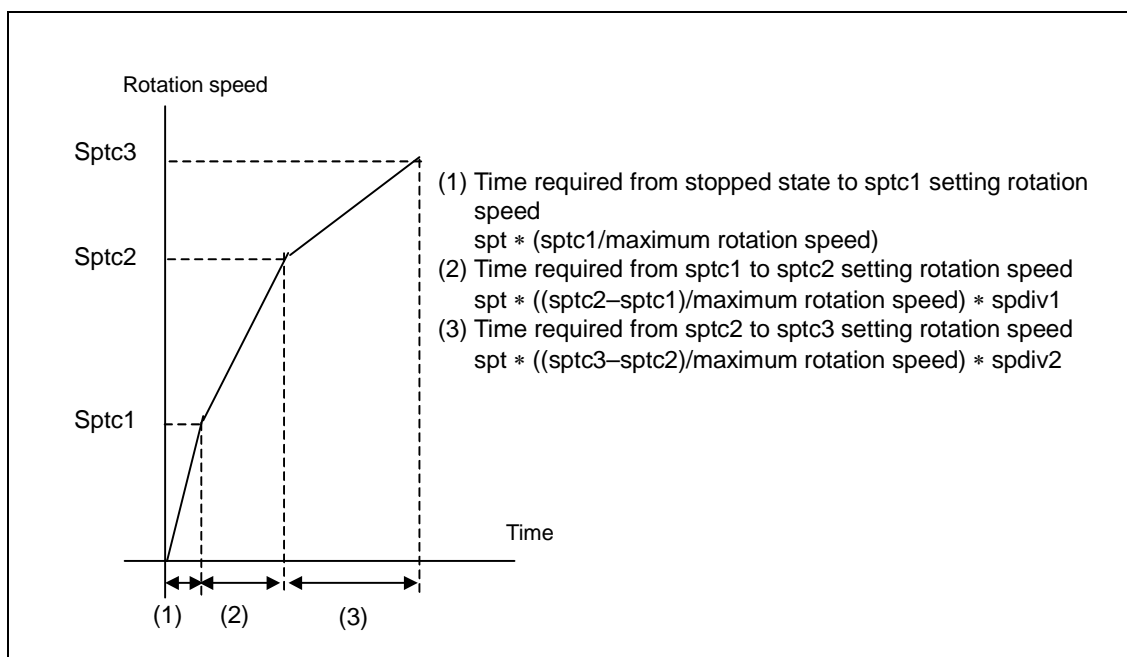
The phase error can be monitored during spindle phase synchronization.

Device No.	Signal name	Abbrev.	Explanation
R477	Phase error monitor	-	The phase error during spindle phase synchronization control is output as a pulse unit.
R478	Phase error monitor (lower limit value)	-	The lower limit value of the phase error during spindle phase synchronization control is output as a pulse unit.
R479	Phase error monitor (upper limit value)	-	The upper limit value of the phase error during spindle phase synchronization control is output as a pulse unit.



#### Multi-speed acceleration/deceleration

Up to eight steps of acceleration/deceleration time constants for spindle synchronization can be selected according to the spindle rotation speed.



### 10.6 Multiple Spindle Control

[E68]



#### Function and purpose

Multiple spindle control is a function used to control the sub-spindle in a machine tool that has a main spindle (1st spindle) and a sub-spindle (2nd spindle).

The spindle control method differs between multiple spindle control I and multiple spindle control II. Whether to use multiple spindle control I or multiple spindle control II depends on the parameter (#1300 ext36/bit0) setting.

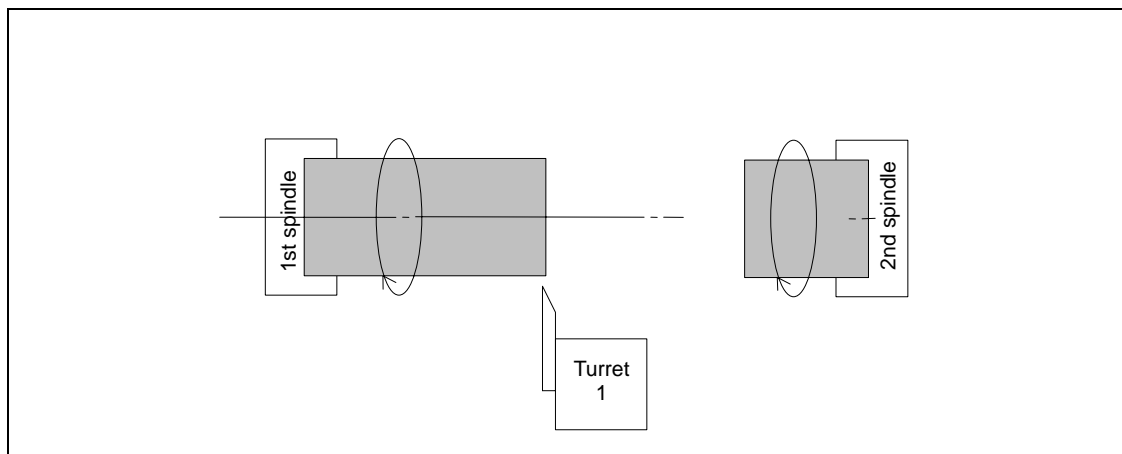
Multiple spindle control I:  
(ext36/bit0 = 0)

Control following the spindle selection command (G43.1, etc.) and spindle control command ([S\*\*\*\*\* ;] or [SO=\*\*\*\*\* ;]), etc.

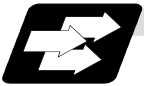
Multiple spindle control II:  
(ext36/bit0 = 1)

Control following the external signal (spindle command selection signal, spindle selection signal) and spindle control command ([S\*\*\*\*\* ;] only), etc.

The spindle selection command [SO=\*\*\*\*\* ;] cannot be used.



### 10.6.1 Multiple Spindle Control I (Multiple Spindle Command)



#### Function and purpose

In addition to the S\*\*\*\*\* command, the 1st and 2nd spindles can be identified and commanded with the SO=\*\*\*\*\* command.



#### Command format

<b>SO=***** ;</b>	
○	Designate the spindle number with one numeric character (1: 1st spindle/2: 2nd spindle). A variable can be designated.
*****	Rotation speed or constant surface speed command. A variable can be designated.

**(Note 1)** A program error (P35) will occur if the ○ value is not 1 or 2.

**(Note 2)** A program error (P33) will occur if G47.1 is issued in a modal.



#### Detailed description

(1) Each spindle command is delimited by the details of ○.

**(Example)**

S1 = 3500 ; 1st spindle 3500(r/min) command  
 S2 = 1500 : 2nd spindle 1500(r/min) command

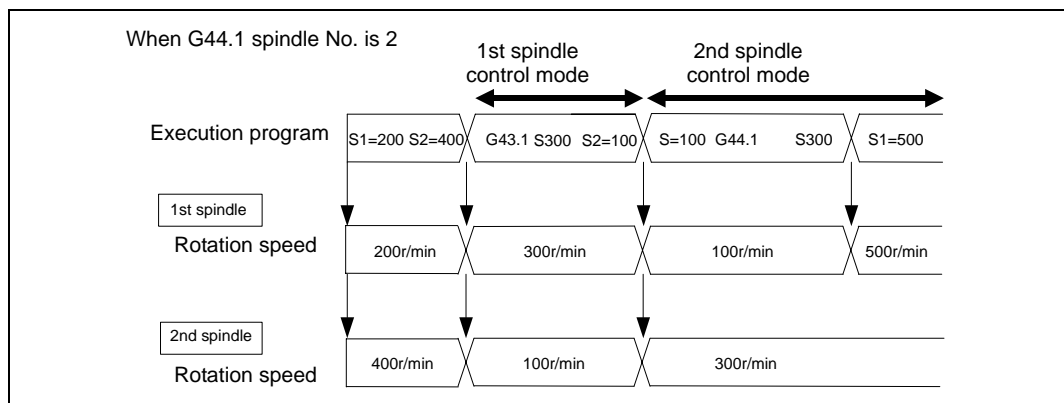
(2) Multiple spindles can be commanded in one block.

(3) If two or more commands are issued to the same spindle in a block, the command issued last will be valid.

**(Example)** S1 = 3500 S1 = 3600 S1 = 3700 ; S1 = 3700 will be valid.

(4) The S\*\*\*\*\* command and SO=\*\*\*\*\* command can be used together.

The spindle targeted for the S\*\*\*\*\* command is delimited by the spindle selection command.



### 10.6.2 Multiple Spindle Control I (Spindle Selection Command)



#### Function and purpose

The following S commands (S\*\*\*\*) can be changed for the 1st spindle or 2nd spindle by issuing the spindle selection command (G43.1, etc. [G group 20]).



#### Command format

<b>G43.1;</b>	<b>1st spindle control mode ON</b>
<b>G44.1;</b>	<b>Selected spindle control mode ON (Selected spindle No. is set with SnG44.1)</b>
<b>G47.1;</b>	<b>All spindles simultaneous control mode ON</b>



#### Detailed description

- (1) The selected spindle No. is set with the parameter (#1534 SnG44.1).
- (2) The spindle selection commands are modal G codes.
- (3) A program error (P33) will occur if the spindle selection command is issued while the multiple spindle control II mode is valid.
- (4) The spindle control mode entered when the power is turned ON or reset can be set with the parameters. The state when the power is turned ON or reset is as follows.

G group 20 modal state	Set with setup parameter "#1199 Sselect". 0:G43.1 1:G44.1 2:G47.1
G44.1 spindle No.	Set with setup parameter "#1534 SnG44.1". 0:2nd spindle 1:1st spindle 2:2nd spindle

- (5) If the S command is issued in the same block as the spindle selection command, it will be valid for the spindle switched to with the spindle selection command.

#### Example)

G43.1 S100 ;	→	Command 100 rotations for 1st spindle
S200 G44.1 S300 ;	→	After commanding 200 rotations for 1st spindle, command 300 rotations for 2nd spindle
(SnG44.1=2)		

- (6) If a spindle that does not exist is set, the 2nd spindle will be used. Note that if there is only one spindle, the 1st spindle will be used.



### Relation with other functions

The following functions change after the spindle selection command.

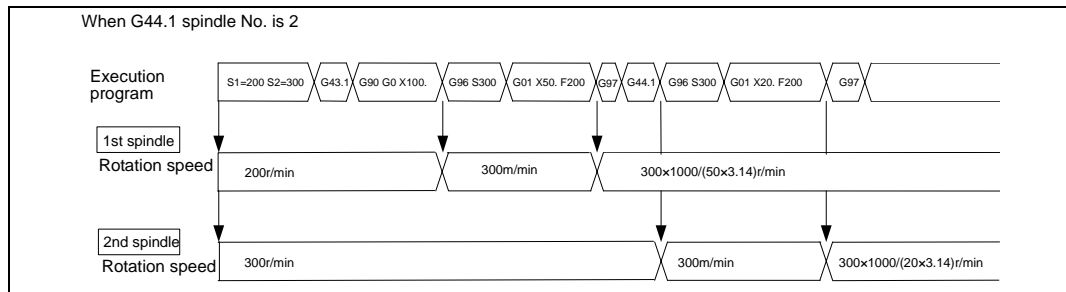
- (1) S command (S\*\*\*\*)  
The S command in G97 (rotation speed command)/G96 (constant surface speed command) will be the command for the spindle designated with the spindle selection command.
- (2) Spindle clamp speed command  
The spindle clamp speed command issued with G92 S\_Q\_ will also follow the spindle selection command mode.
- (3) Per rotation command (synchronous feed)  
Even if F is commanded in the G95 mode, the per rotation feedrate for the spindle designated with the spindle selection command will be applied.
- (4) Constant surface speed control spindle changeover  
The constant surface speed control will also follow the spindle selection command mode.

If spindle other than the current mode is commanded with S $\circ$ =\*\*\*\*, the spindle rotation speed command designated with  $\circ$  will have the priority.

**(Example)**

S2=\*\*\*\* in G43 system mode .....2nd spindle rotation speed command  
 S1=\*\*\*\* in G44 system 2nd spindle selection mode ....1st spindle rotation speed command

Note that the constant surface speed command in the constant surface speed control (G96) will be modal information independent from the rotation speed command.



### 10.6.3 Multiple Spindle Control II



#### Function and purpose

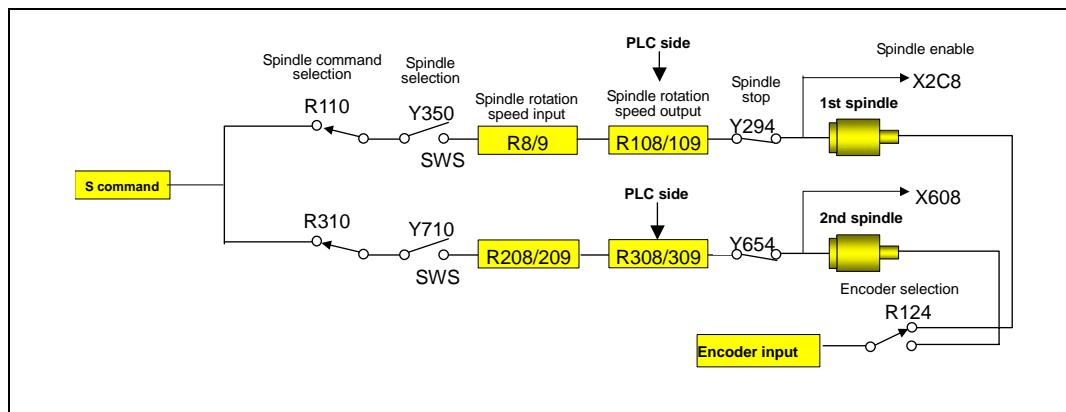
Multiple spindle control II is a function that designates which spindle to select with the signals from PLC. The command is issued to the spindle with one S command.



#### Detailed description

##### (1) Spindle command selection, spindle selection

The S command to the spindle is output as the rotation speed command to the selected spindle when the spindle selection signal (SWS) from the PLC turns ON. The selected spindle rotates at the output rotation speed. The spindle whose selection is canceled when the spindle selection signal (SWS) turns OFF maintains the speed at which it was rotating at before being canceled. This allows each axis to be simultaneously rotated at differing rotation speeds.



**(Note)** Refer to the PLC Interface Manual for details on each signal.



## Relation with other functions

- (1) Spindle clamp speed setting (G92)  
This is valid only on the spindle selected with the spindle selection signal (SWS).  
The spindle not selected with the spindle selection signal (SWS) maintains the speed at which it was rotating at before being canceled.  
(The spindle clamp speed is maintained with the G92 command.)
- (2) Constant surface speed control  
Constant surface speed control can be applied on all spindles.  
The spindle rotation speed is automatically controlled during constant surface speed control, so when machining with constant surface speed, the spindle selection signal (SWS) for that spindle must be left ON.  
The spindle not selected with the spindle selection signal (SWS) maintains the speed at which it was rotating at before being canceled.
- (3) Thread cutting/synchronous feed  
The threads are cut with the spindle selected with the spindle selection signal (SWS). The encoder feedback selected with the encoder selection signal is used.
- (4) Synchronous tap  
The synchronous tap spindle is selected with the spindle selection signal (SWS).  
Select the synchronous tap spindle before issuing the synchronous tap command. Do not change the synchronous tap spindle selection signal during the synchronous tapping mode.  
If a C axis mode command is issued to the synchronous tap spindle, the "M01 operation error 1026" will occur. When the C axis command is canceled, the error will be canceled and machining will resume.
- (5) Asynchronous tap  
The asynchronous tap spindle is selected with the spindle selection signal (SWS).  
Select the asynchronous tap spindle before issuing the tap command. Input a calculation request to change the asynchronous tap spindle selection. Do not change the asynchronous tap spindle selection signal during the asynchronous tapping mode.
- (6) Tap return  
The tap return spindle is selected with the spindle selection signal (SWS).  
Select the spindle for which the tap cycle execution is stopped before turning the tap return signal ON.  
If tap return is executed when a different spindle is selected, the "M01 operation error 1032" will occur. Do not change the spindle selection signal during tap return.



## Restrictions

- (1) The S manual value command is invalid when multiple spindle control II is valid.
- (2) Setup parameter "#1199 Sselect" is invalid when multiple spindle control II is valid.
- (3) The spindle control mode changeover G code cannot be used when multiple spindle control II is valid. A program error (P34) will occur.
- (4) The "S1=\*\*\*" and "S2=\*\*\*" commands are invalid when multiple spindle control II is valid. A program error (P33) will occur.
- (5) The spindle gear shift command output signal (GR1/GR2) is not output when multiple spindle control II is valid.

### 11. Tool Functions

#### 11.1 Tool functions (T8-digits BCD)

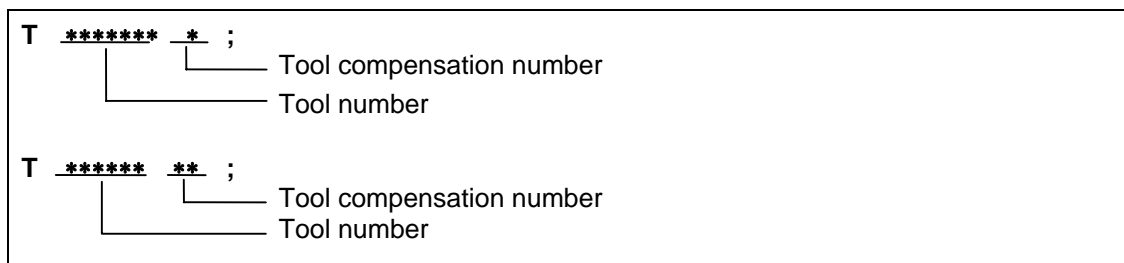


##### Function and purpose

Tool functions are also known as T functions, and are used to designate the tool Nos. and tool compensation Nos. The designations are made by the 8 digits (0 to 99999999) following the address T. These commands are used with the higher-order six or seven digits indicating the tool No., and the lower-order one or two digits indicating the compensation No. Which is to be used is determined by setting in the parameter "#1098 TIno.". The usable T commands differ according to each machine, so refer to the instruction manual issued by the machine tool builder. One set of T commands can be issued in one block.



##### Command format



Reference should be made to the instructions issued by the machine tool builder for the correspondence between the actual tools and the tool numbers commanded in the program.

BCD codes and start signals are output.

If the T function is designated in the same block as a movement command, the commands may be executed in either of the following two orders. The machine specifications determine which sequence applies.

- (a) The T function is executed after the movement command.
- (b) The T function is executed simultaneously with the movement command.

Processing and completion sequences are required for all T commands.



### 12. Tool Offset Functions

#### 12.1 Tool offset



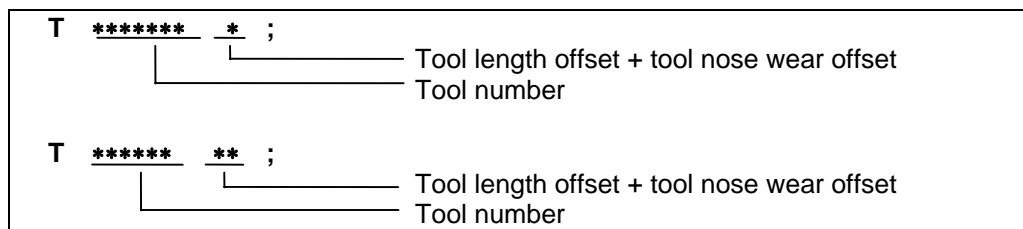
##### Function and purpose

Tool offset is performed by the T functions which are commanded with the 3-, 4- or 8-digit number following address T. There are two types of tool offset: tool length offset and tool nose wear offset. There are two ways to issue the commands: the tool length offset and tool nose wear offset are designated by the last 1 or 2 digits of the T command or the tool nose wear offset is assigned by the last 1 or 2 digits of the T command and the tool length offset by the tool number. Parameter "#1098 TLno." is used to switch between them. Parameter "#1097 T1digit" is also used to select the last 1 or 2 digits for the offset. One group of T command can be assigned in a block.

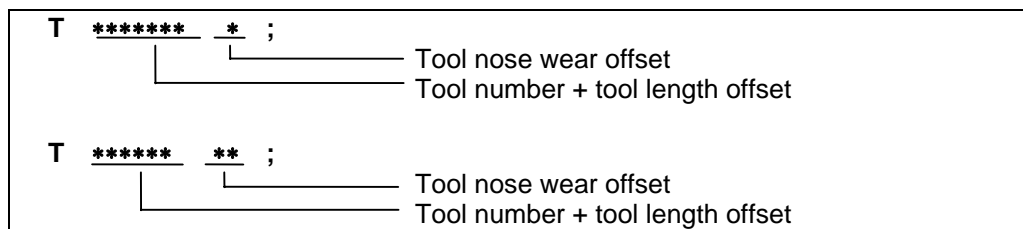


##### Command format

- (1) When designating the tool length and tool nose wear offset number using the last 1 or 2 digits of the T command.



- (2) When differentiating between the tool length offset number and tool nose wear offset number



The lower two digits of the tool No. are the tool length offset No.

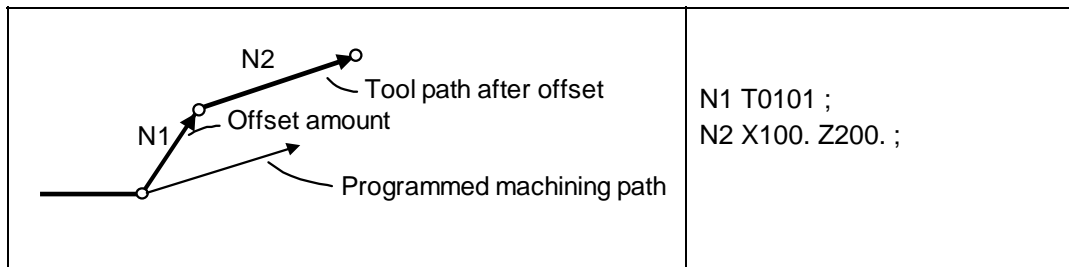
### 12.1.1 Tool offset start



#### Detailed description

There are two ways to execute tool offset and these can be selected by parameter: executing offset when the T command is executed and executing offset in the block with a movement command without performing offset when the T command is executed.

#### (1) Offset with T command execution



Tool length offset and tool nose wear offset are conducted simultaneously.

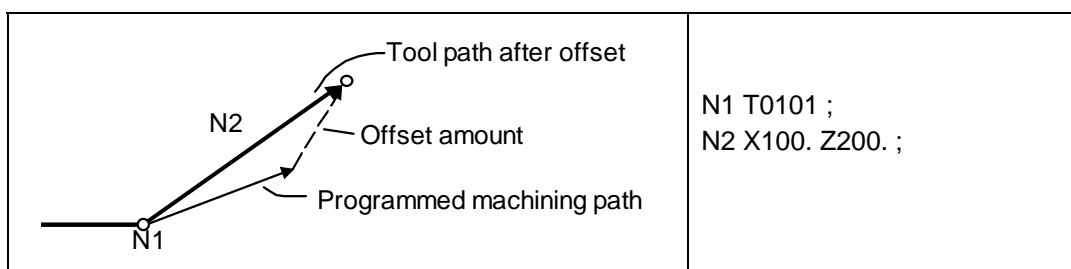
**(Note 1)** The movement applying to offset with the T command execution is rapid traverse in a G00 modal and cutting feed with other modals.

**(Note 2)** When performing offset with T command execution, the path is offset as a linear movement in a circular modal.

**(Note 3)** When performing offset with T command execution, offset will not function until the arrival of any G command except those listed below when the T command has been assigned in the same block as the G commands listed below.

- G04 : Dwell
- G10 : Program tool offset input/program parameter input
- G11 : Program parameter input mode cancel
- G65 : User macro simple call
- G92 : Coordinate system setting

#### (2) Offset with movement command



Tool length offset and tool nose wear offset are conducted simultaneously.

**(Note 1)** When performing offset with a movement command, offset is applied if the offset amount is lower than the parameter "#1084 RadErr" when offset is performed for the first time with a circular command. If the amount is higher, the program error (P70) results. (This also applies when the arc command and T command are in the same block for offsetting with T command execution.)

#### 12.1.2 Expanded method at starting tool offset



##### Function and purpose

By setting the parameter "#1100 Tmove", the offset operation when T is commanded is selected whether that is carried out when the T command is executed or carried out with superimposed on the movement command.

In addition to these two methods, a type to compensate only the wear amount when the T command is executed and to compensate the tool length superimposed on the movement command can be selected with the parameter.



##### Detailed description

To select the type to compensate only the wear amount when the T command is executed and to compensate the tool length superimposed on the movement command, set the "#1100 Tmove" to 2.

When 2 is set, the length offset will follow setting 1 (superimposed on movement command), and the wear offset will follow setting 0 (move when T is commanded).

The offset operation for each "#1100 Tmove" setting is as shown below.

When Length offset amount: 7.000mm Wear offset amount: 0.500mm

Machining program example	Machine value		
	#1100 Tmove = 0	#1100 Tmove = 1	#1100 Tmove = 2
G28 X. ;	0.000	0.000	0.000
G0 T116. ;	7.500	0.000	0.500
G0 X100. ;	107.500	107.500	107.500
:			

**(Note 1)** If the following G commands are issued in the same block as the T command, the wear will not be compensated in that block.

Same block compensation standby: G04, G10, G11, G65, G92

The wear amount will be compensated in a block where other G commands are issued. (All axes)

**(Note 2)** After the offset amount is temporarily canceled, the offset amount will return for each movement command axis. The movement at this time will follow parameter "#1101 Tabsmv" for both length offset and wear offset.

G command involving temporary cancel: G28, G30, G53

**(1) Relation with "#1101 Tabsmv" (tool offset amount superimposition during absolute command)**

When "#1100 Tmove" is set to 2, the offset operation will follow the "#1101 Tabsmv" setting.

#### (2) Parameters

#	Items	Details	Setting range
1100	Tmove	<p>Designate the timing to perform tool length offset and wear offset.</p> <ul style="list-style-type: none"> <li>0: Compensates when T command is executed.</li> <li>1: Superimposes and compensates with the movement command in the block where the T command is located. If there is no movement command in the same block, compensation will be executed after the movement command is superimposed in the next movement command block.</li> <li>2: Compensates wear amount when the T command is executed. Superimposes and compensates a tool length offset amount with the movement command in the same block. If there is no movement command in the same block, compensation will be executed after the movement command is superimposed in the next movement command block.</li> </ul>	0 to 2
1101	Tabsmv	<p>Designate the type of movement command superimposed when #1100 Tmove is set to 1 or 2.</p> <ul style="list-style-type: none"> <li>0: Compensates regardless of the movement command type.</li> <li>1: Compensates only at the movement command in the absolute value command.</li> </ul>	0,1

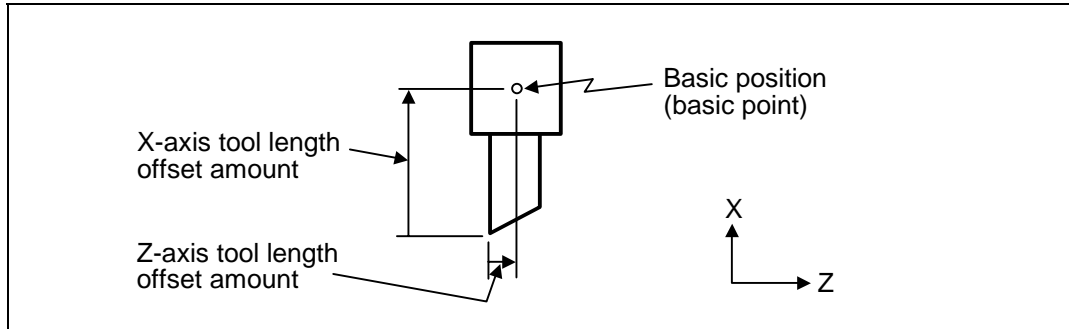
### 12.2 Tool length offset



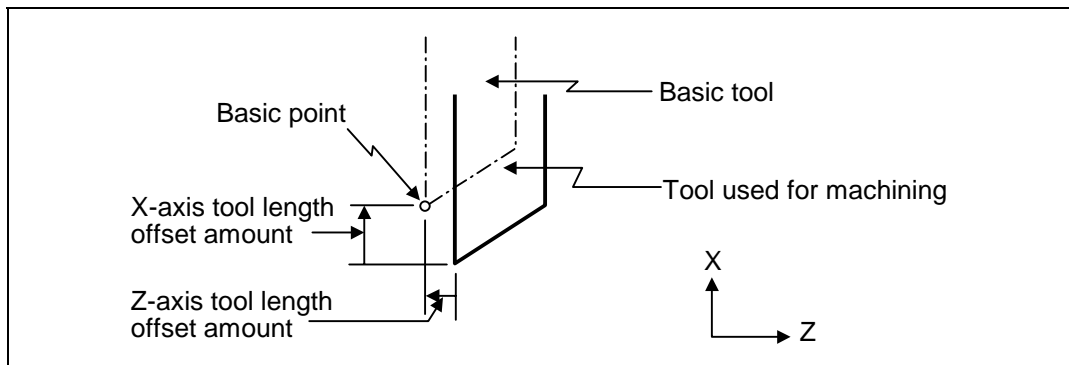
#### Tool length offset amount setting

This function offsets the tool length with respect to the programmed basic position. This position may generally be set to either the center position of the turret or the tool nose position of the basic tool.

#### (1) Center position of turret

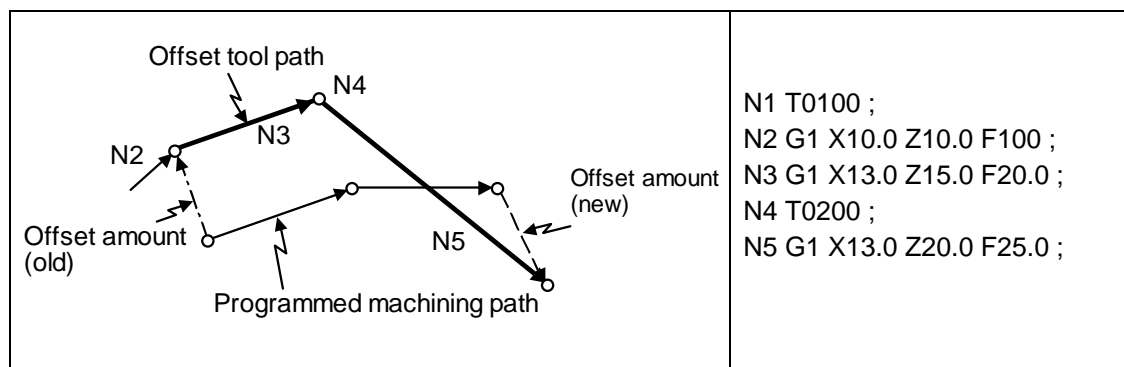


#### (2) Tool nose position of basic tool



#### Tool length offset number change

When tool numbers are changed, the tool length offset corresponding to the new tool numbers are added to the movement amounts in the machining program.



In this example, the tool length is offset with the tool number and offset is performed in the block with the movement command.



### Tool length offset cancel

**(1) When an offset number of 0 has been assigned**

Offset is canceled when 0 has been assigned as the tool length offset number by the T command.

	<pre>N1 X10.0 Z10.0 F10 ; N2 T0000 ; N3 G1 X10.0 Z20.0 ;</pre> <p>In this case, offset is performed by the block with the movement command.</p>
--	---

**(2) When the assigned offset amount is 0**

Offset is canceled when the offset amount in the tool length offset number assigned by the T command is 0.

	<pre>N1 G1 X10.0 Z10.0 F10 ; N2 T0100 ; N3 G1 X10.0 Z20.0 ;</pre> <p>In this case, offset is performed by the block with the movement command.</p>
--	--

**(Note 1)** When G28, G29 or G30 is commanded, the machine moves to the position where the offset was canceled and the offset amount is stored in the memory. This means that with the next movement command the machine will move to the offset position.

**(Note 2)** Even if the offset amount of the offset number currently selected by MDI is changed during automatic operation, the changed offset amount will not be valid unless a T command with the same number is executed again.

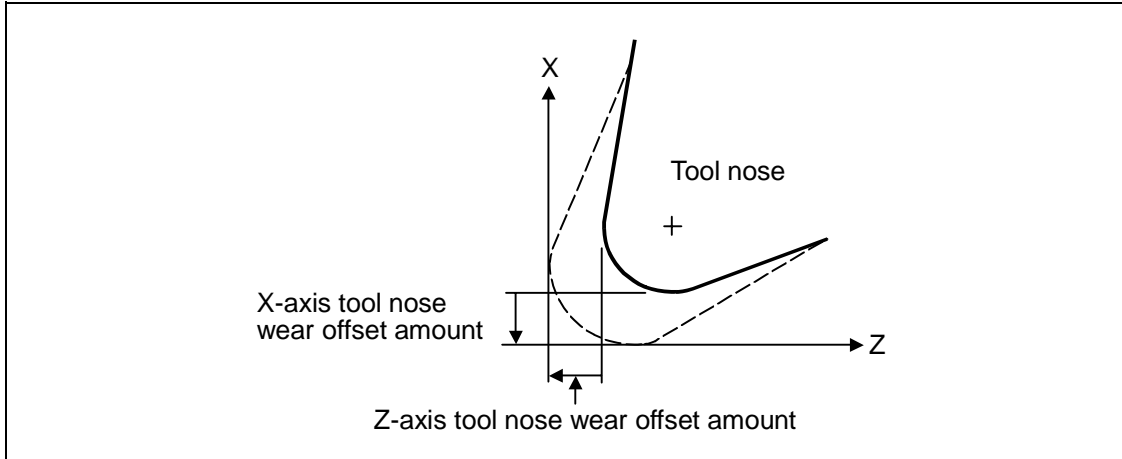
**(Note 3)** The tool length offset and tool nose wear offset amounts are cleared by resetting and by emergency stop. They can be retained by parameter "#1099 Treset".

#### 12.3 Tool nose wear offset



##### Tool nose wear offset amount setting

The wear sustained by the tool being used can be offset.



##### Tool nose wear offset cancel

Tool nose wear offset is canceled when 0 has been assigned as the offset number.

	<pre>N1 G1 X10.0 Z10.0 F10 ; N2 T0100 ; N3 G1 X10.0 Z20.0 ;</pre> <p>In this case, offset is performed in the block with the movement command.</p>
--	--

- (Note 1)** When G28, G29 or G30 is commanded, the machine moves to the position where the offset was canceled and the offset amount is stored in the memory. This means that with the next movement command the machine will move to the offset position.
- (Note 2)** Even if the offset amount of the offset number currently selected by MDI is changed during automatic operation, the changed offset amount will not be valid unless a T command with the same number is executed again.
- (Note 3)** The tool length offset and tool nose wear offset amounts are cleared by resetting and by emergency stop. They can be retained by parameter "#1099 Treset".

## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

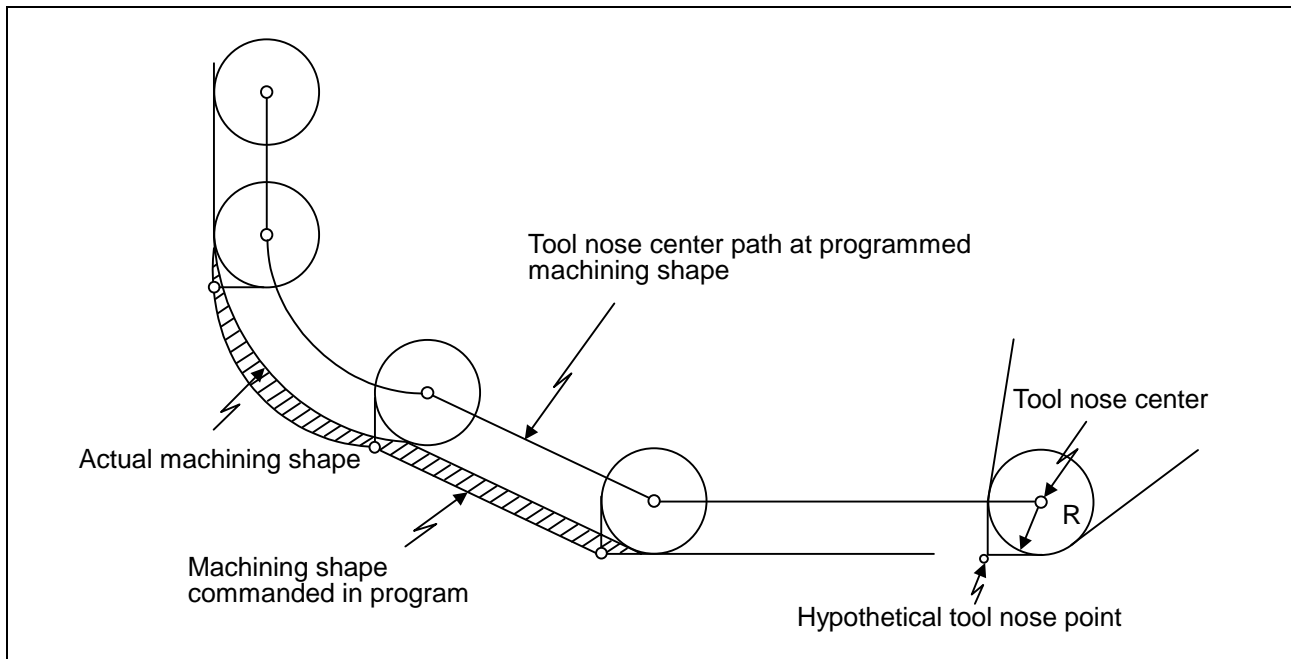
#### 12.4 Tool nose radius compensation; G40, G41, G42, G46



##### Function and purpose

The tool nose is generally rounded and so a hypothetical tool nose point is treated as the tool nose for programming purposes. When this is done, an error caused by the tool nose rounding arises during taper cutting or circular cutting between the actually programmed shape and the cutting shape. Tool nose radius compensation is a function for automatically calculating and offsetting this error by setting the tool nose radius value.

The command codes enable the offset direction to be fixed or automatically identified.



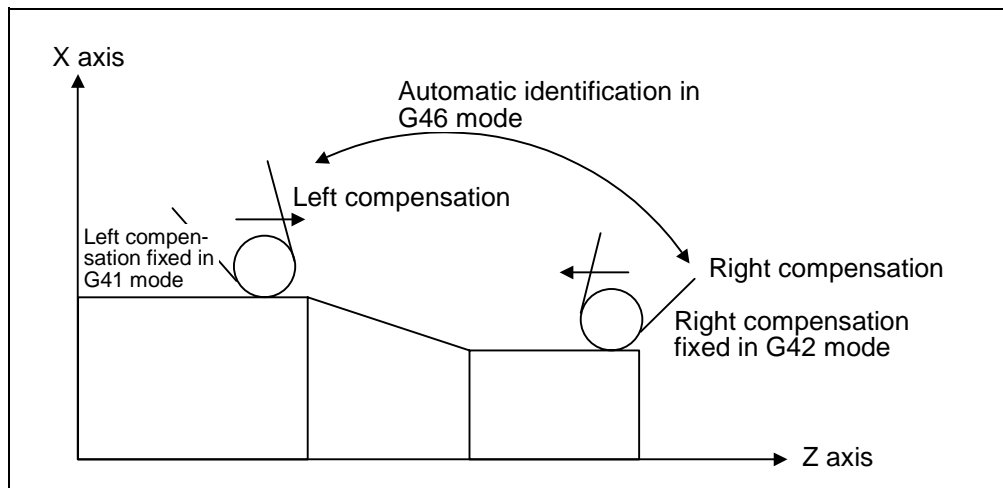
##### Functions and command formats

Code	Function	Command format
G40	Tool nose radius compensation mode cancel	G40 (Xx/Uu Zz/Ww li Kk) ;
G41	Tool nose radius compensation left mode ON	G41 (Xx/Uu Zz/Ww) ;
G42	Tool nose radius compensation right mode ON	G42 (Xx/Uu Zz/Ww) ;
G46	Tool nose radius compensation automatic direction identification mode ON	G46 (Xx/Uu Zz/Ww) ;

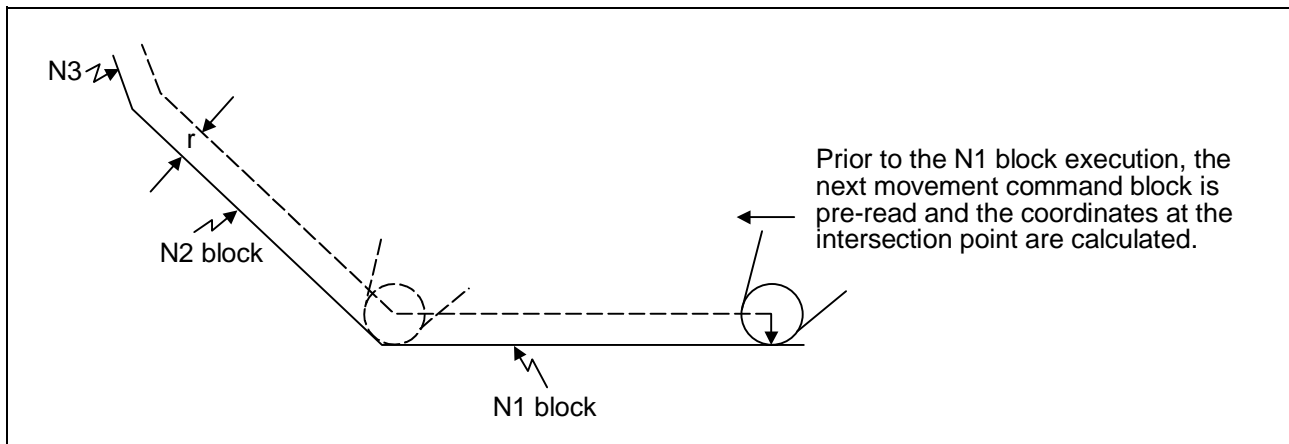
**(Note 1)** By means of the preset hypothetical tool nose point and movement commands in the machining program, the G46 tool nose radius compensation function automatically identifies the compensation direction and provides tool nose radius compensation.

**(Note 2)** G40 serves to cancel the tool nose radius compensation mode.





**(Note 3)** Tool nose radius compensation pre-reads the data in the following two movement command blocks (up to 5 blocks when there are no movement commands) and controls the tool nose radius center path by the intersection point calculation method so that it is offset from the programmed path by an amount equivalent to the nose radius.



**(Note 4)** In the above figure, "r" is the tool nose radius compensation amount (nose radius).

**(Note 5)** The tool nose radius compensation amount corresponds to the tool length number and it is preset along with the tool nose point.

**(Note 6)** If there are 4 or more blocks without movement amounts among 5 continuous blocks, overcutting or undercutting will result. Blocks in which optional block skip is valid are ignored.

**(Note 7)** Tool nose radius compensation is also valid for fixed cycles (G77 ~ G79) and for rough cutting cycles (G70, G71, G72, G73).

However, in the rough cutting cycles, the finished shape with the tool nose radius compensation applied will be cut with the compensation canceled and, upon completion of the cutting, operation will automatically return to the compensation mode.

**(Note 8)** With thread cutting commands, compensation is temporarily canceled 1 block before.

**(Note 9)** A tool nose radius compensation (G41 or G42) command can be assigned during tool nose radius compensation (G46). There is no need to cancel the G40 compensation.

**(Note 10)** The compensation plane, movement axes and next advance direction vector follow the plane selection command designated by G17, G18 or G19.

G17	X-Y plane X, Y, I, J
G18	Z-X plane Z, X, K, I
G19	Y-Z plane Y, Z, J, K

## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

#### 12.4.1 Tool nose point and compensation directions

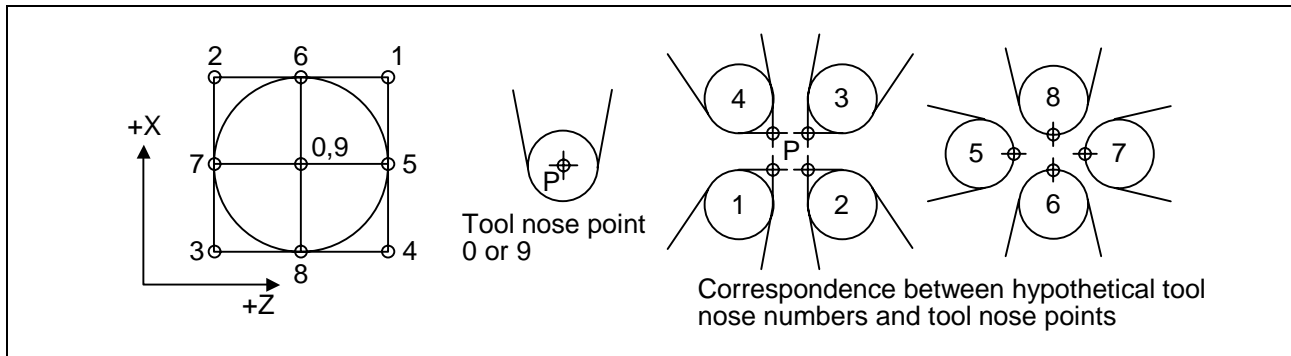


##### Tool nose point

Since the tool nose is generally rounded, the programmed tool nose position is aligned with point P shown in the examples of the figures below.

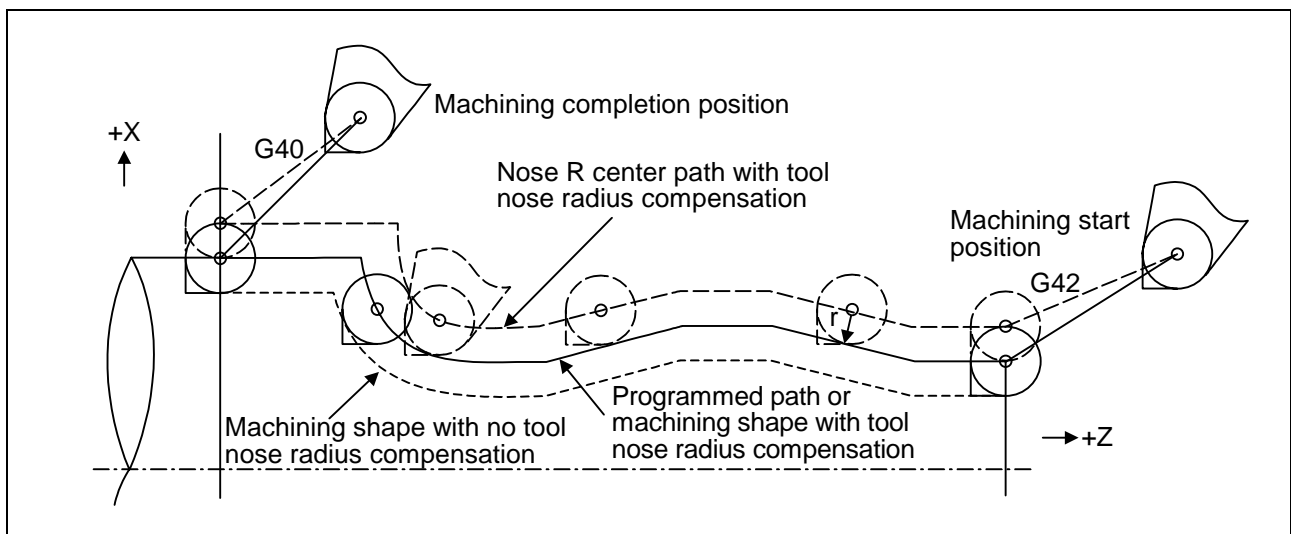
With tool nose radius compensation, one point among those in the figures below indicating the position relationship is selected for each tool length number and preset.

(Selection from points 1 to 8 in the G46 mode and 0 to 9 in the G41/G42 mode.)

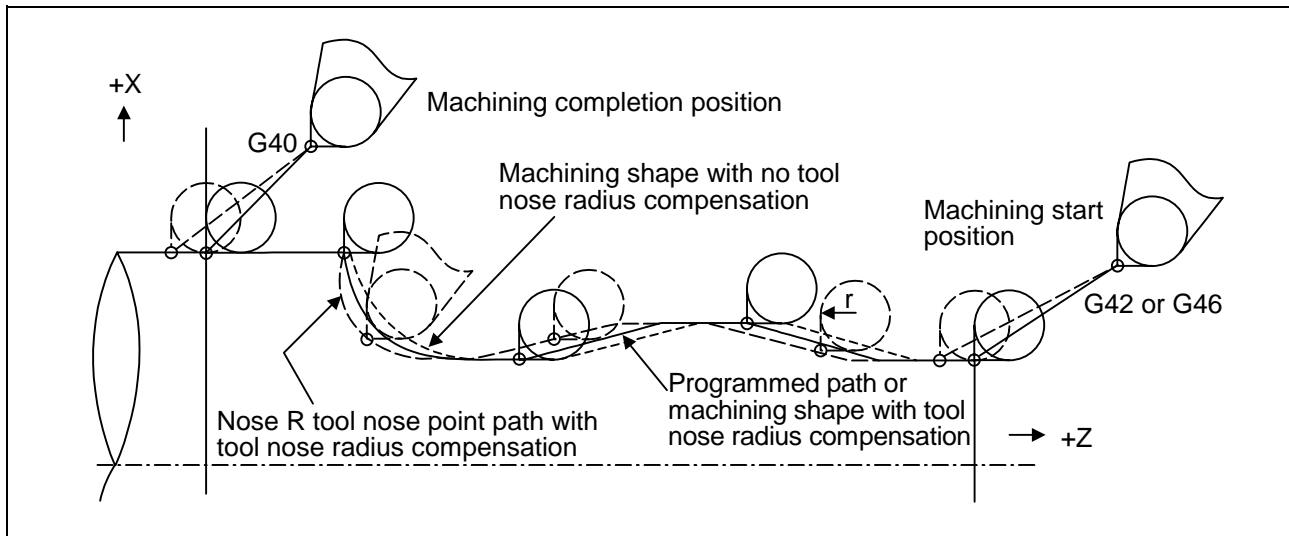


##### Tool nose point and compensation operation

(1) When the nose R center has been aligned with the machining start position



(2) When the tool nose point has been aligned with the machining start position



#### Compensation directions

- (1) The compensation direction of the G41/G42 commands is determined by the G41/G42 codes. The direction in a G46 command is automatically determined in accordance with the following table from the relationship between the tool nose points and the commanded movement vectors.
- (2) When tool nose radius compensation has been started and the initial movement vector (including G0) corresponds to an "x" mark in the table, the compensation direction cannot be specified and so it is determined by the next movement vector. When the direction cannot be determined even after reading 5 blocks ahead, program error (P156) results.
- (3) When an attempt is made to reverse the compensation direction during tool nose radius compensation, program error (P157) results except when the reversal is done in the G00 block. Even if the directions differ before and after the G28, G30 or G53 block, an error will not result since compensation is temporarily canceled. Using a parameter (#8106 G46 reverse axis error evade), the tool can also be moved unchanged in the same compensation direction.

## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

- (4) When the compensation direction during tool nose radius compensation coincides with an "x" in the table below, the direction complied with the previous compensation direction. Determining the compensation direction by the movement vectors and tool nose point in command G46

Compensation direction of tool nose		Tool nose points								Compensation direction of tool nose	
		1	2	3	4	5	6	7	8		
Direction of tool nose advance		Right	Right	Left	Left	X	Right	X	Left	→	
		X	Right	X	Left	Left	Right	Right	Left		
		Left	Right	Right	Left	Left	X	Right	X	↑	
		Left	X	Right	X	Left	Left	Right	Right		
		Left	Left	Right	Right	X	Left	X	Right	←	
		X	Left	X	Right	Right	Left	Left	Right		
		Right	Left	Left	Right	Right	X	Left	X	↓	
		Right	X	Left	X	Right	Right	Left	Left		
Movement vector (tool nose points 1 to 4)		Right	Right	Left	Left	X	Right	X	Left	Movement vector (tool nose points 5 to 8)	
		X	Right	X	Left	Left	Right	Right	Left		
		Left	Right	Right	Left	Left	X	Right	X		
		Left	X	Right	X	Left	Left	Right	Right		
		Left	Left	Right	Right	X	Left	X	Right		
		X	Left	X	Right	Right	Left	Left	Right		
		Right	Left	Left	Right	Right	X	Left	X		
		Right	X	Left	X	Right	Right	Left	Left		

**(Note 1)** An "x" mark in the tables indicates that the compensation direction is not determined from the movement vector (tool nose points).

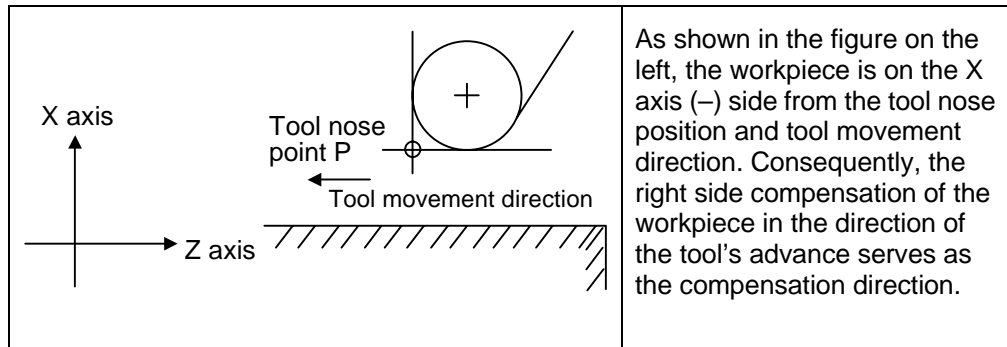
**(Note 2)** The "" mark denotes a movement vector in the 45° direction. (The other movement vectors are based on this.)

**(Note 3)** The "" mark denotes a movement vector with a range larger than 45° and smaller than 135°. (The other movement vectors are based on this.)

## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

**(Example)** With tool nose point 3, movement vector in the Z axis (-) direction  
(with ← movement vector)



#### 12.4.2 Tool nose radius compensation operations



##### Tool nose radius compensation cancel mode

The tool nose radius compensation cancel mode is established by any of the following conditions.

- (1) After the power has been turned ON
- (2) After the reset button on the setting and display unit has been pressed
- (3) After the M02 or M30 command with reset function has been executed
- (4) After the compensation cancel command (G40) has been executed
- (5) After tool number 0 has been selected (T00 has been executed)

The offset vectors are zero in the compensation cancel mode, and the tool nose point path coincides with the program path.

Programs including tool nose radius compensation must be terminated in the compensation cancel mode.



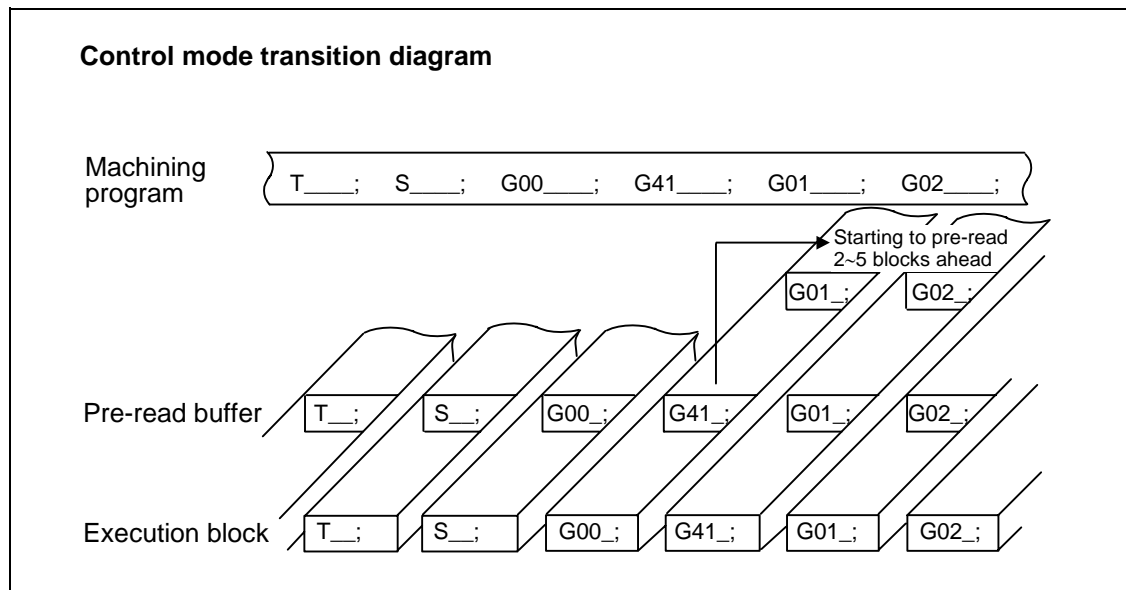
#### Tool nose radius compensation start (start-up)

Tool nose radius compensation starts when all the following conditions are met in the compensation cancel mode.

- (1) The G41, G42 or G46 command has been issued.
- (2) The movement command is any command except a circular command.

At the start of compensation, the operation is executed after at least 2 to 5 blocks have been read continuously for intersection point calculation regardless of single block operation (Two blocks are pre-read if movement commands are present; up to 5 blocks are pre-read if such commands are not present.)

During compensation mode, up to 5 blocks are pre-read and the compensation is arithmetically processed.



There are two ways of starting the compensation operation: type A and type B. The type to be used is selected by setting bit2 of parameter "#1229 set 01/bit2". This type is used in common with the compensation cancel type. In the following explanatory figure, "S" denotes the single block stop point.

Type A	When the start-up and cancel commands are operated during nose R and radius compensation, their blocks are not handled by intersection operation processing; they are handled as offset vectors in the direction vertical to that of the commands.
Type B	When the start-up and cancel commands are operated during nose R and radius compensation, the intersection operation processing of the command block and the next block is executed.

In the following explanatory figure, "S" denotes the single block stop point.

## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation



#### Start operation for tool nose radius compensation

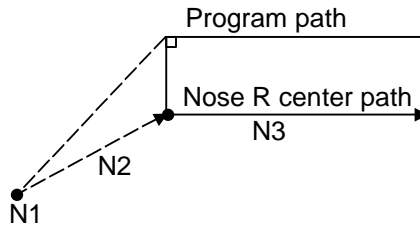
With a G41, G42 or G46 command alone, the tool does not move as much as the specified amount of tool nose radius compensation. It does not start tool nose radius compensation with a G00 command. It starts tool nose radius compensation normally with a G01, G02 or G03 command.

Note that even if there is an axis command, the tool nose radius compensation will not be applied unless there is movement.

(1) When an independent G41/G42/G46 command is issued at an inside corner

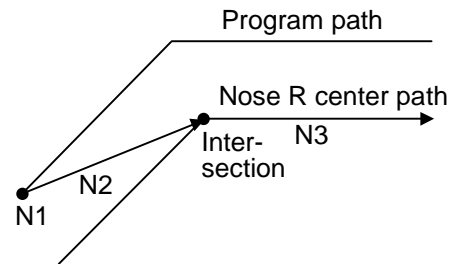
```

}
N1 G42 ;
N2 G00 X_Z_ ;
N3 G01 X_Z_F_ ;
}
    
```



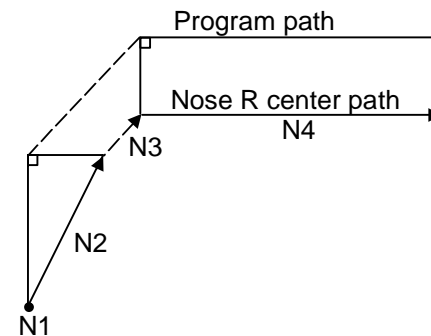
```

}
N1 G42 ;
N2 G01 X_Z_F_ ;
N3 G01 X_Z_ ;
}
    
```



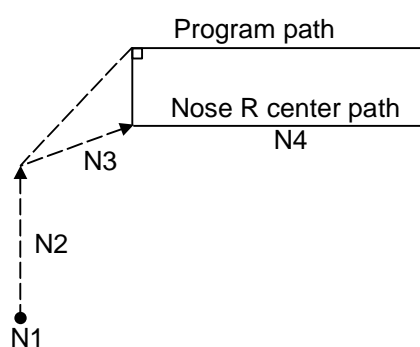
```

}
N1 G42 ;
N2 G01 X_Z_F_ ;
N3 G00 X_Z_ ;
N4 G01 X_Z_ ;
}
    
```



```

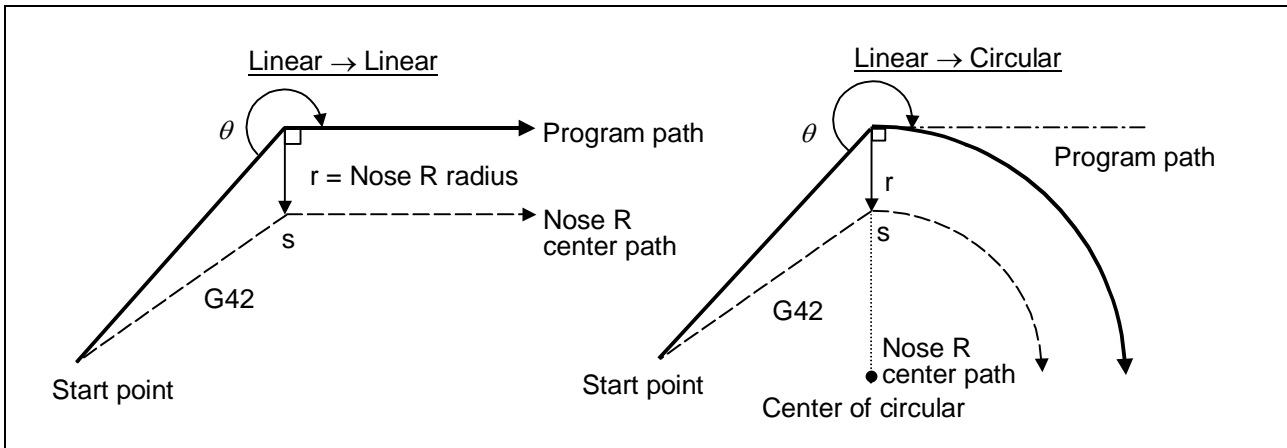
}
N1 G42 ;
N2 G00 X_Z_ ;
N3 G00 X_Z_ ;
N4 G01 X_Z_F_ ;
}
    
```



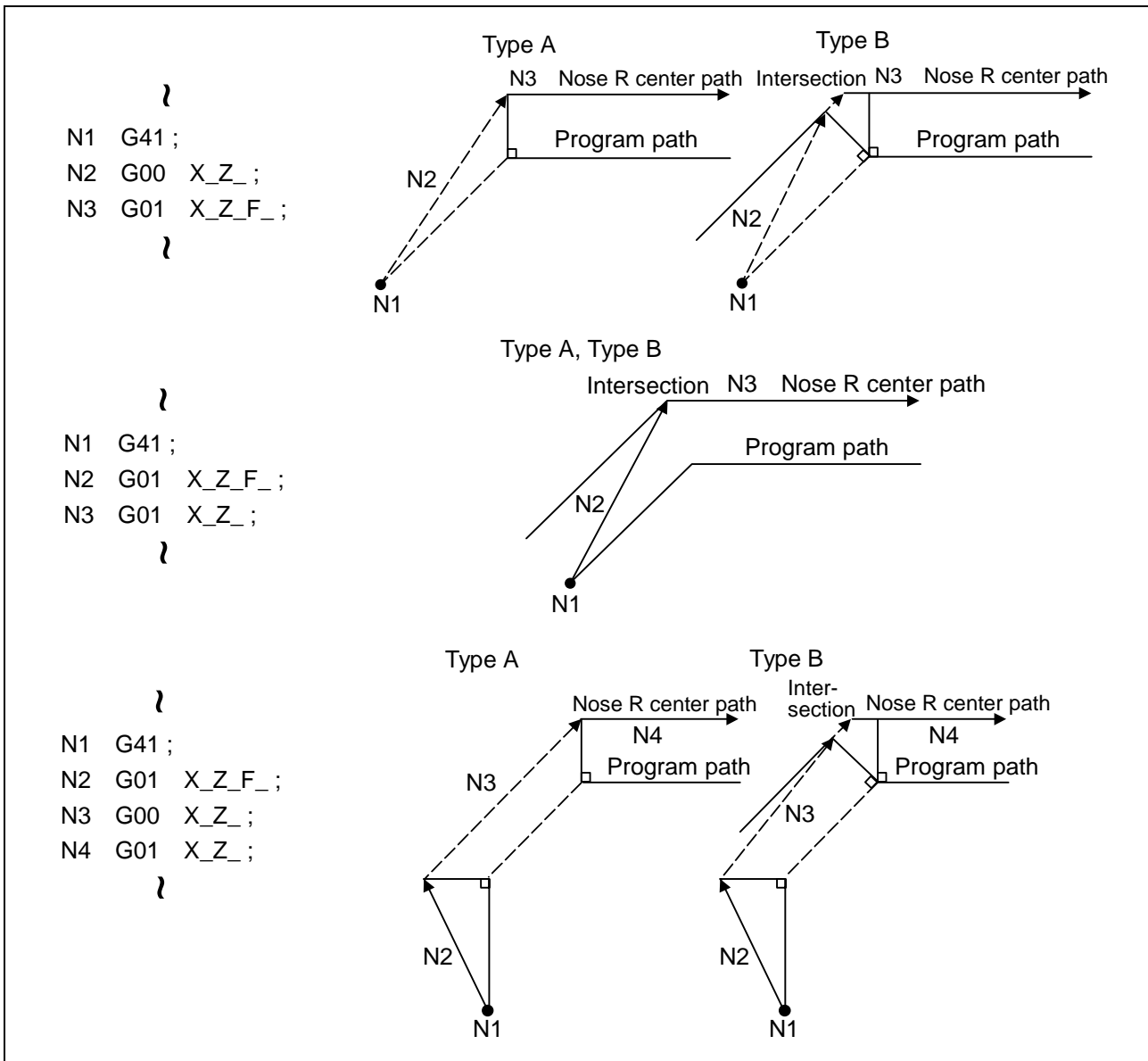
## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

- (2) When a G41/G42/G46 command is issued at an inside corner, in the same block as a movement command



- (3) When an independent G41/G42/G46 command is issued at an outside corner (obtuse angle)





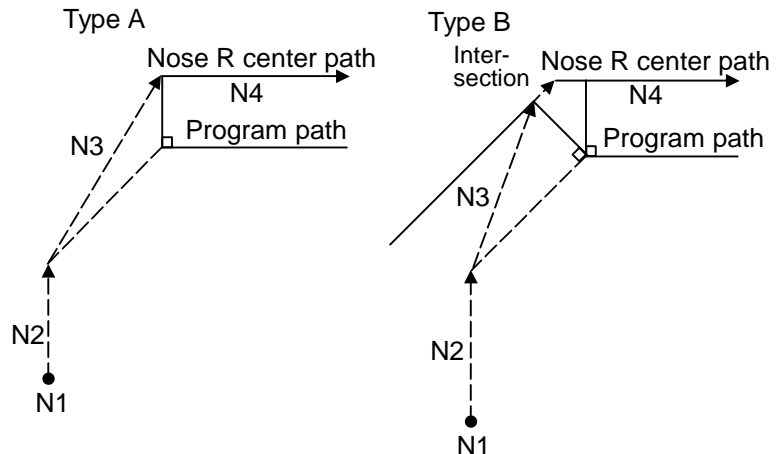
## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

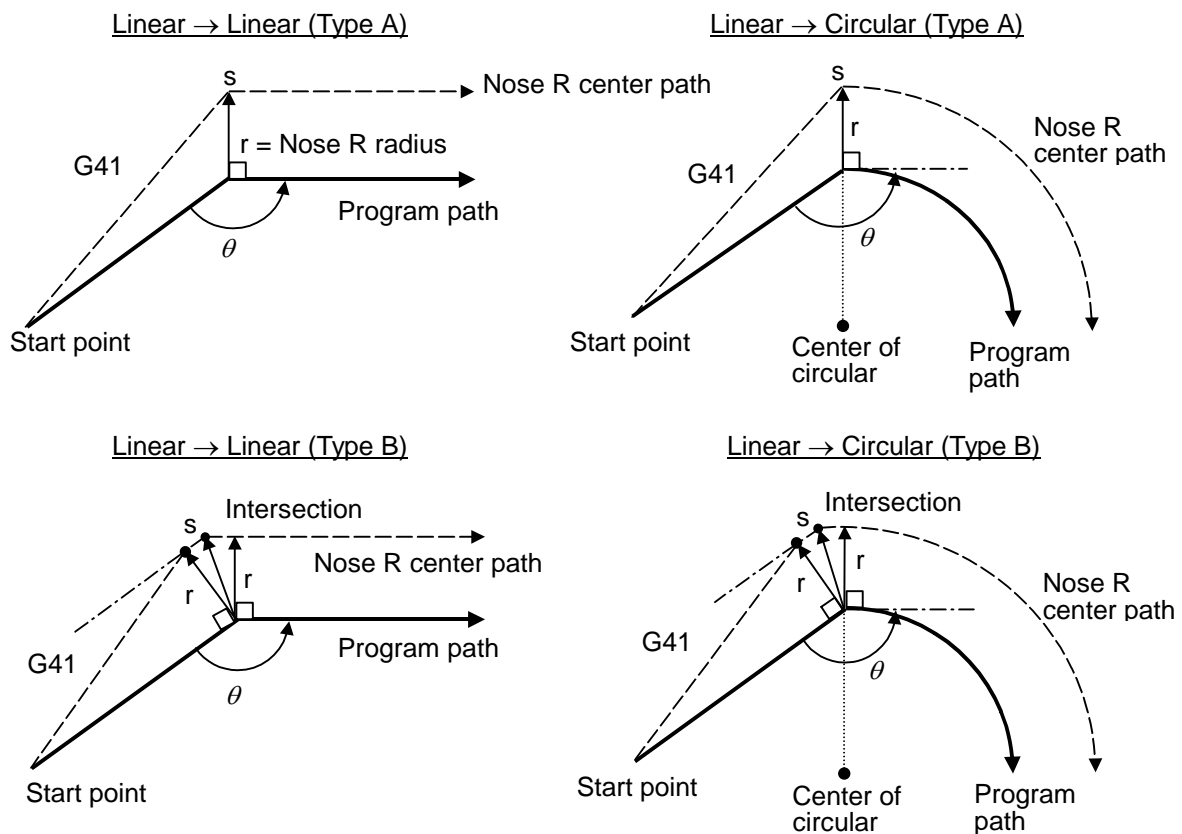
```

}
N1 G41 ;
N2 G00 X_Z_ ;
N3 G00 X_Z_ ;
N4 G01 X_Z_F_ ;
}

```



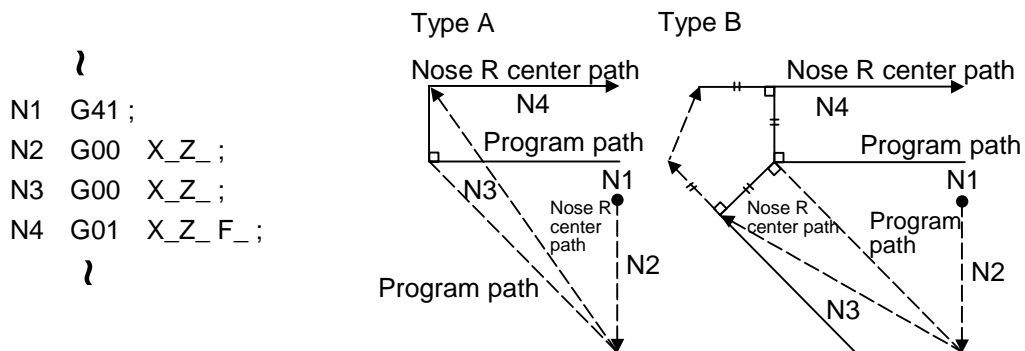
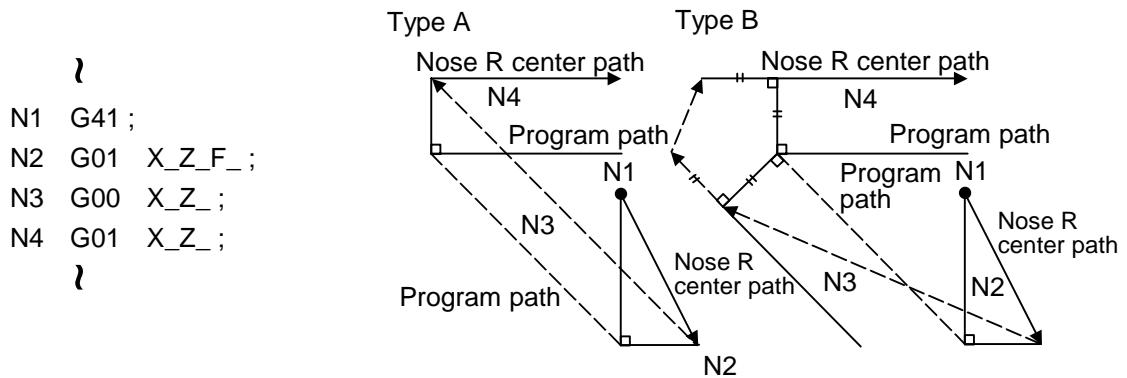
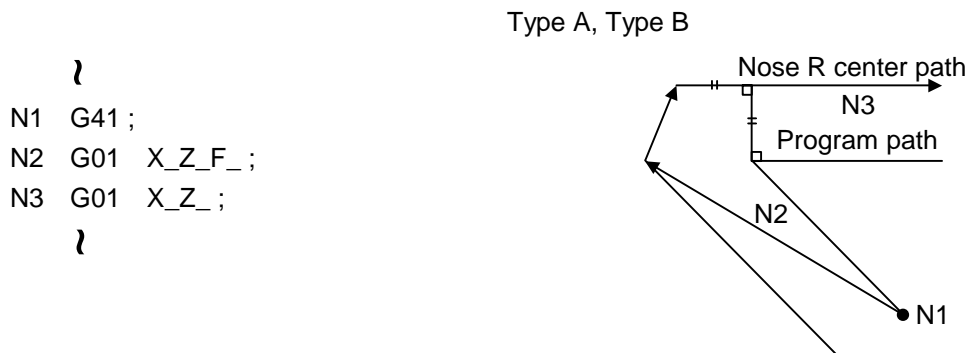
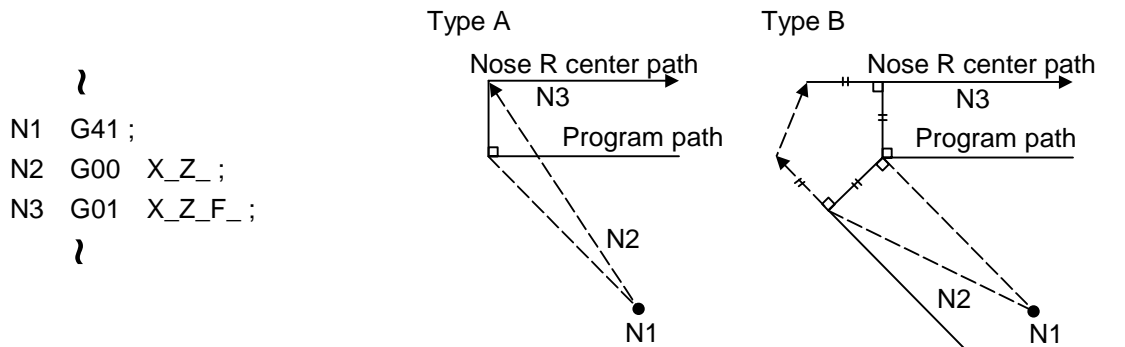
- (4) When a G41/G42/G46 command is issued at an outside corner (obtuse angle), in the same block as a movement command [ $90^\circ \leq \theta < 180^\circ$ ]



## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

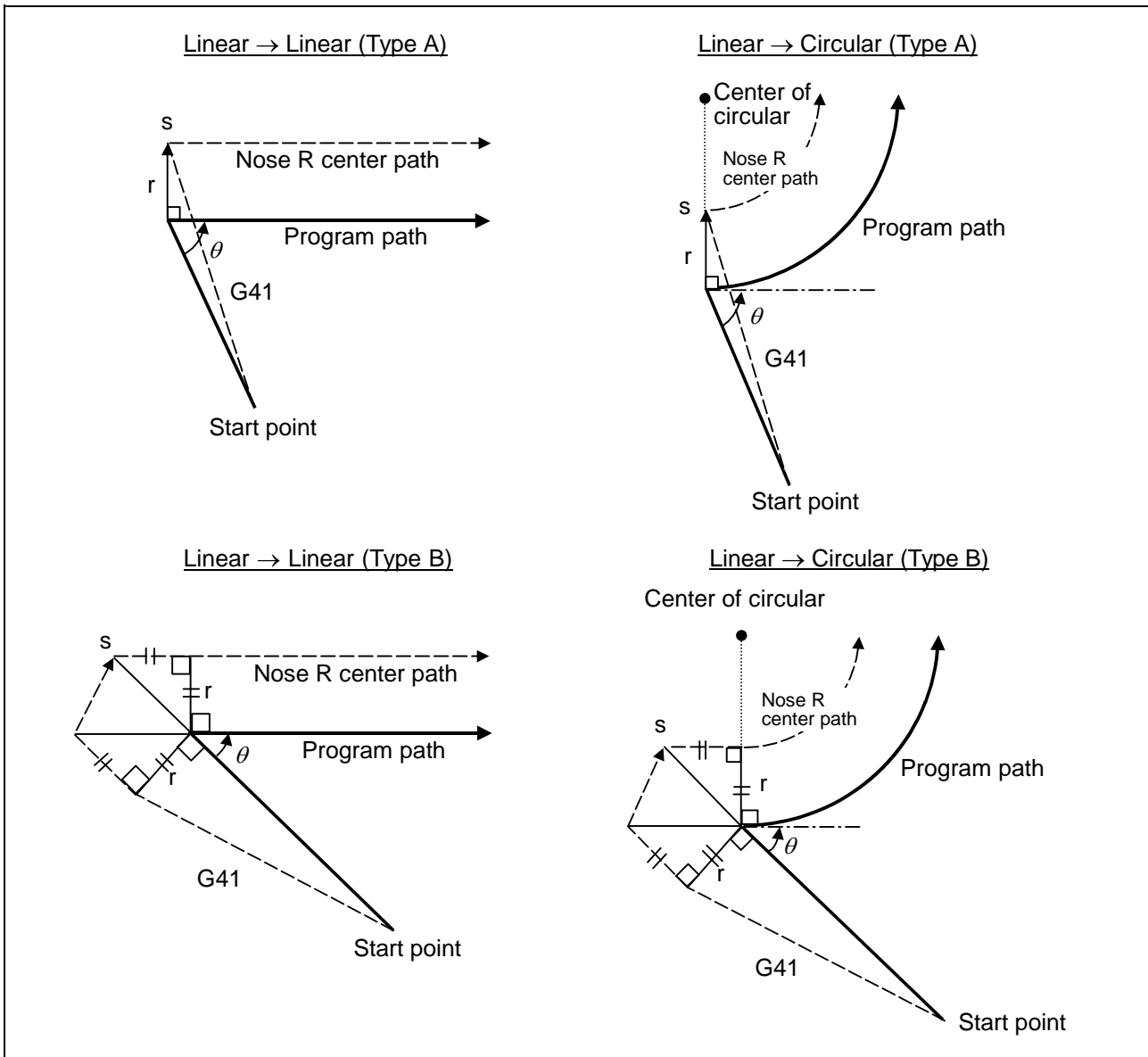
(5) When an independent G41/G42/G46 command is issued at an outside corner (acute angle)



## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

- (6) When a G41/G42/G46 command is issued at an outside corner (acute angle), in the same block as a movement command [  $\theta < 90^\circ$  ]



**(Note 1)** Where is no axis movement command in the same block, compensation is performed perpendicularly to the next block direction.

## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation



#### Operations in compensation mode

The tool center path is obtained and compensated from the linear and arc designations for the program path (G00, G01, G02, G03).

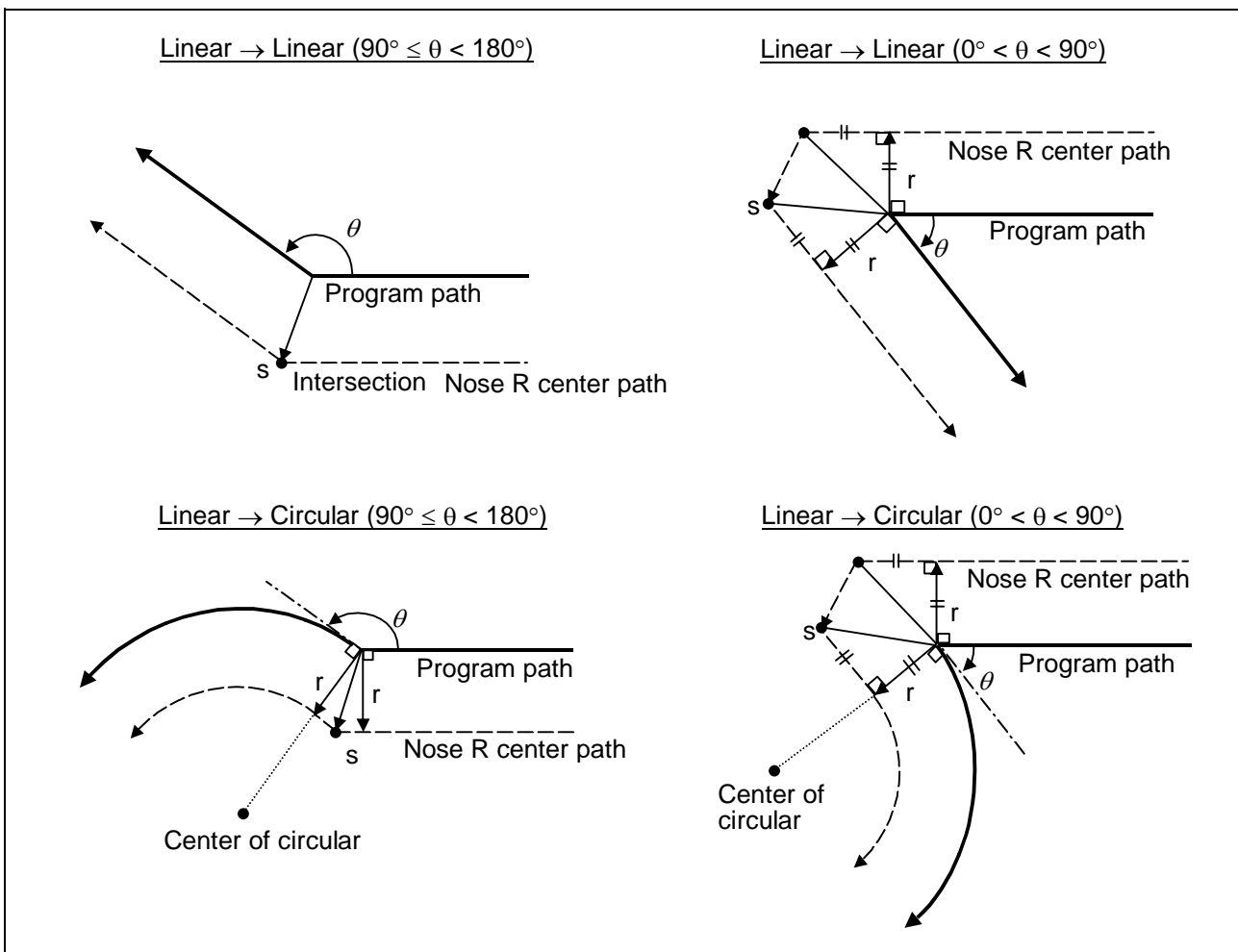
Compensation is valid both for positioning and for interpolation commands such as circular and linear interpolation.

Even if the same compensation command (G41, G42, G46) is issued in a tool nose radius compensation (G41, G42, G46) mode, the command will be ignored.

When 4 or more blocks not accompanying movement are commanded continuously in the compensation mode, overcutting or undercutting will result.

When the M00 command has been issued during tool nose radius compensation, pre-reading is prohibited.

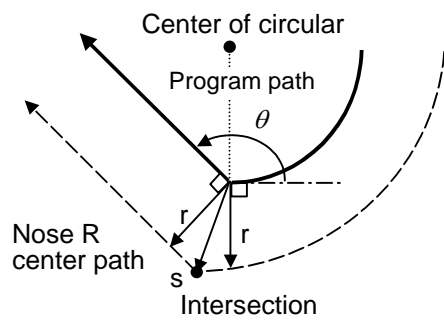
#### (1) Machining an outside corner



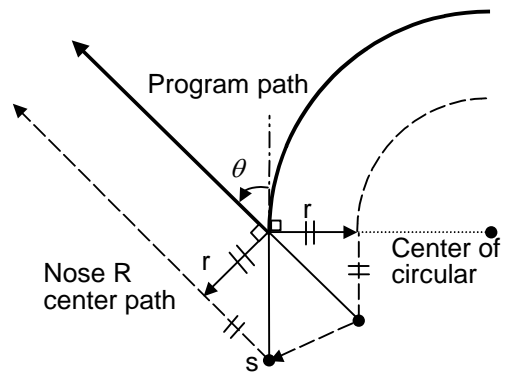
## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

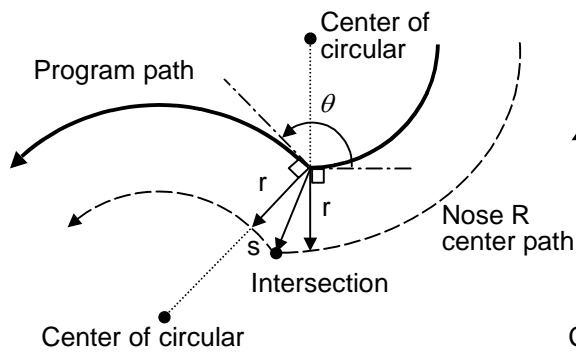
Circular  $\rightarrow$  Linear ( $90^\circ \leq \theta < 180^\circ$ )



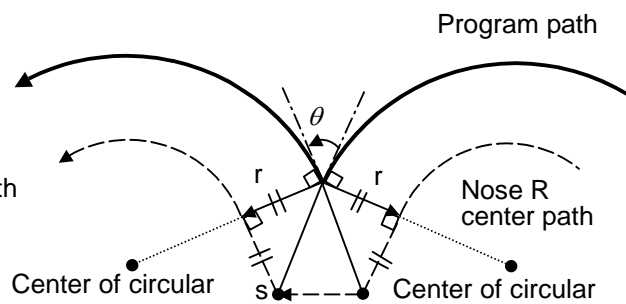
Circular  $\rightarrow$  Linear ( $0^\circ < \theta < 90^\circ$ )



Circular  $\rightarrow$  Circular ( $90^\circ \leq \theta < 180^\circ$ )



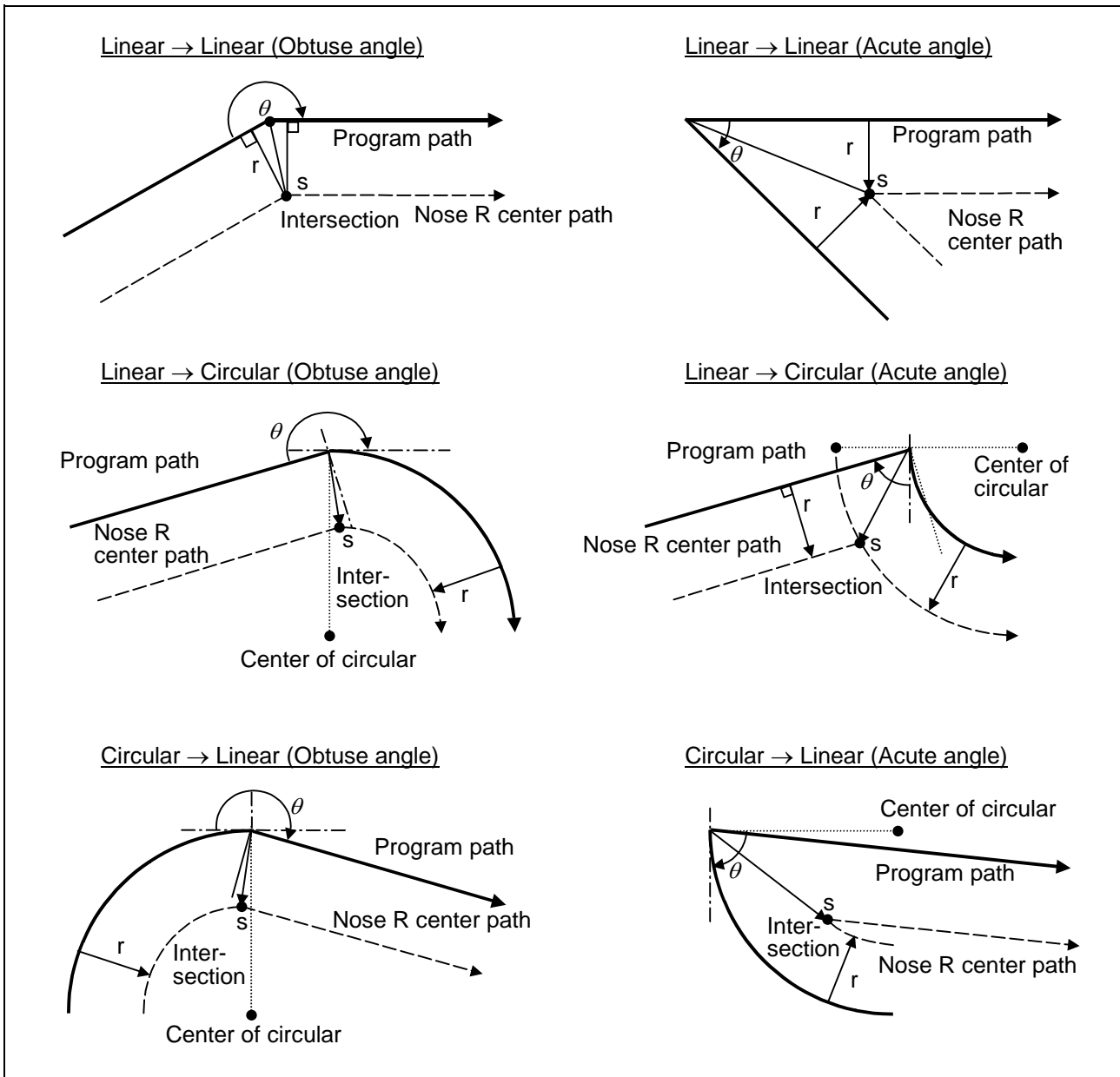
Circular  $\rightarrow$  Circular ( $0^\circ < \theta < 90^\circ$ )



## 12. Tool Offset Functions

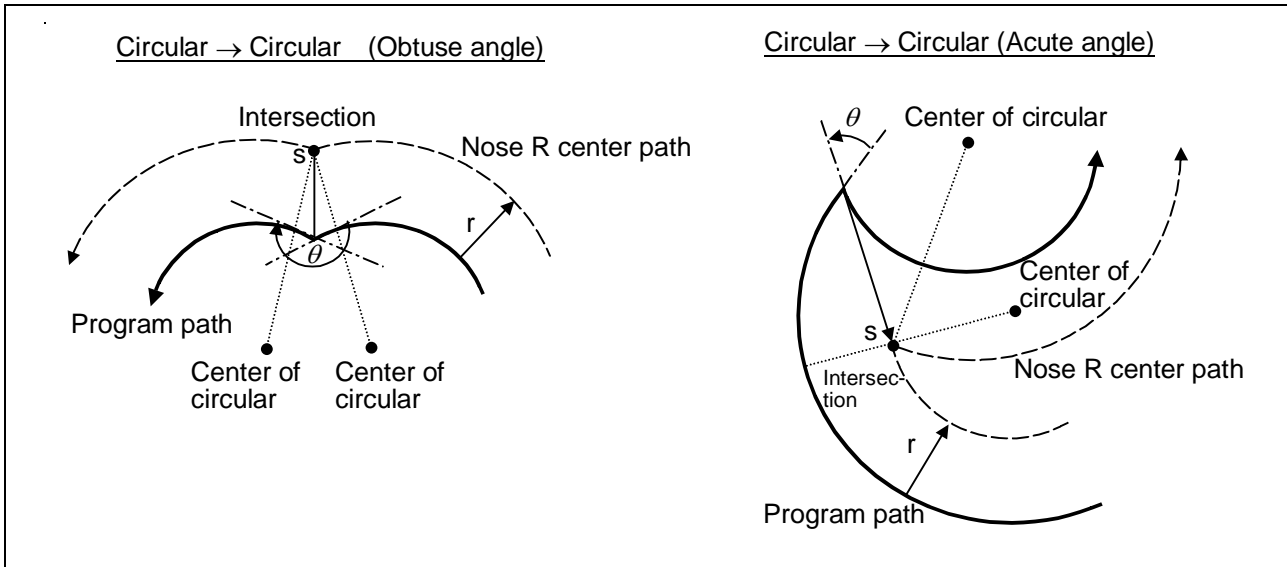
### 12.4 Tool nose radius compensation

(2) Machining an inner wall

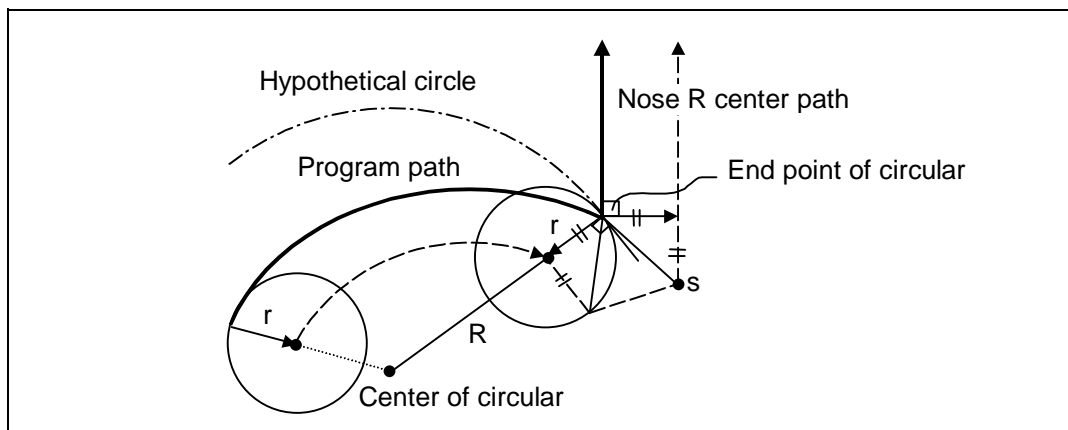


## 12. Tool Offset Functions

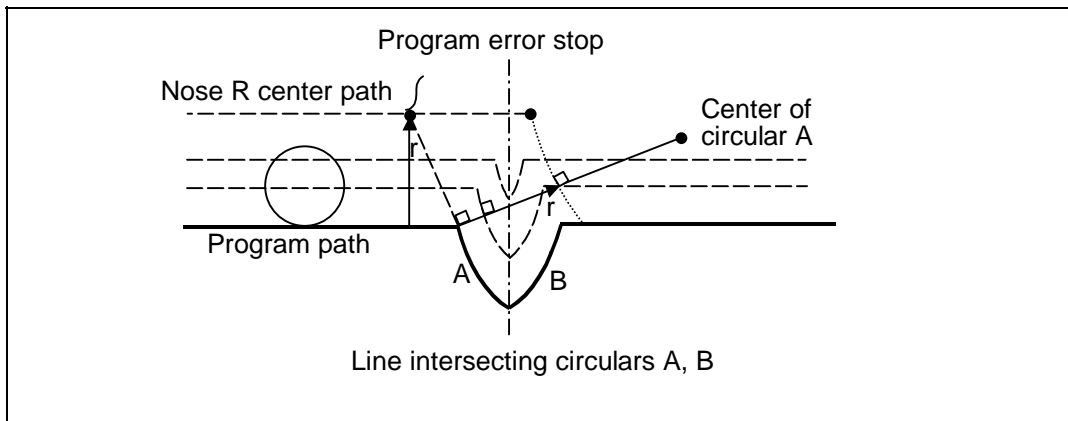
### 12.4 Tool nose radius compensation



- (3) When the arc end point is not on the arc  
 If the error applying after compensation is within the parameter (#1084 RadErr), the area from the arc start point to the end point is interpolated as a spiral arc.



- (4) When the inner intersection point does not exist  
 In an instance such as that shown in the figure below, the intersection point of circulars A and B may cease to exist due to the offset amount. In such cases, program error (P152) appears and the tool stops at the end point of the previous block.





#### Tool nose radius compensation cancel

If either of the following conditions is met in the tool nose radius compensation mode, the compensation will be canceled. However, the movement command must be a command which is not a circular command.

If the compensation is canceled by a circular command, program error (P151) results.

- (1) The G40 command has been executed.
- (2) The T00 tool number has been executed.

The cancel mode is established once the compensation cancel command has been read, 5-block pre-reading is suspended and 1-block pre-reading is applied instead.



#### Tool nose radius compensation cancel operation

Issuing a tool nose radius compensation cancel command results in the following.

- (1) If the command before G40 is G00 after the tool nose radius compensation is completed, the tool nose radius compensation is temporarily stopped, and the tool nose radius compensation is canceled in that state by an independent G40 command.
- (2) If there is an interpolation command before G40 after the tool nose radius compensation is completed, the tool nose radius compensation is not canceled by the independent G40 command, so the nose R center can stop in a vertical position. Instead, the tool nose radius compensation is canceled by the first axis movement command after G40. The nose radius compensation will not be canceled by the axis command if there is no axis movement. If there is no axis movement command after G40, and the program finishes by M02, etc., the tool nose radius compensation will remain valid. Resetting will cancel the tool nose radius compensation, but the operation will not be canceled.
- (3) When an independent T00 command is issued in a block, a tool nose radius cancel mode will result in that block, and the axis will move to the tool nose radius cancel position.



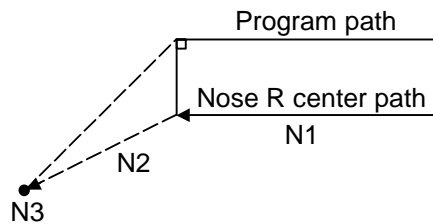
## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

- (4) Relation of an inside corner/outside corner and cancel  
 (a)-1 When an independent G40 command is issued at an inside corner

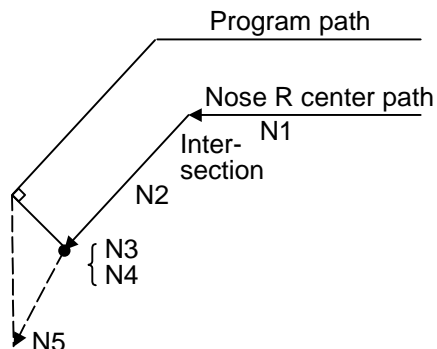
```

}
N1 G01 X_Z_F_;
N2 G00 X_Z_;
N3 G40;
}
  
```



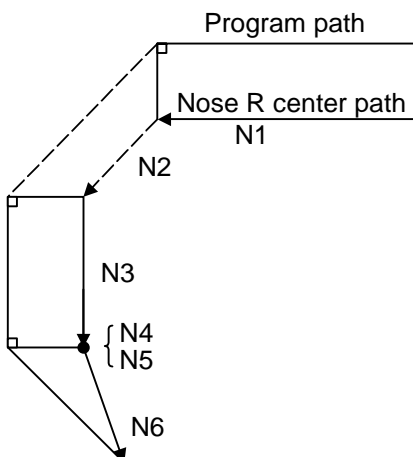
```

}
N1 G01 X_Z_F_;
N2 G01 X_Z_;
N3 G40;
N4 M05;
N5 G00 X_Z_;
}
  
```



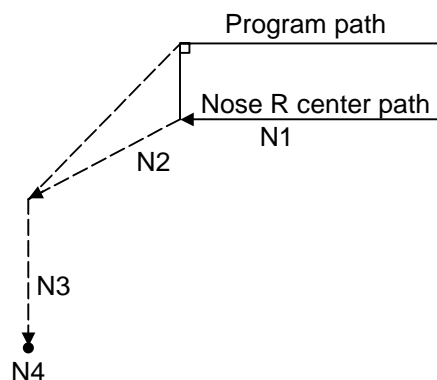
```

}
N1 G01 X_Z_F_;
N2 G00 X_Z_;
N3 G01 X_Z_;
N4 G40;
N5 M05;
N6 G01 X_Z_;
}
  
```



```

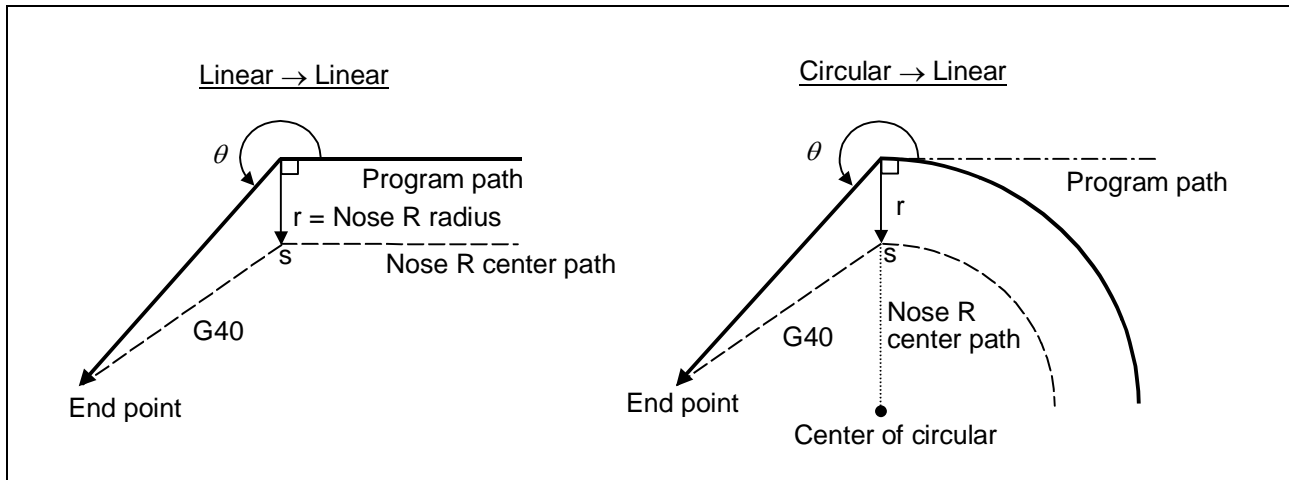
}
N1 G01 X_Z_F_;
N2 G00 X_Z_;
N3 G00 X_Z_;
N4 G40;
}
  
```



## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

(a)-2 When a G40 command is issued at an inside corner, in the same block as a movement command



# 12. Tool Offset Functions

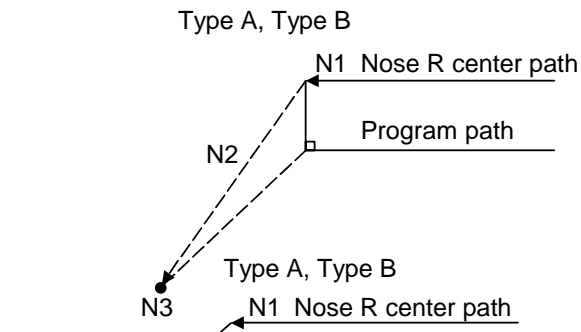
## 12.4 Tool nose radius compensation

(b)-1 When an independent G40 command is issued at an outside corner (obtuse angle)

```

}
N1 G01 X_Z_F_;
N2 G00 X_Z_;
N3 G40;
}

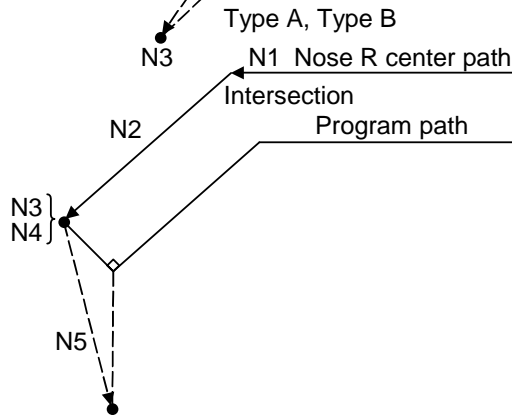
```



```

}
N1 G01 X_Z_F_;
N2 G01 X_Z_;
N3 G40;
N4 M05;
N5 G00 X_Z_;
}

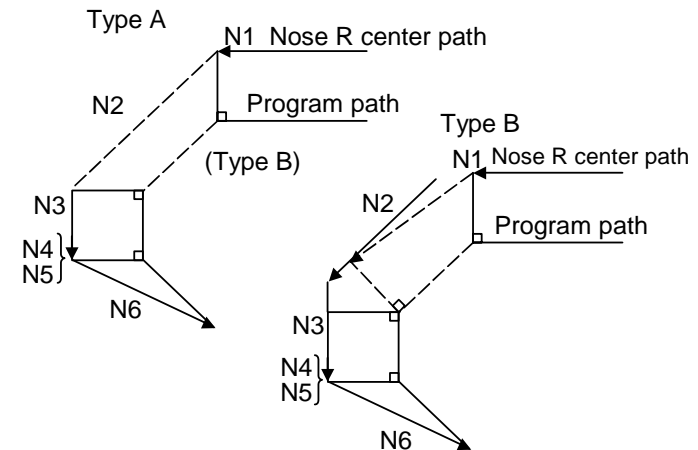
```



```

}
N1 G01 X_Z_F_;
N2 G00 X_Z_;
N3 G01 X_Z_;
N4 G40;
N5 M05;
N6 G01 X_Z_;
}

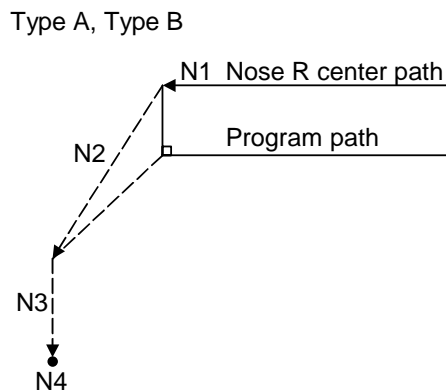
```



```

}
N1 G01 X_Z_F_;
N2 G00 X_Z_;
N3 G00 X_Z_;
N4 G40;
}

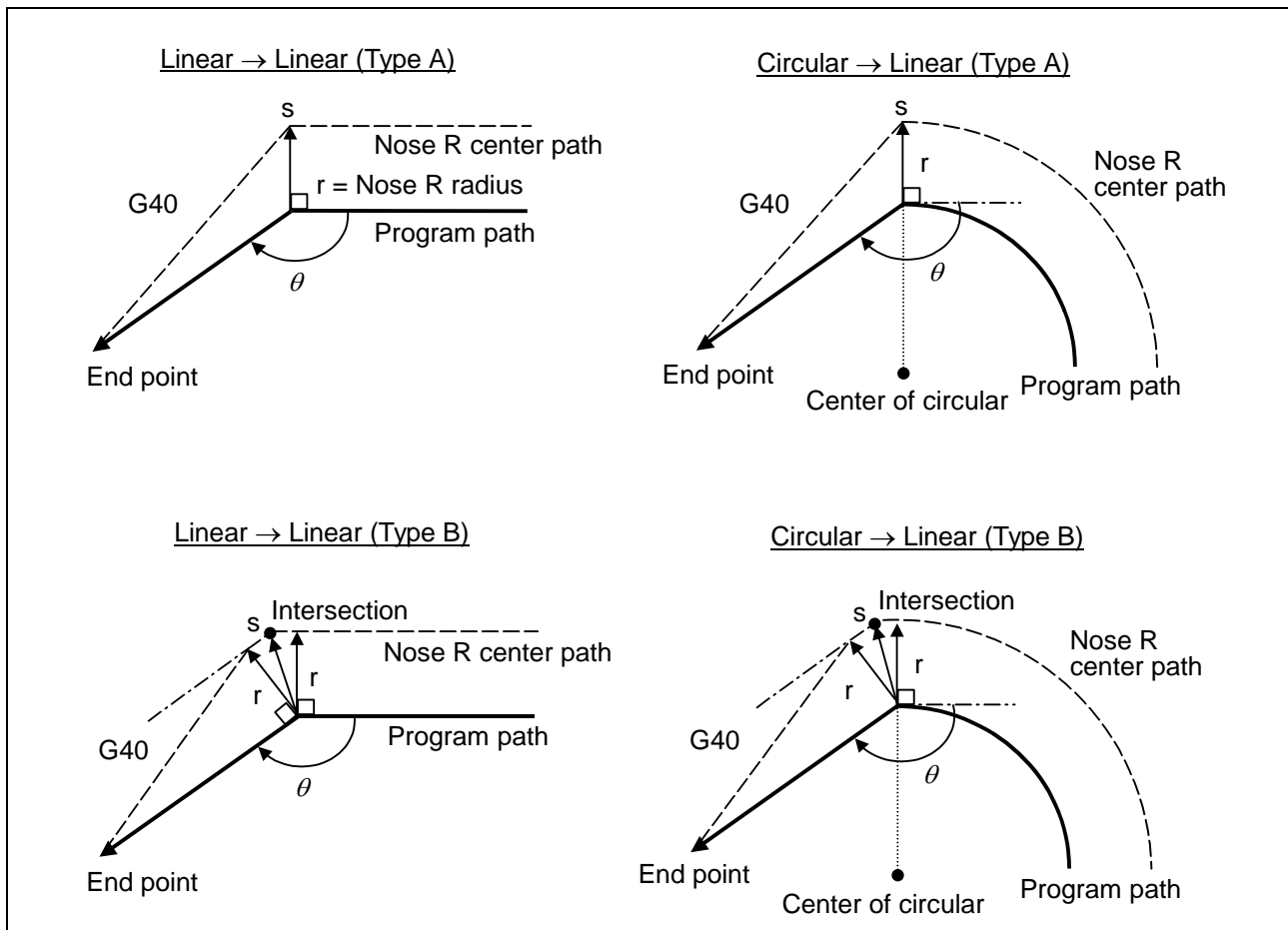
```



## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

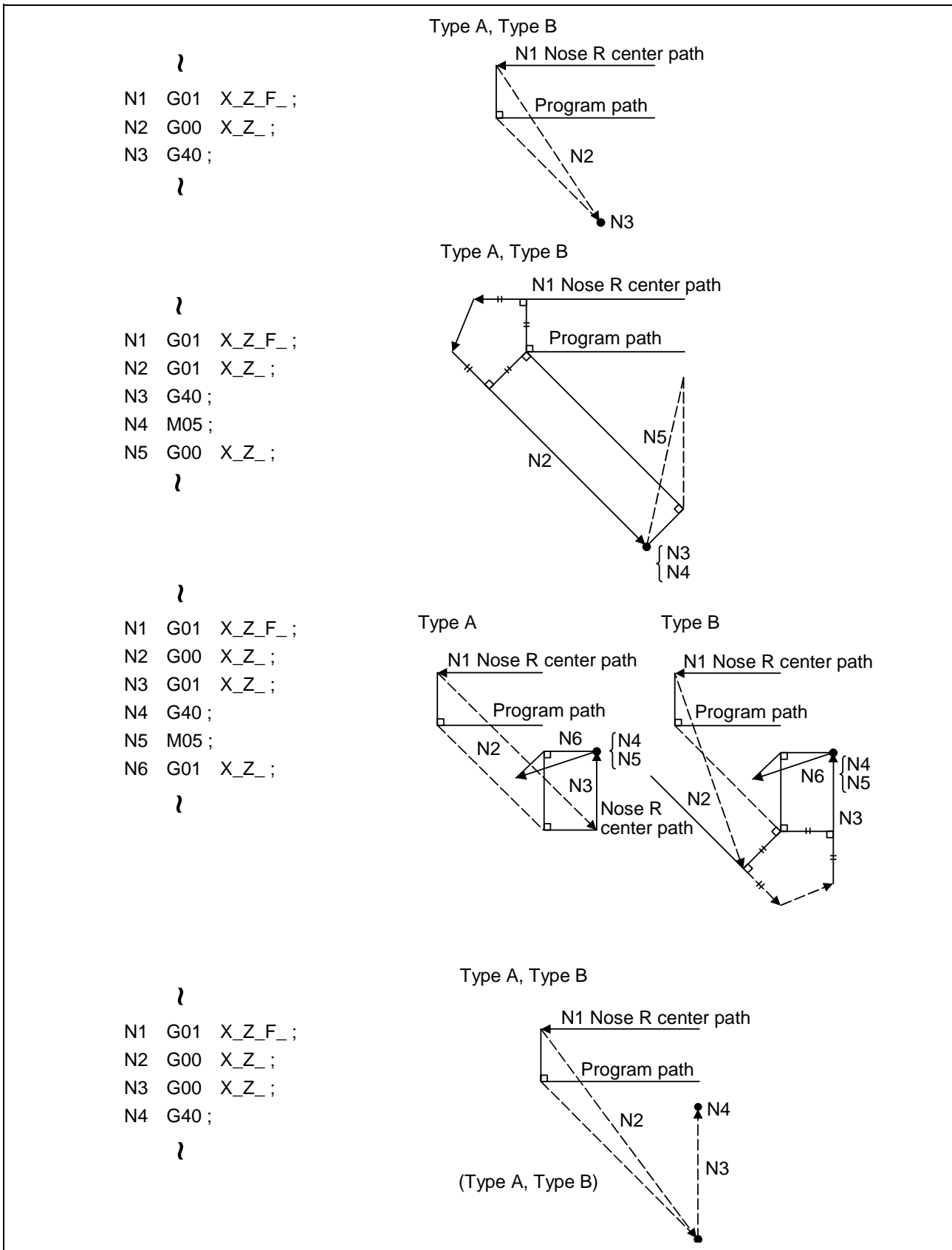
(b)-2 When a G40 command is issued at an outside corner (obtuse angle), in the same block as a movement command



## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

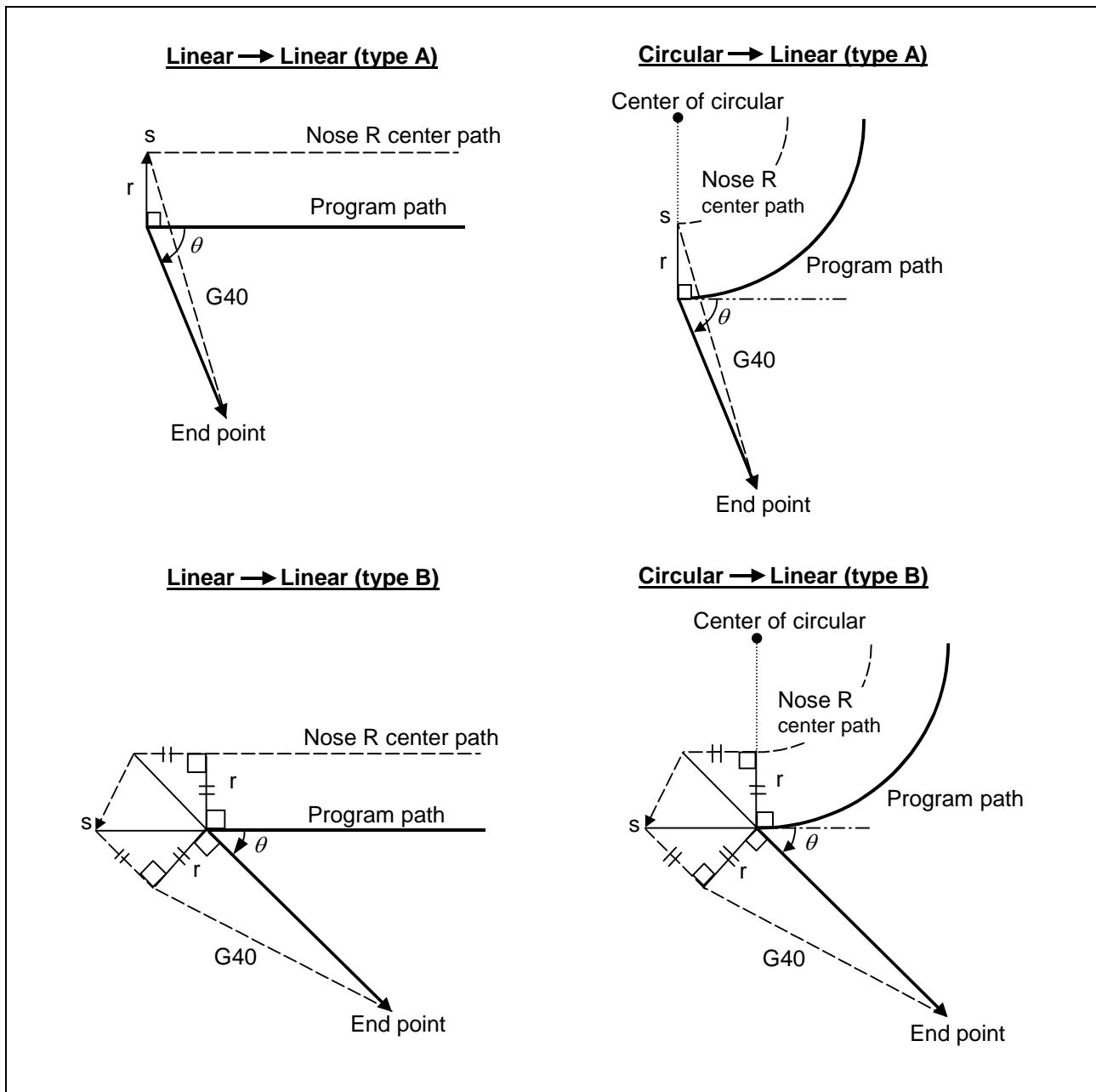
(c)-1 When an independent G40 command is issued at an outside corner (acute angle)



## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

(c)-2 When a G40 command is issued at an outside corner (acute angle), in the same block as a movement command



## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

#### 12.4.3 Other operations during tool nose radius compensation

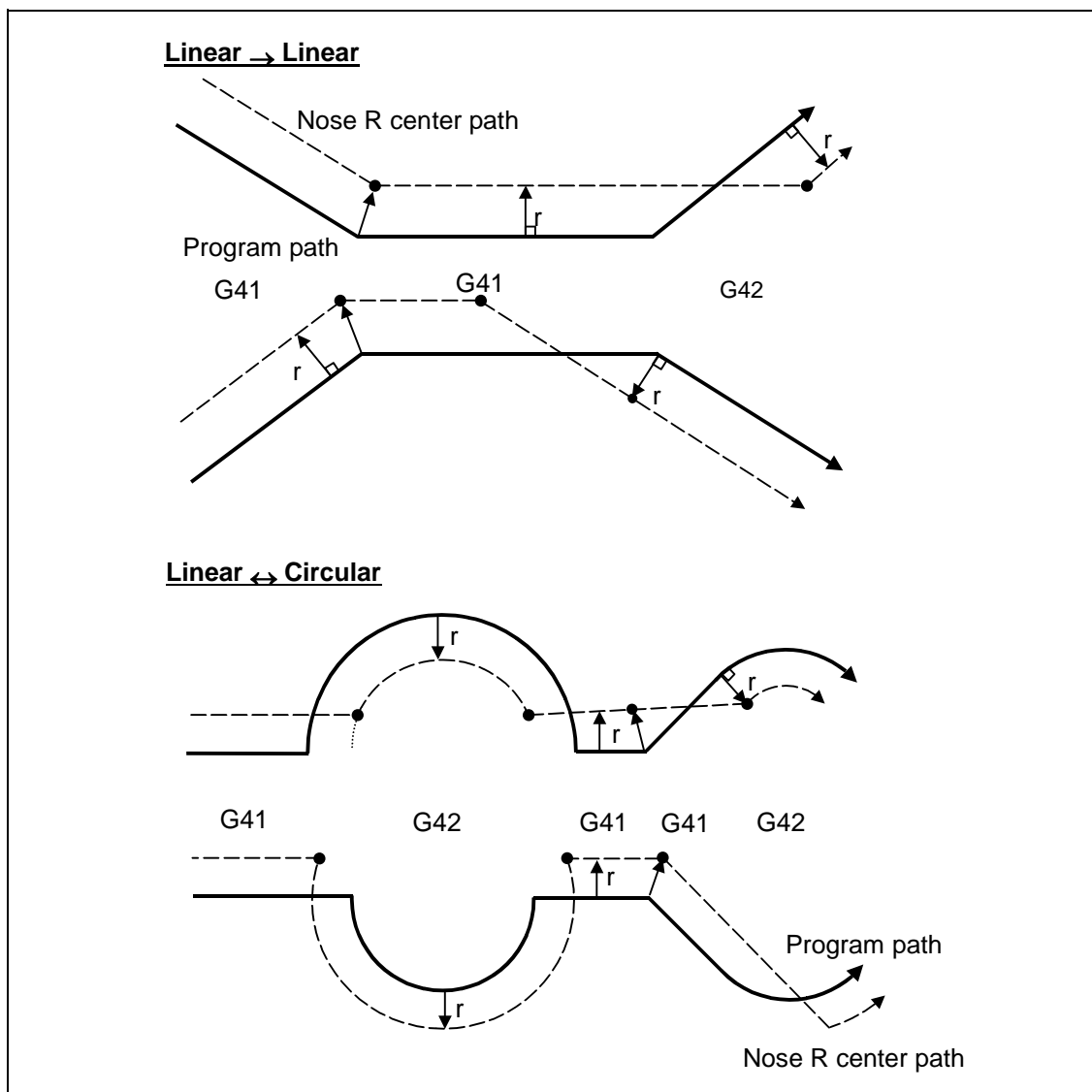


#### Changing the compensation direction during tool nose radius compensation

The compensation direction is determined by the tool nose radius compensation commands (G41, G42).

G code	Compensation direction
G41	Left-side compensation
G42	Right-side compensation

The compensation direction can be changed by changing the compensation command in the compensation mode without the compensation having to be first canceled. However, no change is possible in the compensation start block and the following block.



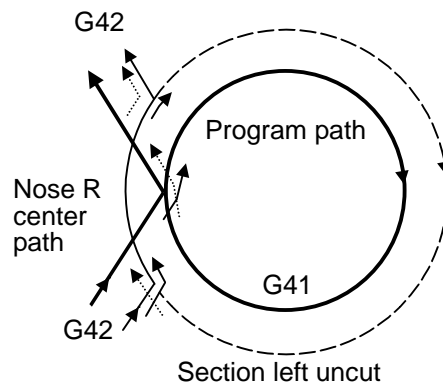
## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

#### Linear return



#### Arc exceeding 360° due to compensation method



In the cases below, it is possible that the circular may exceed 360°.

a. With offset direction selection based on G41/G42

In cases like this, compensation is provided shown in the figure on the left and a section will be left uncut.

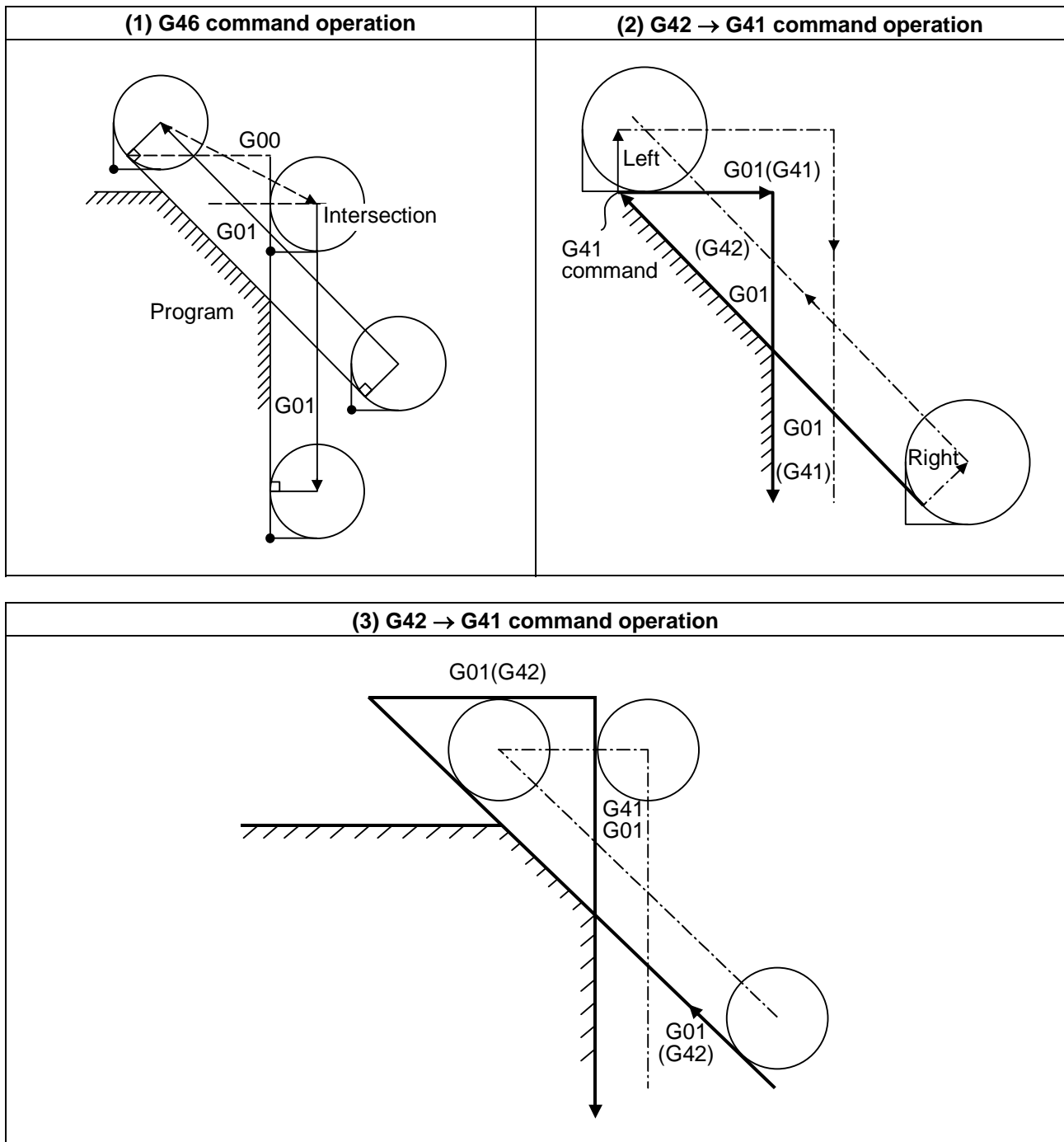


# 12. Tool Offset Functions

## 12.4 Tool nose radius compensation



Tool nose radius compensation of path closed by G46/G41/G42

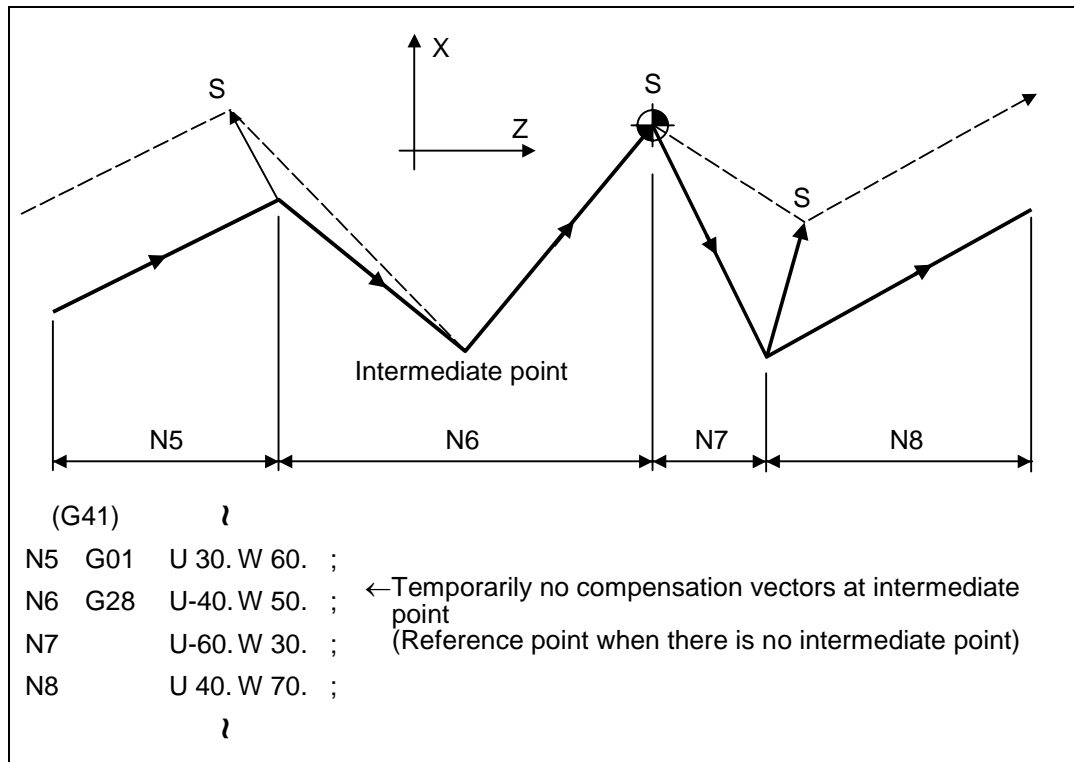




#### Command for eliminating offset vectors temporarily

When the following command is issued in the compensation mode, the offset vectors are temporarily eliminated and a return is then made automatically to the compensation mode. In this case, the compensation is not canceled, and the tool goes directly from the intersection point vector to the point without vectors or, in other words, to the programmed command point. When a return is made to the compensation mode, it goes directly to the intersection point.

#### (1) Reference point return command



(2) The compensation vector will be eliminated temporarily with the G53 command (basic machine coordinate system selection).

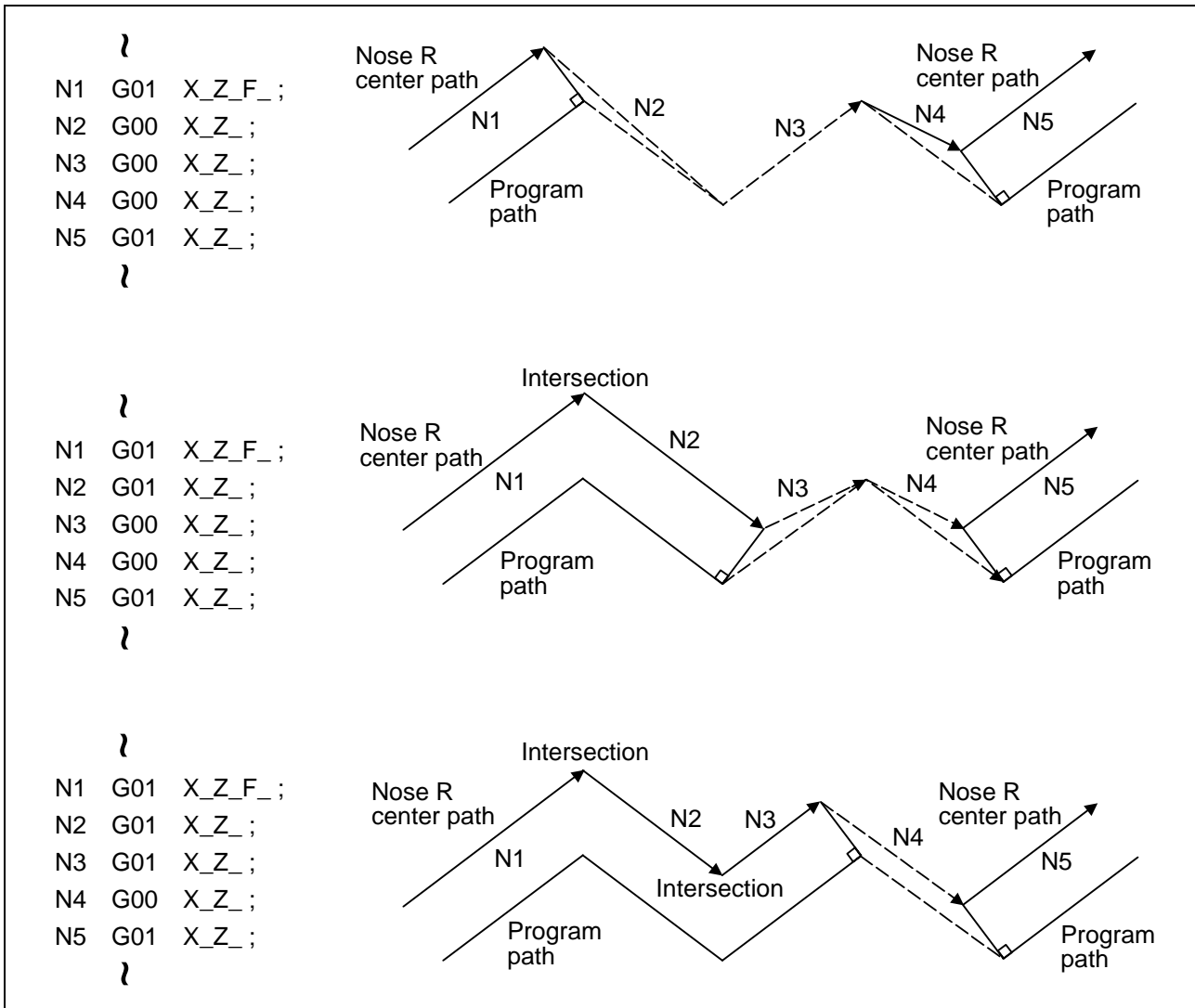
(Note 1) The offset vectors do not change with the coordinate system setting (G92) command.

## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

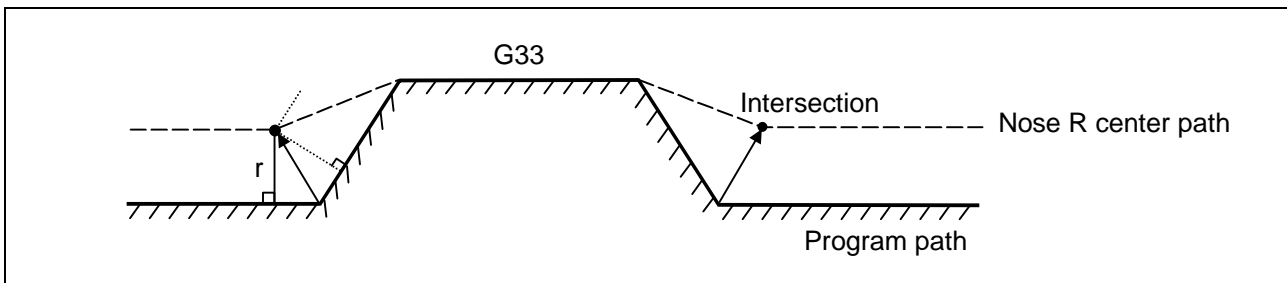
#### (3) Positioning (G00) commands

Tool nose radius compensation is temporarily canceled for G00 commands.



#### (4) G33 thread cutting command

Tool nose radius compensation does not apply to the G33 block.



### (5) Compound fixed cycles

When a compound fixed cycle I command (G70, G71, G72, G73) is assigned, the tool nose radius compensation is temporarily canceled, the finished shape to which tool nose radius compensation has been applied is cut with the compensation canceled and, upon completion, a return is automatically made to the compensation mode.



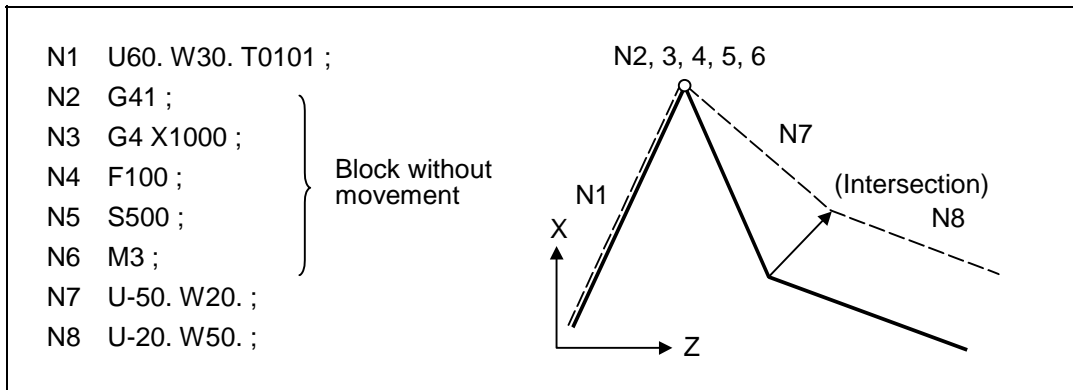
### Blocks without movement

The following blocks are known as blocks without movement.

a.M03 ;	.....	M command	} No movement
b.S12 ;	.....	S command	
c.T0101 ;	.....	T command	
d.G04 X500 ;	.....	Dwell	
e.G10 P01R50 ;	.....	Offset amount setting	
f.G92 X600. Z500. ;	.....	Coordinate system setting	
g.Y40. ;	.....	Movement but not on offset plane	
G.00 ;	.....	G code only	
i.U0 ;	.....	Zero movement amount	
			} Movement amount is zero

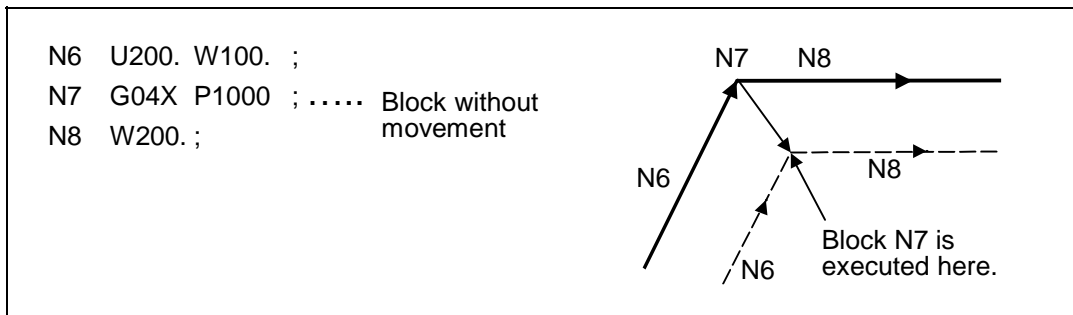
### (1) When commanded at compensation start

Compensation vector cannot be generated when 4 or more blocks continue without movement.



### (2) When command is assigned in the compensation mode

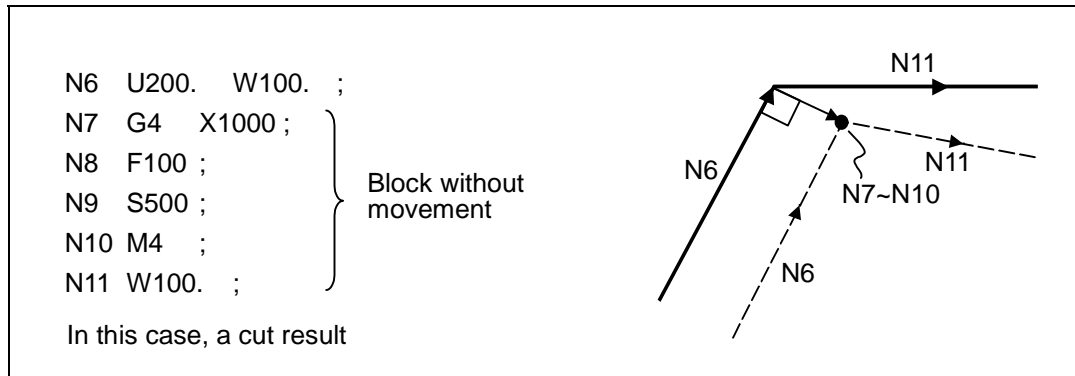
When 4 or more blocks without movement follow in succession in the compensation mode, the intersection point vectors will be created as usual.



## 12. Tool Offset Functions

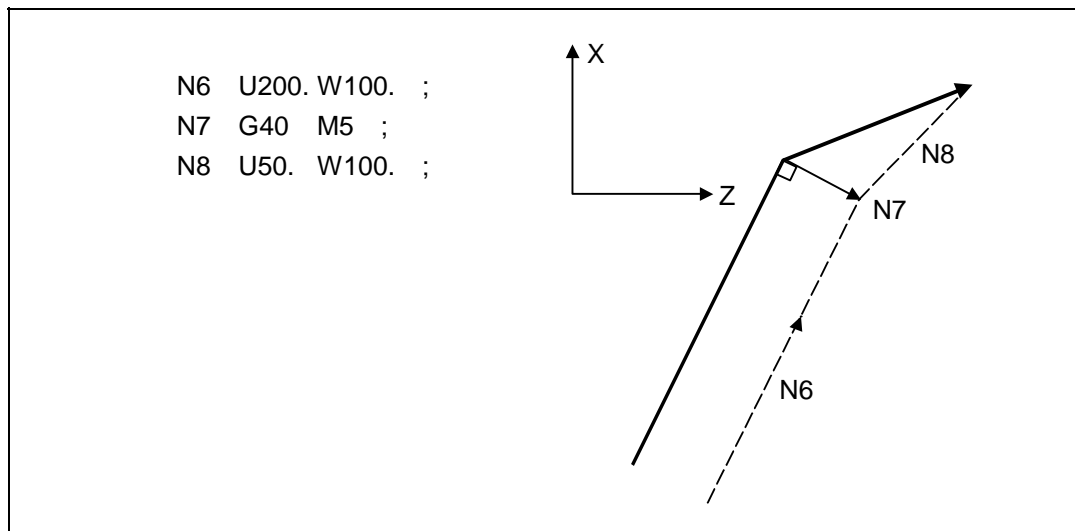
### 12.4 Tool nose radius compensation

When 4 or more blocks without movement follow in succession, the offset vectors are created perpendicularly at the end point of the previous block.



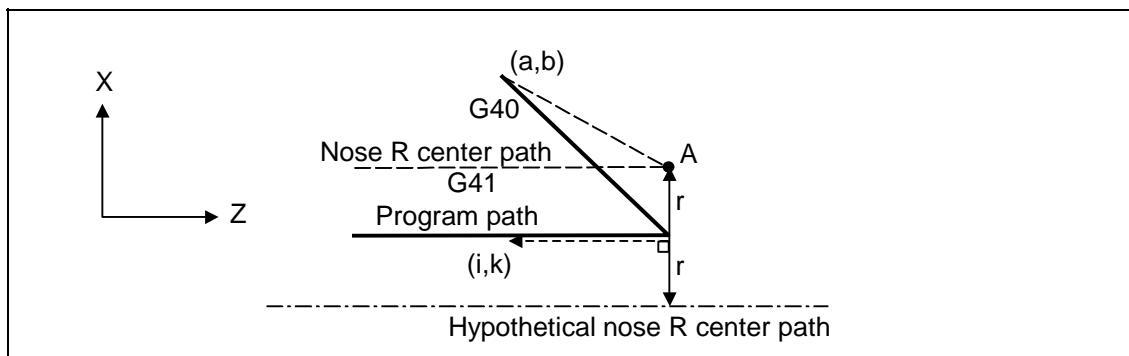
#### (3) When commanded together with compensation cancel

Only the offset vectors are canceled when a block without movement is commanded together with the G40 command.



#### When I, J, K are commanded in G40

A perpendicular vector is created in the block before G40.

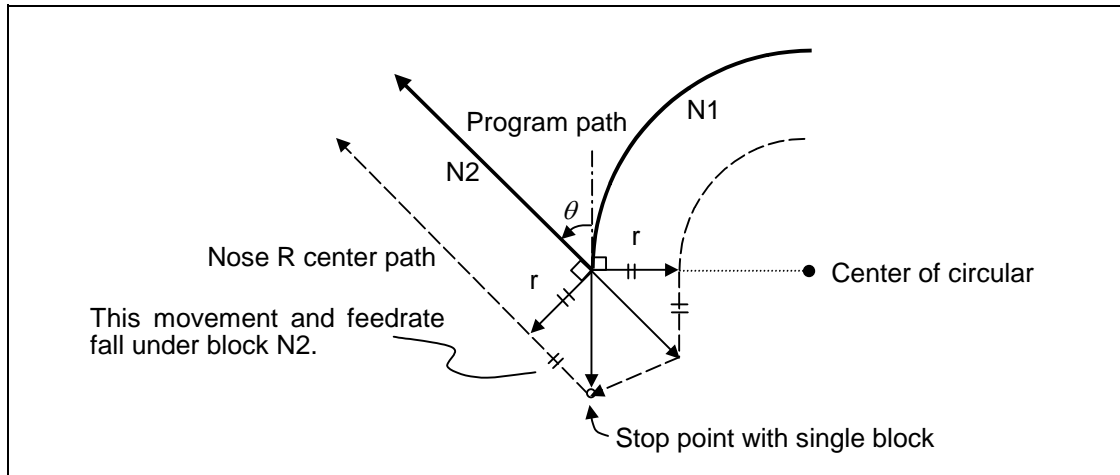




#### Corner movement

When a multiple number of compensation vectors are created at the joints between movement command blocks, the tool will move in a linear between those vectors. This action is called corner movement.

When the vectors do not coincide, the tool moves in order to machine the corner. Consequently, operation in the single block mode will execute the previous block + corner movement as a single block and the remaining joining movement + following block will be executed as a single block in the following operation.



## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

#### 12.4.4 G41/G42 commands and I, J, K designation



##### Function and purpose

The compensation direction can be intentionally changed by issuing the G41/G42 command and I, J, K in the same block.



##### Command format

**G18 (Z-X plane) G41/G42 X\_ Z\_ I\_ K\_ ;**

Assign a linear command (G00, G01) in a movement mode.

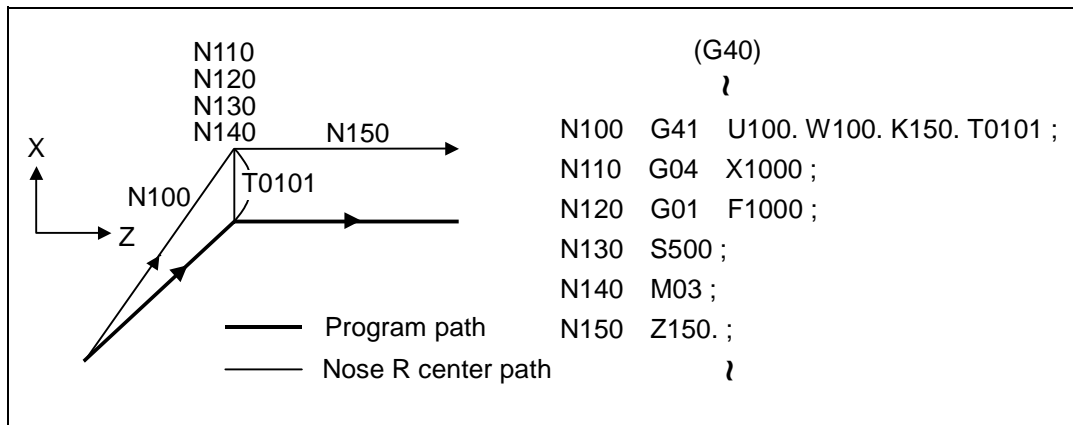


##### I, K type vectors (G18 X-Z plane selection)

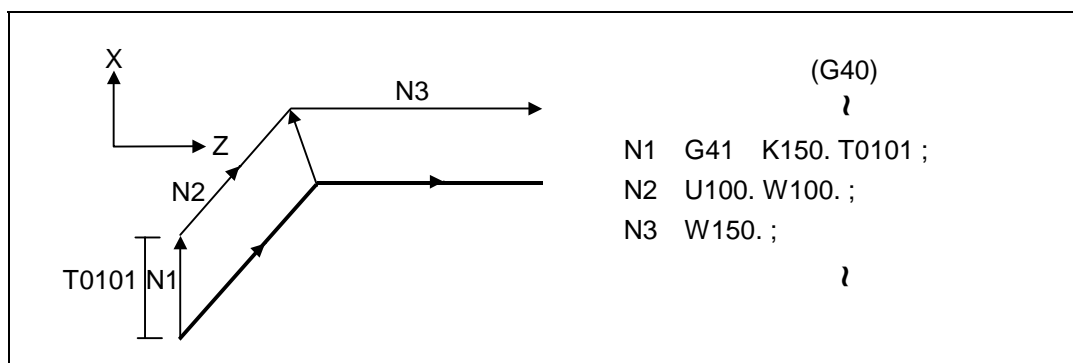
The new I, K type vector (G18 plane) created by this command is now described. (Similar descriptions apply to vector I, J for the G17 plane and to J, K for the G19 plane.)

As shown in the figures below, the vectors with a size equivalent to the offset amount are made to serve as the I, K type compensation vector perpendicularly to the direction designated by I, K without the intersection point of the programmed path being calculated. The I, K vector can be commanded even in the mode (G41/G42 mode in the block before) and even at the compensation start (G40 mode in the block before).

##### (1) When I, K is commanded at compensation start



##### (2) When there are no movement commands at the compensation start



## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

#### (3) When I, K has been commanded in the mode (G18 plane)

(1) I, K type vector  
 (2) Intersection point calculation type vector

```

}
(G18 G41)
N100 G41 T0101 G01 W150. I50. ;
N110 G02 W100. I50. ;
N120 G01 W100.
}
  
```

——— Program path  
 - - - - - Nose R center path  
 - - - - - Path for intersection point calculation

#### (4) When I, K has been commanded in a block without movement

```

N1 G41 T0101 G01 F1000 ;
N2 U100. W100. ;
N3 G41 K50. ;
N4 W150. ;
N5 G40 ;
  
```



## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation



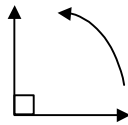
#### Direction of offset vectors

##### (1) In G41 mode

Direction produced by rotating the direction commanded by I, K through 90° to the left from the forward direction of the Y axis (3rd axis) as seen from the zero point

**(Example 1)** With K100.

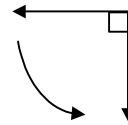
Offset vector direction



(0, 100) IK direction

**(Example 2)** With K -100.

(0, -100) IK direction



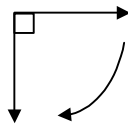
Offset vector direction

##### (2) In G42 mode

Direction produced by rotating the direction commanded by I, K through 90° to the right from the forward direction of the Y axis (3rd axis) as seen from the zero point

**(Example 1)** With K100.

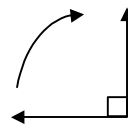
(0, 100) IK direction



Offset vector direction

**(Example 2)** With K -100.

Offset vector direction

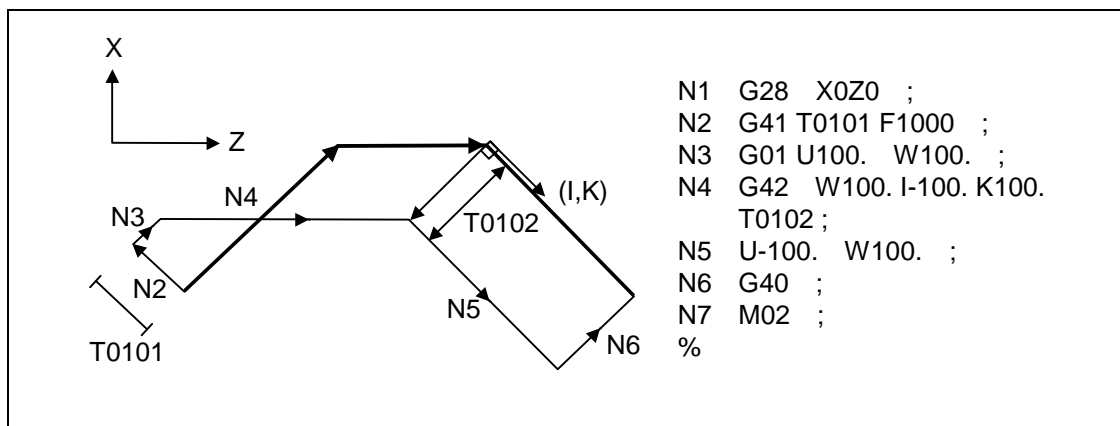


(0, -100) IK direction



#### Selection of offset modal

The G41 or G42 modal can be selected at any time.





### Offset amount for offset vectors

The offset amounts are determined by the offset number (modal) in the block with the I, K designation.

**<Example 1>**

```

(G41 T0101)
}
N100 G41 W150. K50. ;
N110 U-100. W100. ;
}
    
```

Vector (A) is the offset amount entered in tool offset number modal 1 in the N100 block.

**<Example 2>**

```

(G41 T0101)
}
N200 G41 W150. K50. T0102 ;
N210 U-100. W100. ;
}
    
```

Vector (B) is the offset amount entered in tool offset number modal 2 in the N200 block.

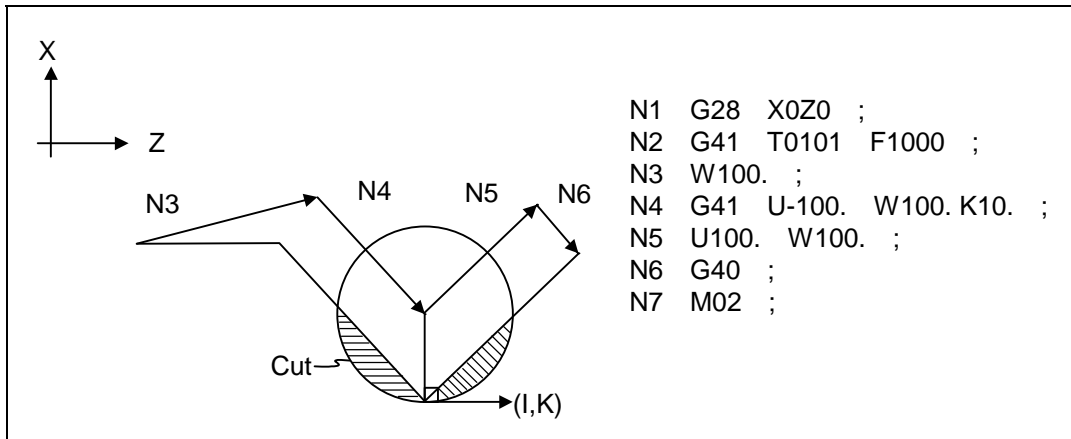
## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation



#### Precautions

- (1) Issue the I, K type vector in a linear mode (G0, G1). If it is issued in an arc mode at the start of compensation, program error (P151) will result.  
An I, K designation in an arc mode functions as an arc center designation in the offset mode.
- (2) When the I, K type vector has been designated, it is not deleted (avoidance of interference) even if there is interference. Consequently, overcutting may arise in such a case.



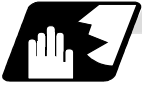
- (3) Refer to the following table for the offset methods based on the presence and/or absence of the G41 and G42 commands and I, K (J) command.

G41/G42	I, K (J)	Offset method
No	No	Intersection point calculation type vector
No	Yes	Intersection point calculation type vector
Yes	No	Intersection point calculation type vector
Yes	Yes	I, K type vector No insertion block

## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

#### 12.4.5 Interrupts during tool nose radius compensation

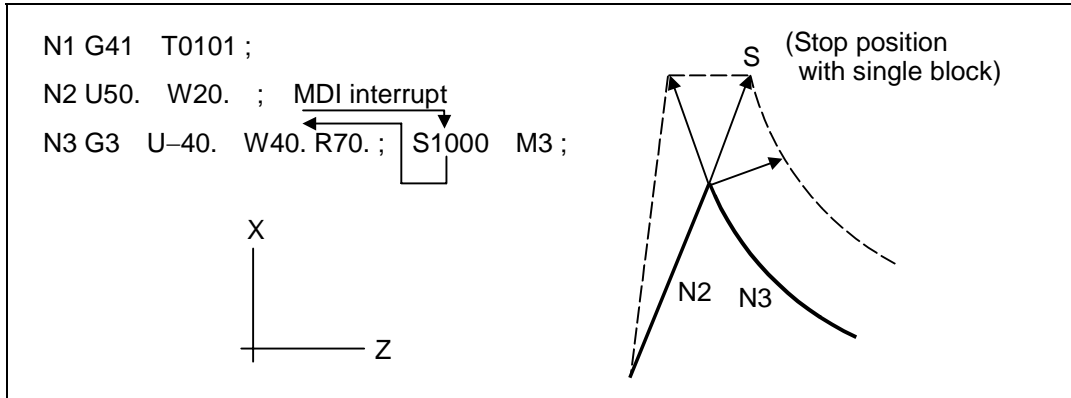


#### MDI interrupt

Tool nose radius compensation is valid in any automatic operation mode - whether tape, memory or MDI operation.

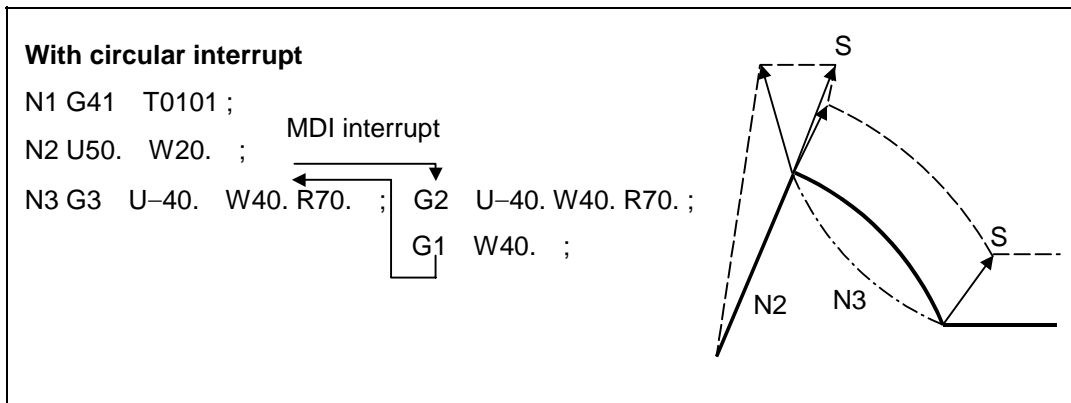
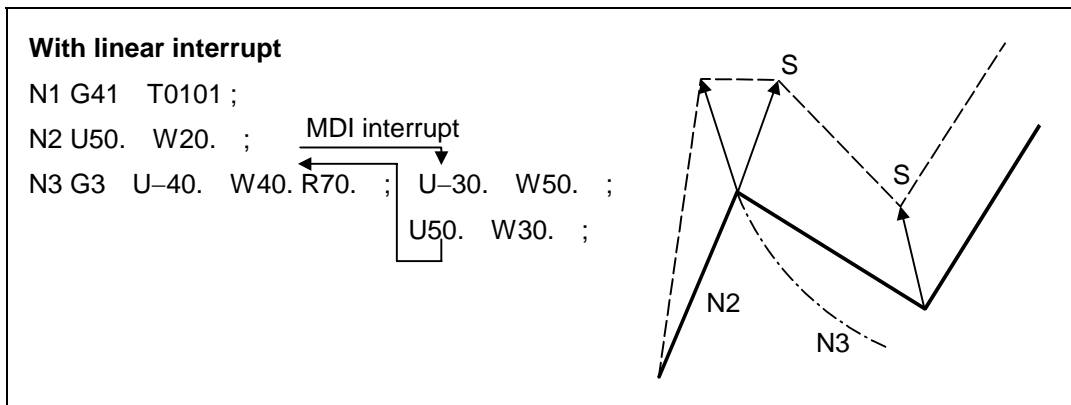
An interrupt based on MDI will give the result as in the figure below after block stop during tape or memory operation.

- (1) Interrupt without movement (tool path does not change)



- (2) Interrupt with movement

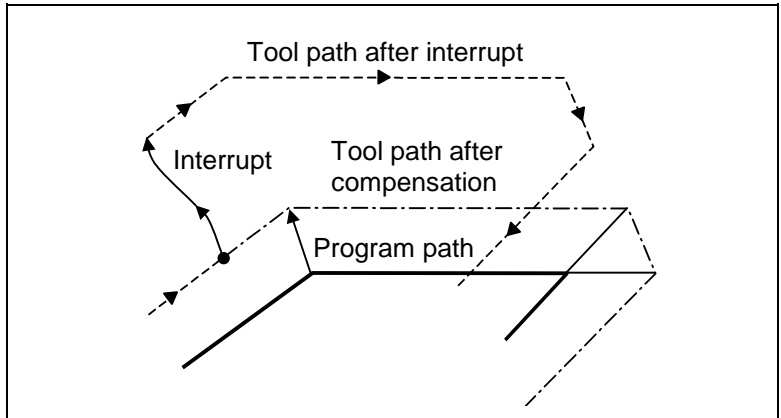
The offset vectors are automatically re-calculated at the movement block after interrupt.



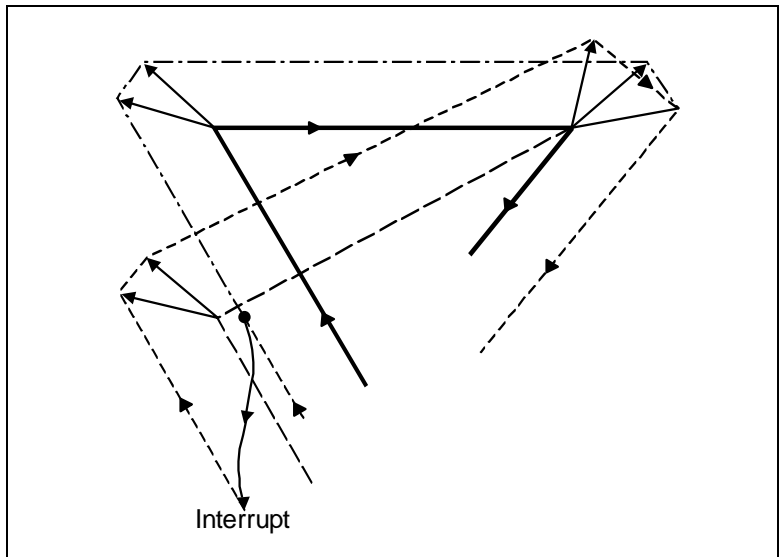


### Manual interrupt

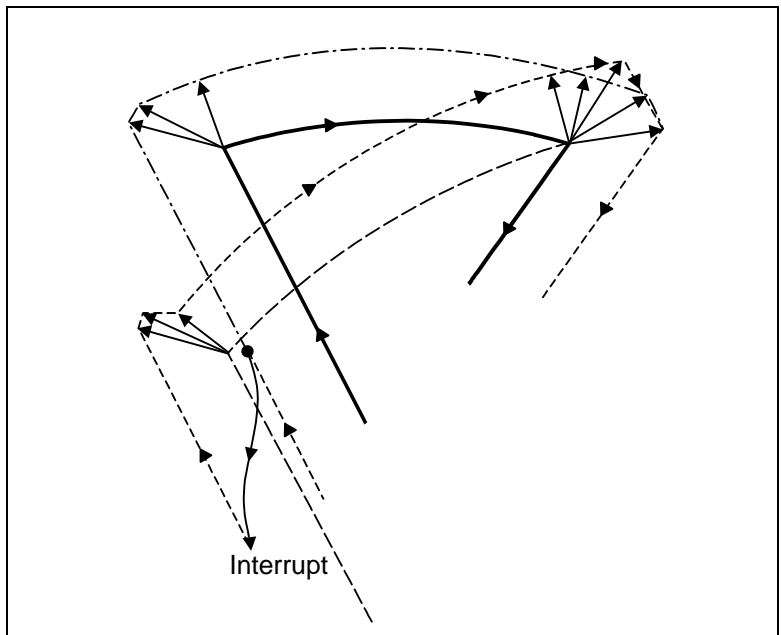
- (1) Interrupt with manual absolute OFF.  
The tool path is shifted by an amount equivalent to the interrupt amount.



- (2) Interrupt with manual absolute ON.  
In the incremental value mode, the same operation results as with manual absolute OFF.



In the absolute value mode, however, the tool returns to its original path at the end point of the block following the interrupted block, as shown in the figure.



#### 12.4.6 General precautions for tool nose radius compensation



##### Assigning the compensation amounts

- (1) The compensation amount is normally assigned by designating the number of the compensation amount by the last 1 or 2 digits of the T code. Depending on the machine specifications, the high-order digits may be used. The T code will remain valid once designated until another T code is subsequently commanded.  
Besides being used to designate the compensation amounts for tool nose radius compensation, the T codes are also used to designate the offset amounts for tool length offset.
- (2) The compensation amounts are normally changed when a different tool has been selected in the compensation cancel mode. However, when an amount is changed in the compensation mode, the vectors at the end point of the block are calculated using the compensation amount designated in that block.



##### Errors during tool nose radius compensation

- (1) An error results when any of the following commands are programmed during tool nose radius compensation.  
G17, G18, G19 ("P112" when a plane differing from that applying during the compensation has been commanded)  
G31 ("P608")  
G74, G75, G76 ("P155")  
G81 ~ G89 ("P155")
- (2) An error ("P158") results when a tool nose point other than 1 to 8 has been designated in the G46 mode.
- (3) An error ("P156") results when the compensation direction is not determined by the movement vector of the initial cutting command even when the tool nose radius compensation operation has started in the G46 mode and 5 blocks have been pre-read.
- (4) An error ("P151") results when a circular command is issued in the first or last block of the tool nose radius compensation.
- (5) A program error ("P157") results when the offset direction is reversed in the G46 mode. A parameter can be set to move the tool in the same compensation direction. (Control parameter #8106 "G46 reversal error avoidance")
- (6) A program error ("P152") results during tool nose radius compensation when the intersection point is not determined with single block skip in the interference block processing.
- (7) A program error results when there is an error in one of the pre-read blocks during tool nose radius compensation.
- (8) A program error ("P153") results when interference arises under no interference avoidance conditions during tool nose radius compensation.
- (9) A program error ("P150") results when a tool nose radius compensation command is issued even though the tool nose radius compensation specification has not been provided.

## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

#### 12.4.7 Interference check



#### Function and purpose

A tool, whose tool nose has been compensated under the tool nose radius compensation function by the usual 2-block pre-read, may sometimes cut into the workpiece. This is known as interference, and interference check is the function which prevents this from occurring. There are two types of interference check, as indicated below, and each can be selected for use by parameter.

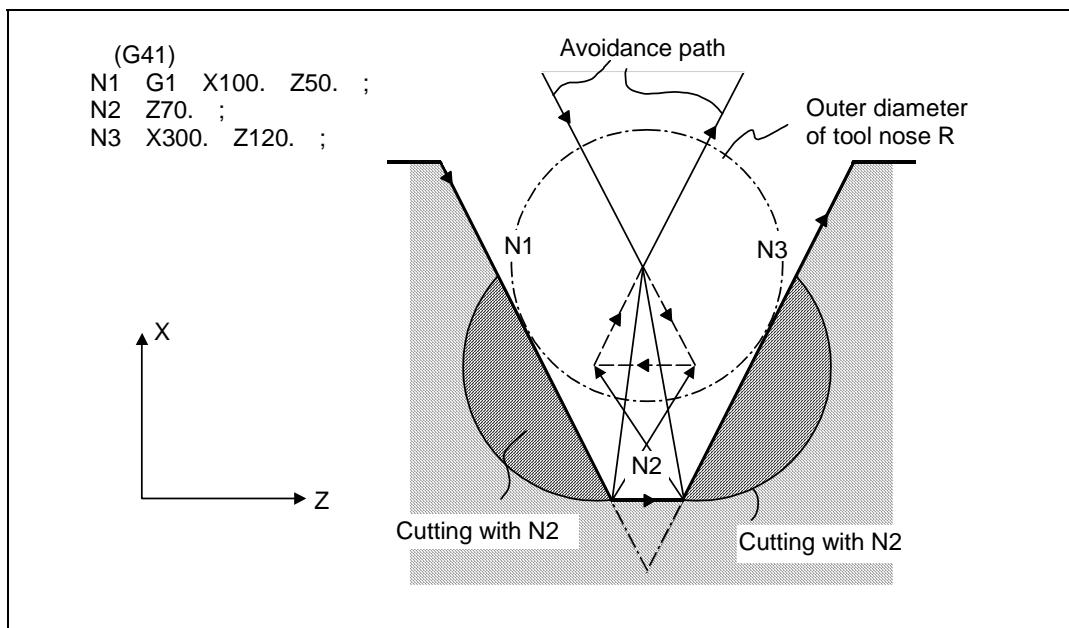
Function	Parameter	Operation
Interference check alarm function	Interference check avoidance OFF Interference check invalid OFF	A program error results before the execution of the block in which the cut arises, and operation stops.
Interference check avoidance function	Interference check avoidance ON Interference check invalid OFF	The tool path is changed so that workpiece is not cut into.
Interference check invalid function	Interference check invalid OFF	Cutting continues as is, even if the workpiece is cut into. Use in the fine segment program.

(Note) #8102 Interference avoidance  
#8103 Interference check invalid



#### Detailed description

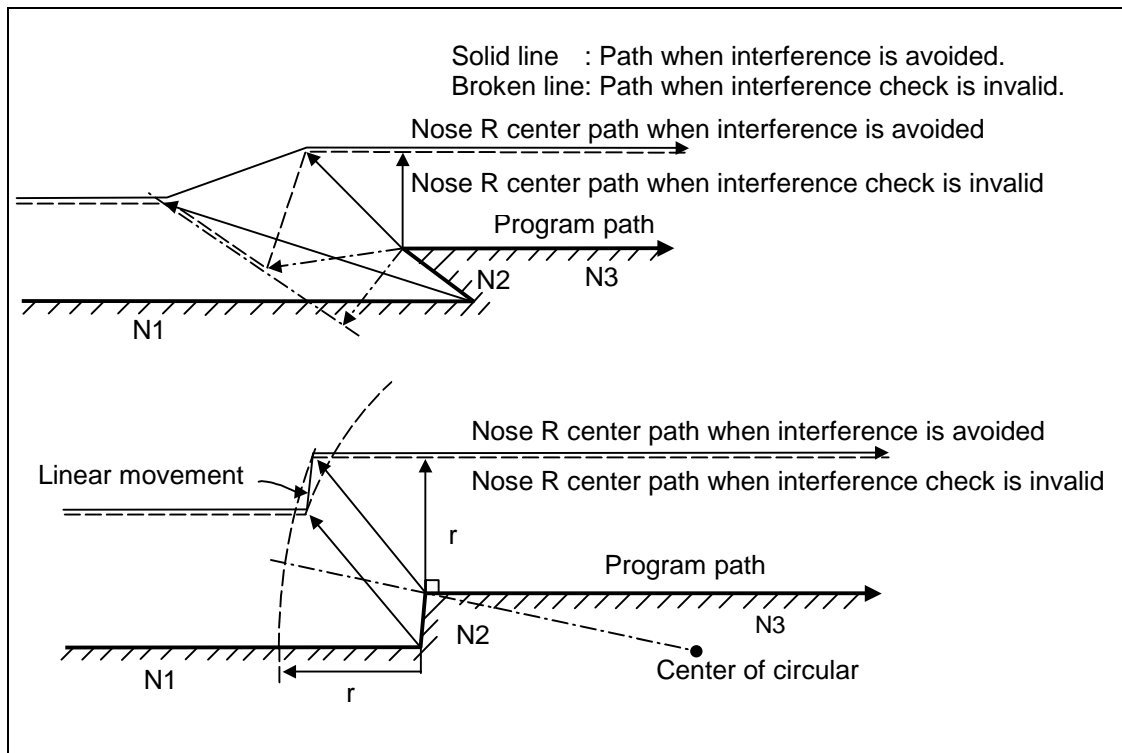
#### (Example)



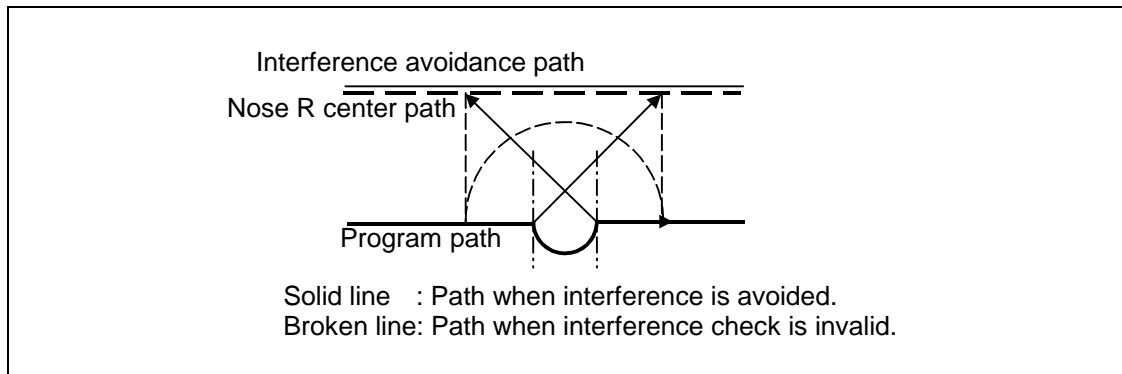
- (1) With alarm function  
The alarm occurs before N1 is executed and so, using the edit function, N1 can be changed as below and machining can be continued: N1 G1 X-100. Z-20. ;
- (2) With avoidance function  
The intersection point of N1 and N3 is calculated and the interference avoidance vectors are created.
- (3) With interference check invalid function  
The tool passes while cutting into the straight line between N1 and N3.



### Operation during interference avoidance



In the case of the figure below, the groove will be left uncut.





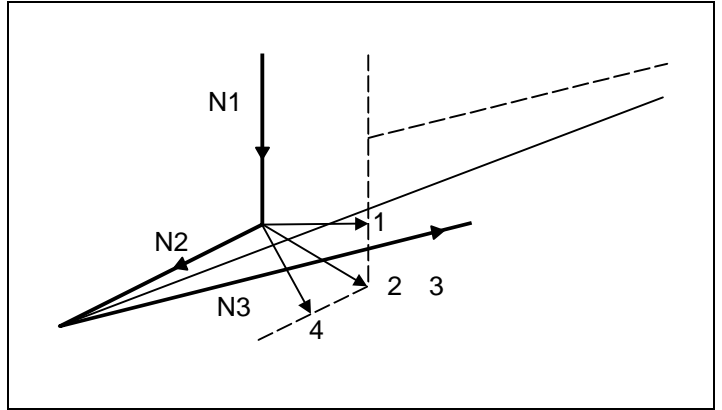


### Interference check alarm

The interference check alarm occurs under the following conditions.

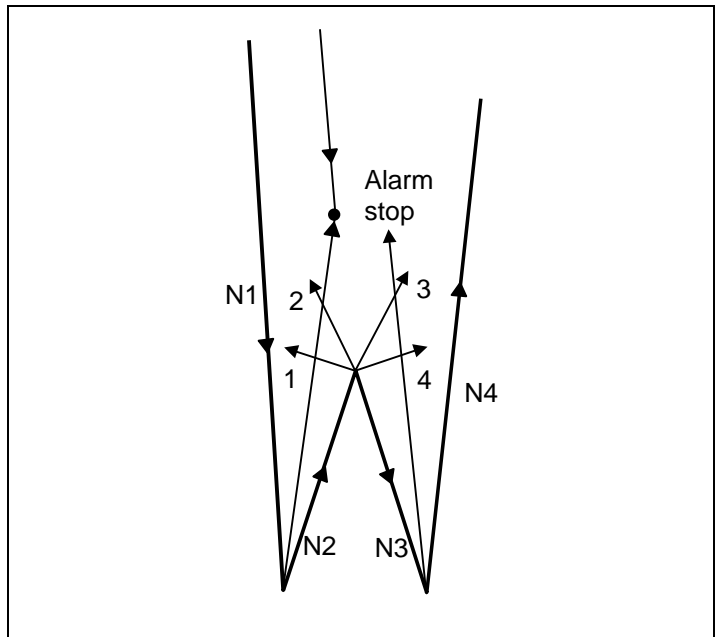
- (1) When the interference check alarm function has been selected
  - (a) When all the vectors at the end block of its own block have been deleted

When, as shown in the figure on the right, vectors 1 to 4 at the end point of the N1 block have all been deleted, program error (P153) results prior to N1 execution.

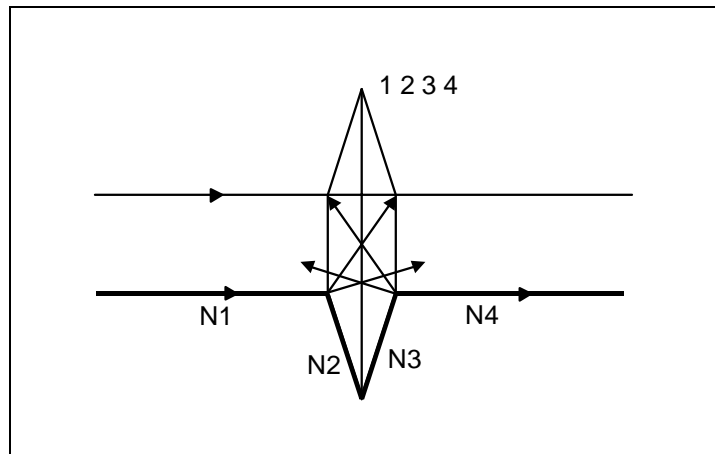


- (2) When the interference check avoidance function has been selected
  - (a) When there are valid vectors at the end point of the following block even when all the vectors at the end point of its own block have been deleted

- (i) When, in the figure on the right, the N2 in interference check is conducted, the N2 end point vectors are all deleted but the N3 end point vectors are regarded as valid. Program error (P153) now occurs at the N1 end point.

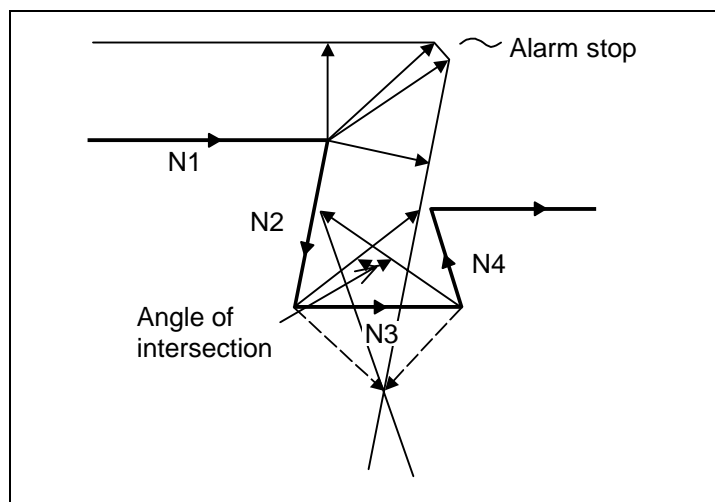
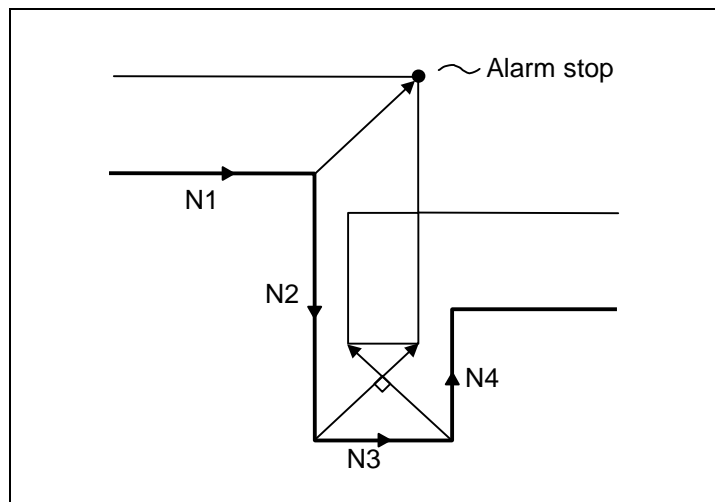


- (ii) In a case such as that shown in the figure on the below, the tool will move in the reverse direction at N2. Program error (P153) now occurs prior to N1 execution.



- (b) When avoidance vectors cannot be created

- (i) Even when, as in the figure on the below, the conditions for creating the avoidance vectors are met, it may still be impossible to create these vectors or the interference vectors may interfere with N3. As a result, program error (P153) will occur at the N1 end point when the vector intersecting angle is more than  $90^\circ$ .

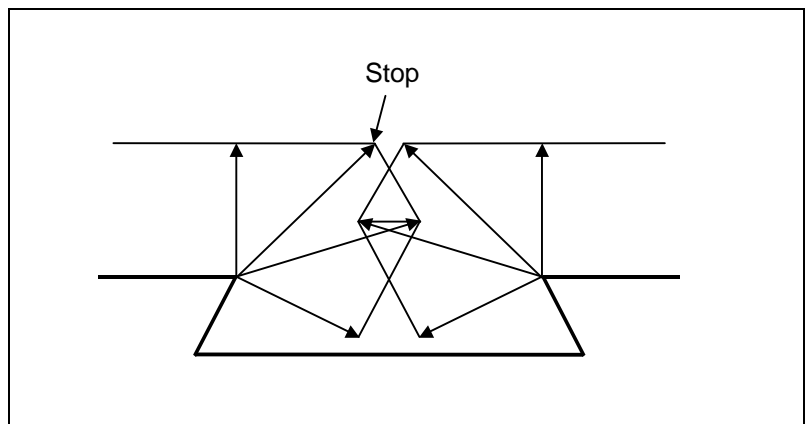
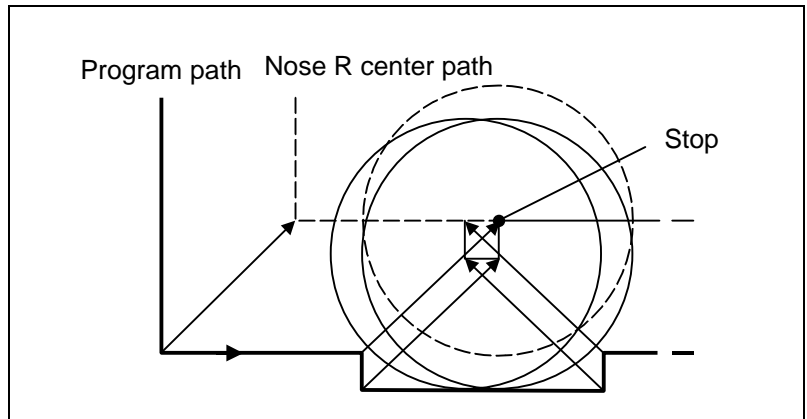


## 12. Tool Offset Functions

### 12.4 Tool nose radius compensation

- (c) When the program advance direction and the advance direction after compensation are reversed

In the following case, interference is still regarded as occurring even when there is actually no interference. When grooves which are narrower than the tool nose R diameter or which have parallel or widening walls are programmed.



#### 12.5 Programmed offset input; G10, G11



##### Function and purpose

The amount of tool offset and workpiece offset can be set or changed by the G10 command. When commanded with absolute values (X,Z,R), the commanded offset amounts serve as the new amounts; when commanded with incremental values (U,W,C), the new offset amounts are equivalent to the commanded amounts plus the current offset amount settings.



##### Command format

##### (1) Workpiece offset input (L2)

<b>G10 L2 P_X_(U_)Z_(W_);</b>	
P	Offset number
X	X-axis offset amount (absolute)
U	X-axis offset amount (incremental)
Z	Z-axis offset amount (absolute)
W	Z-axis offset amount (incremental)

##### (2) Tool length offset input (L10)

<b>G10 L10 P_X_(U_)Z_(W_)R_(C_)Q_;</b>	
P	Offset number
X	X-axis offset amount (absolute)
U	X-axis offset amount (incremental)
Z	Z-axis offset amount (absolute)
W	Z-axis offset amount (incremental)
R	Tool nose radius compensation amount (absolute)
C	Tool nose radius compensation amount (incremental)
Q	Hypothetical tool nose point

##### (3) Tool nose wear offset input (L11)

<b>G10 L11 P_X_(U_)Z_(W_)R_(C_)Q_;</b>	
P	Offset number
X	X-axis offset amount (absolute)
U	X-axis offset amount (incremental)
Z	Z-axis offset amount (absolute)
W	Z-axis offset amount (incremental)
R	Tool nose radius compensation amount (absolute)
C	Tool nose radius compensation amount (incremental)
Q	Hypothetical tool nose point

##### (4) When there is no L command with tool length offset input (L10) or tool nose wear offset input (L11)

Tool length offset input command : P = 10000 + offset number  
 Tool nose wear offset input : P = offset number

##### (5) Offset input cancel

<b>G11;</b>
-------------



### Detailed description

- (1) The following table shows the offset numbers and the setting ranges of the hypothetical tool nose points.

Add ress	Significance of address	Setting range		
		L2	L10	L11
P	Offset number	0:External workpiece offset 1:G54 workpiece offset 2:G55 workpiece offset 3:G56 workpiece offset 4:G57 workpiece offset 5:G58 workpiece offset 6:G59 workpiece offset	When L command is present: 1 ~ Max. number of offset sets  When L command is not present: 10001 ~ 10000 + Max. number of offset	When L command is/is not present: 1 ~ Max. number of offset sets
Q	Hypothetical tool nose point	—	0 ~ 9	

- (Note1)** The maximum number of offset sets for P (offset number) with tool offset input (L10 or L11) is up to a total of 80 with the addition of options.  
(The No. of sets will differ according to the model so check the specifications.)

- (2) The setting range for the offset amount is given below.  
Program error (P35) occurs for any value not listed in the table after command unit conversion.  
With an incremental value command, the setting range for the offset amount is the sum of the present setting and command value.

- (a) When using E60

Input setting unit	Tool length offset amount		Tool wear offset amount	
	Metric system	Inch system	Metric system	Inch system
#1003 = B	±999.999 (mm)	±99.9999 (inch)	±99.999 (mm)	±9.9999 (inch)

- (b) When using E68

Input setting unit	Tool length offset amount		Tool wear offset amount	
	Metric system	Inch system	Metric system	Inch system
#1003 = B	±9999.999 (mm)	±999.9999 (inch)	±999.999 (mm)	±99.9999 (inch)
#1003 = C	±999.9999 (mm)	±99.99999 (inch)	±99.9999 (mm)	±9.99999 (inch)



## Precautions and Restrictions

- (1) Offset amount setting range check  
The maximum value of the wear offset amount and the maximum additional value for the wear offset input check respectively take precedence for a single-time offset amount in the maximum value and incremental value command of the wear offset amount, and when an amount greater than these values has been commanded, program error (P35) results.
- (2) G10 is an unmodal command and is valid only in the commanded block.
- (3) Offset input can be performed similarly for the third axis but even when the C axis has been designated as the third axis, address C is treated as an incremental command value of the tool nose radius in the L10 or L11 command.
- (4) If an illegal L number or tool compensation No. is commanded, program error (P172 or P170) will result, respectively.
- (5) When the P command in the workpiece offset input is a command is omitted, it will be handled as the currently selected workpiece offset input.
- (6) Program error (P35) results when the offset amount exceeds the setting range.
- (7) X, Z and U, W are input together in a single block but when an address that commands the same offset input (X, U or Z, W) is commanded, the address which is input last is valid.
- (8) Offset will be input if even one address following G10 L(2/10/11) P\_ is commanded. Program error (P33) results when not even a single command has been assigned.

**(Example)** G10 L10 P3 Z50. ;

↓

[Tool length data]	
#	Z
3	50.000

Input as per left

- (9) Decimal points are valid for offset amounts.
- (10) G40 ~ G42 are ignored when they have been commanded in the same block as G10.
- (11) Do not command G10 in the same block as fixed cycles and subprogram call commands. This will cause malfunctioning and program errors.
- (12) When a T command has been issued in the same block as G10 with parameter "#1100 Tmove0" as zero, the offset will be accomplished in the following block.
- (13) For the multiple C-axis system, both C axis workpiece offsets are rewritten with the workpiece offset input.

### 12.6 Tool life management II



#### Function and purpose

Tool life management divides the tools being used into several groups, and manages the life (usage time, No. of uses) of the tools in each group. When the life is reached, a similar spare tool in the same group will be selected in order. This tool life management function with spare tools allows unmanned operation over a long time.

(1) Number of life-managed tools	Maximum 80 tools
(2) Number of groups	Maximum 80 groups
(3) Group No.	1 to 9999
(4) Number of tools in one group	Maximum 16 tools
(5) Life time	0 to 999999 minutes (approx. 16667 hours)
(6) Number of lives	0 to 999999 times

The tool life management data can be set from the NC program or from the Tool Life Management screen.

Refer to the Operation manual for the method of setting from the Tool Life Management screen. When using the NC program, register the data with the same method as program compensation input.



#### Command format

<b>G10 L3 ;</b>	<b>Life management data registration start</b>	
<b>P_ L_ N_ ;</b>	<b>Registration of group No., life per tool, method</b>	} <b>First group</b>
<b>T_ ;</b>	<b>Registration of tool No.;</b>	
<b>T_ ;</b>		
<b>P_ L_ N_ ;</b>	<b>Registration of next group No., life per tool, method</b>	
<b>T_ ;</b>	<b>Registration of tool No.;</b>	
<b>T_ ;</b>		
<b>G11 ;</b>	<b>End of life management data registration</b>	
<b>P</b>	Group No., (1 to 9999)	
<b>L</b>	Life per tool (0 to 999999 minutes, or 0 to 999999 times)	
<b>N</b>	Method (0:Time management, 1:No. of times management)	
<b>T</b>	Tool No.; the spare tool is selected in the order registered here. (Tool number 1 to 999999. Compensation No. 1 to 80) Tn follows the specifications.	



#### Precautions and restrictions

- (1) The tool life data is registered by executing the above program in the memory or MDI mode.
- (2) When the above program is executed, all data (group No., tool No., life data) registered previously will be deleted. The registered data is held even when the power is turned OFF.
- (3) The group No. designated with P does not have to be consecutive, but it should be set in ascending order if possible. This will make monitoring on the screen easier. The group No. cannot be commanded in duplicate.
- (4) If the life data L\_ is omitted, the life data for that group will be "0". If N\_, which designates the method, is omitted, the method for that group will follow the base specification parameter "#1106 Tcount".
- (5) Programming with a sequence No. is not possible between G10 L3 and G11.
- (6) If the usage data count valid signal (Y2CA) is ON, G10 L3 cannot be commanded. (P177 LIFE COUNT ACT)



#### Example of program

##### (1) Format

:	
T□□□□99 ;	Start use of □□□□ group tool
:	
:	
T□□□□88 ;	Cancel □□□□ group tool offset (Equivalent to TΔΔ00: ΔΔis number of tool being used).
:	
:	
M02 (M30) ;	End of machining program

##### (2) Actual example

:	
T0119 ;	Start use of group 01 tool
:	
:	
T0188 ;	Cancel group 01 tool offset If the number of the tool being used is 17, this is equivalent to T1700.
:	
:	
T0609 ;	Selects tool No. 06 and offset No. 09. * Life management is not carried out for tool 06.
:	
:	
T0600 ;	Cancel of group 06 tool offset
:	
:	
T0299 ;	Start use of group 02 tool
:	
:	
T0199 ;	Start use of group 01 tool If the selected tool has several compensation numbers, the second compensation No. is selected.





### Example of tool selection operation (When one tool has several compensation numbers)

- (1) To use several compensation numbers with one tool, select the next compensation No. for each T□□□□99 command.
- (2) If T□□□□99 is commanded for more times than the number of registered compensation numbers, the last compensation No. will be selected, and the operation will continue. (Refer to following.)

<u>Registration to group 1</u>	<u>Program</u>	<u>Tool selection</u>
T1701	T0199 ;	Equivalent to T1701
:		
T1702	T0199 ;	Equivalent to T1702
:		
T1703	T0199 ;	Equivalent to T1703
:		
T2104	T0199 ;	Equivalent to T1703
	:	:
(Group 1)	:	:(Hereafter, same until tool 17 reaches life.)

- (3) If the above program is executed after resetting with M02/M30, or by resetting with external reset, the selection will be made from the head compensation No.

#### 12.6.1 Counting the tool life



##### Function and purpose

The tool life can be counted with the time-count type or No. of uses-count type. The count method and timing for the No. of uses-count type can be changed to type 2 with the parameter setting (#1277 ext13/bit0).

If the usage data is equivalent to or more than the life data as a result of the count up, a spare tool in the group will be selected with the next group selection command (T□□□□99), and the newly selected tool will be counted.

When all tools in the group have reached their lives and a spare tool cannot be selected, the count will continue.

##### (1) Counting the time of uses when the time-count type method is selected

In the cutting mode (G01, G02, G03, G31, G33, etc.), the time that the tool is used is counted with 100ms units.

The time is not counted during dwell, machine lock, miscellaneous function lock or dry run. Whether to count or not for a single block can be changed by setting the parameter.

- (Note)**
- The maximum value for the life is 999999 minutes.
  - The data is displayed as a minute unit on the Tool Life Management screen.

### (2) Counting the No. of uses when the No. of uses-count type method is selected

#### (a) Type 1 (#1277 ext13/bit0: 0)

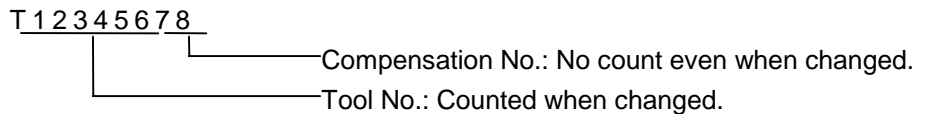
The No. of uses is counted when the number of the tool being used is changed with the tool selection command (T□□□□99) and when the program is in the cutting mode (excluding machine lock, miscellaneous function lock and dry run states).

The use is not counted if the cutting mode is not entered even once after the No. changes. Whether to count or not during single block can be changed with the parameter setting.

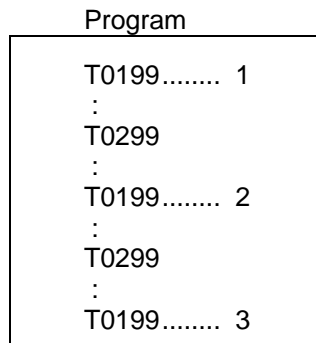
**(Note)** • The maximum value for the life is 999999 times.

- If only the compensation No. of the tool being used changes, a count will not be made.

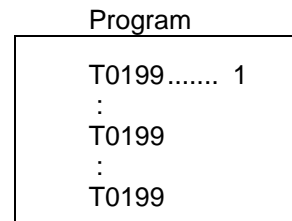
**Example:** When T code of tool being used is T12345678



<<Operation example>>



Group 01 has been used three times.



Group 01 has been used one time.

- \* The No. of uses is for one program execution. If the program is executed again after resetting, it will be counted.

#### (b) Type 2 (#1277 ext13/bit0: 1)

- Only the group used for cutting from when the machining program starts to when it is reset is incremented by "1". The count is made at the reset.
- If recount M is commanded, the group used up to that point will be incremented by "1" in the counter.

**(Note 1)** A count is not made in the machine lock, miscellaneous function lock or dry run states.

**(Note 2)** During single block, select whether to count with the parameter.

**(Note 3)** The maximum value of the life is 999999 times.

## 13. Program Support Functions

### 13.1 Fixed cycles for turning



#### Function and purpose

When performing rough cutting and other such operations during turning, these functions enable shapes normally commanded in several blocks to be commanded in a single block. In other words, they simplify the machining program. The following types of fixed cycles for turning are available.

G code	Function
G77	Longitudinal turning cycle
G78	Thread cutting cycle
G79	Face turning cycle



#### Command format

```
G77 X/U_ Z/W_ R_ F_ ;
```

(Same for G78, G79)



#### Detailed description

- (1) Fixed cycle commands are modal G codes and so they are valid until another command in the same modal group or a cancel command is issued.  
The following G code cancel commands are available.
  - G00, G01, G02, G03
  - G07,
  - G09,
  - G10, G11,
  - G27, G28, G29, G30,
  - G31,
  - G33, G34,
  - G37,
  - G92,
  - G52, G53,
  - G65,
- (2) The fixed cycle call becomes the movement command block call.  
The movement command block call calls the fixed cycle macro subprogram only when there is an axis movement command during the fixed cycle mode. It is executed until the fixed cycle is canceled.
- (3) A manual interrupt can be applied while a fixed cycle for turning (G77 ~ G79) is being executed. Upon completion of the interrupt, however, the tool must be returned to the position where the manual interrupt was applied and then the fixed cycle for turning should be resumed.  
If it is resumed without the tool having been returned, all subsequent operations will deviate by an amount equivalent to the manual interrupt.

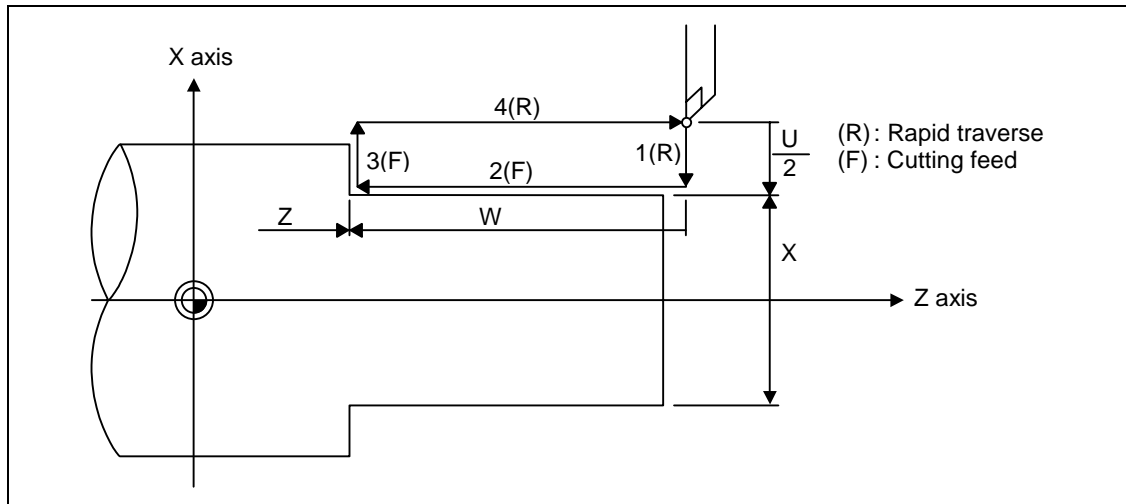
### 13.1.1 Longitudinal cutting cycle; G77



#### Straight thread cutting

This function enables continuous straight cutting in the longitudinal direction using the following command.

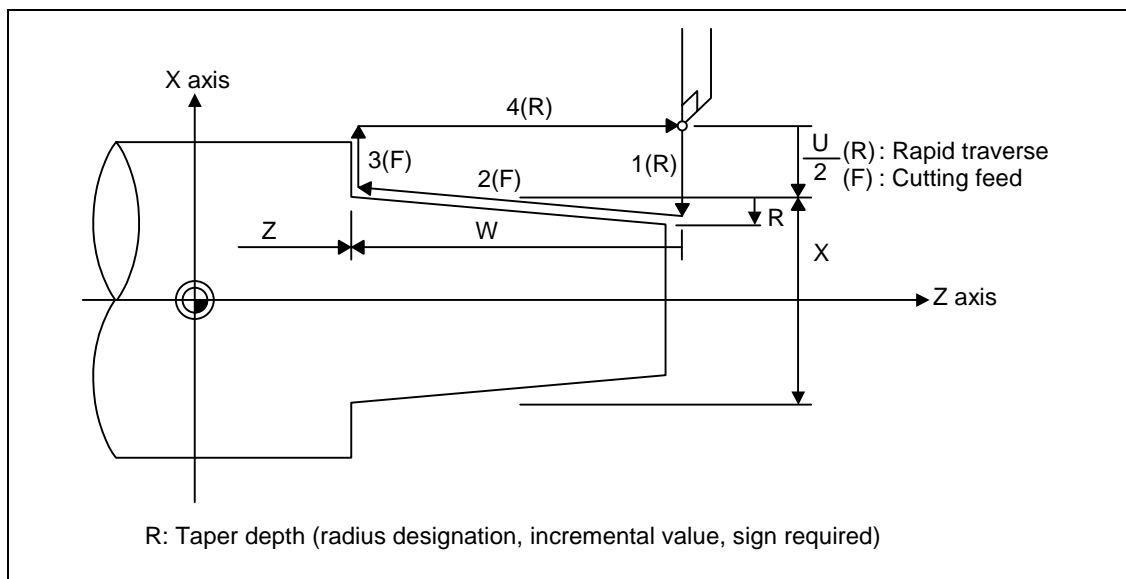
G77 X/U\_ Z/W\_ F\_ ;



#### Taper thread cutting

This function enables continuous taper cutting in the longitudinal direction using the following command.

G77 X/U\_ Z/W\_ R\_ F\_ ;

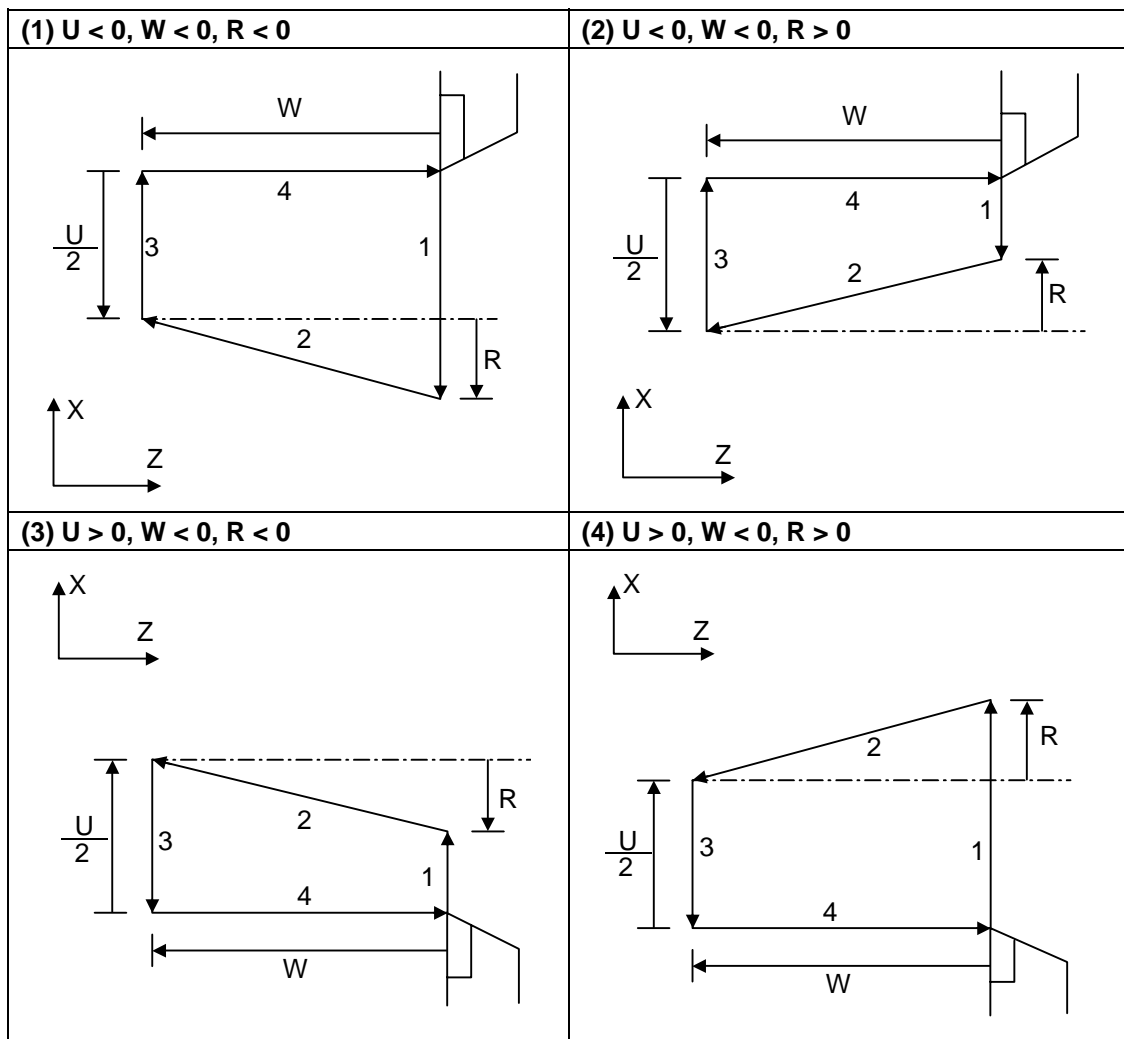


With a single block, the tool stops at the end points of operations 1, 2, 3 and 4.

# 13. Program Support Functions

## 13.1 Fixed cycles for turning

Depending on the U, W and R signs, the following shapes are created.



Program error (P191) results with shapes (2) and (3) unless the following condition is satisfied.

$$|u/2| \geq |r|$$

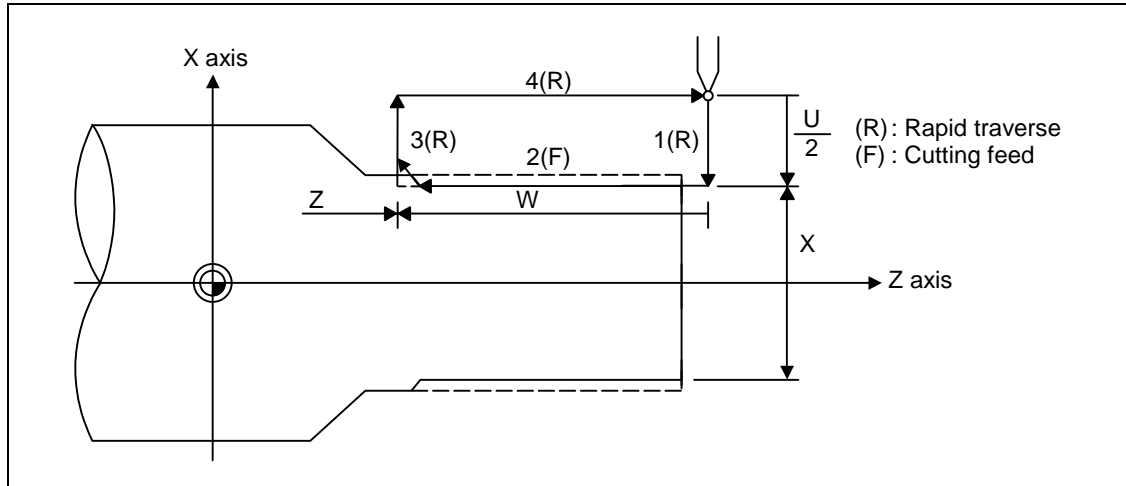
### 13.1.2 Thread cutting cycle; G78



#### Straight cutting

This function enables straight thread cutting using the following command.

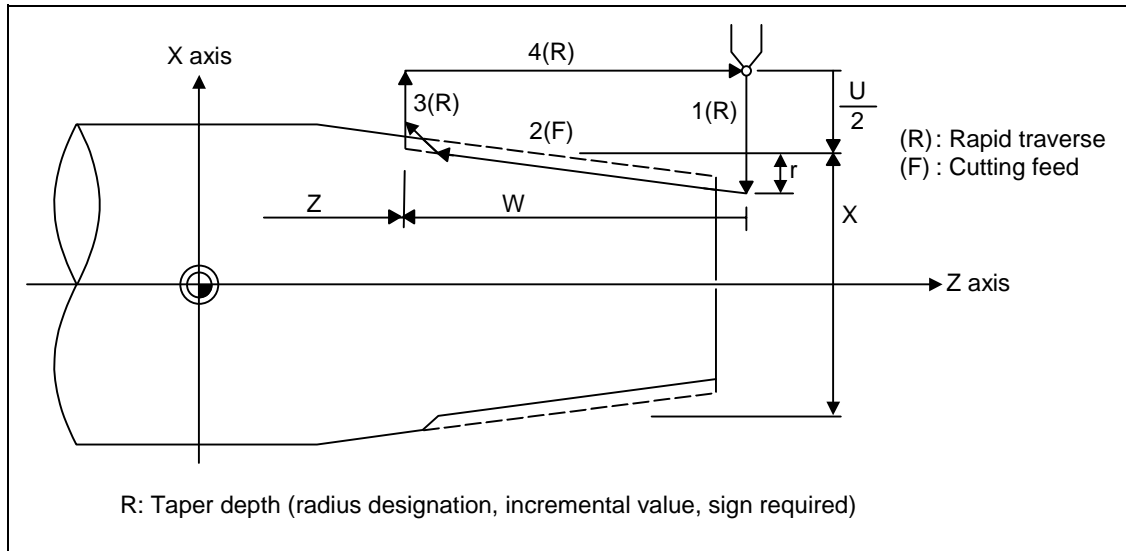
G78 X/U\_ Z/W\_ F/E\_ Q\_ ;



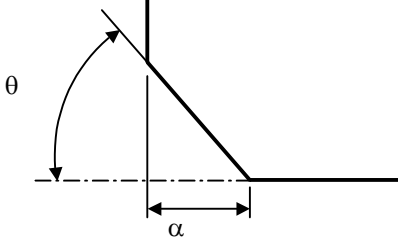
#### Taper cutting

This function enables taper thread cutting using the following command.

G78 X/U\_ Z/W\_ R\_ F/E\_ Q\_ ;



### Details for chamfering

	<p><math>\alpha</math> : Chamfering amount of thread If the thread lead is assumed to be L, then the parameter can be set in 0.1L units from 0 to 12.7L.</p> <p><math>\theta</math> : Chamfering angle of thread The parameter can be set in 1° units from 0 to 89°</p>
---	---

With a single block, the tool stops at the end points of operations 1, 3 and 4.

When the feed hold function is applied during a thread cutting cycle, automatic operation will stop if no thread is then being cut, or if applied in the interval from the execution of the thread cutting command to the axis movement. If thread cutting is proceeding when the function is applied, the operation stops at the next movement completion position (completion of operation 3) of the thread cutting.

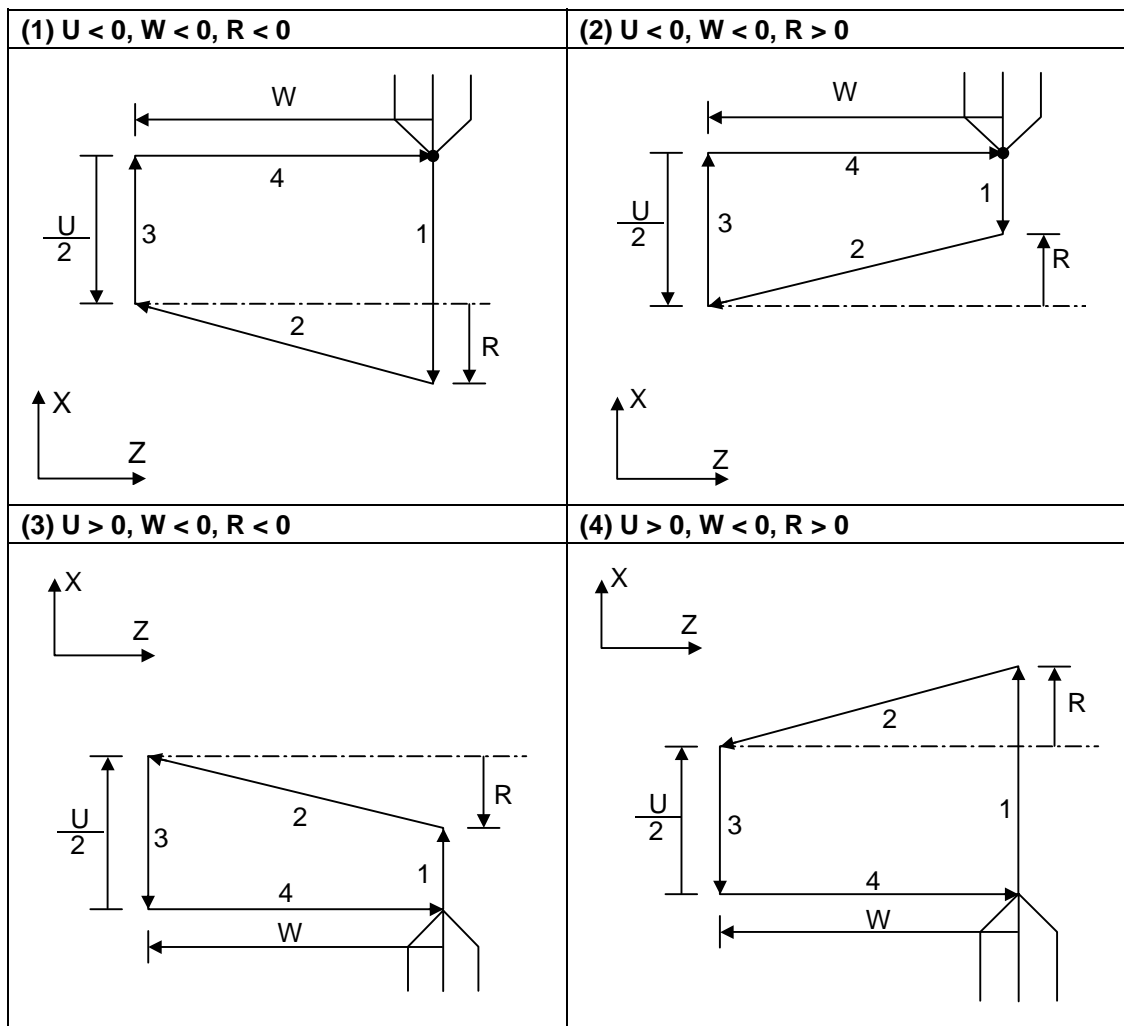
The dry run valid/invalid status does not change during thread cutting.



# 13. Program Support Functions

## 13.1 Fixed cycles for turning

Depending on the U, W and R signs, the following shapes are created.



Program error (P191) results with shape (2) and (3) unless the following condition is satisfied.

$$|u/2| \geq |r|$$

- (Note 1)** The thread cutting start shift angle is not a modal. If there is no Q command with G78, this will be handled as "Q0".
- (Note 2)** If a value exceeding 360.000 is command for G78 Q, this will be handled as "Q360.000".
- (Note 3)** G78 cuts one row with one cycle. To cut two rows, change the Q value, and issue the same command.

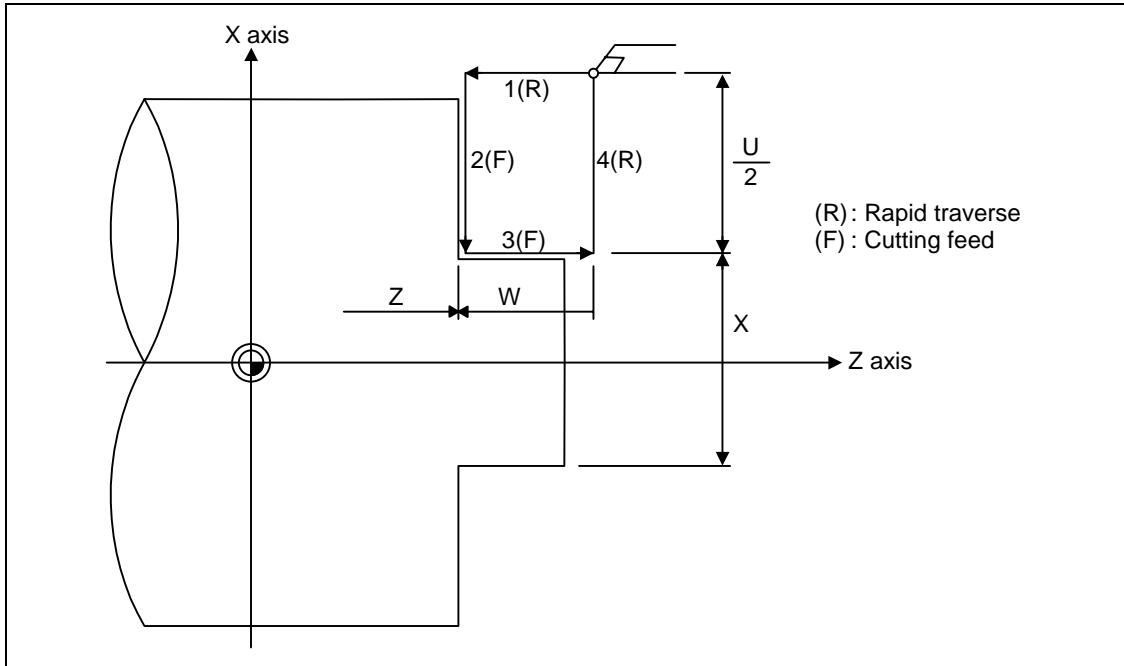
### 13.1.3 Face cutting cycle; G79



#### Straight cutting

This function enables continuous straight cutting in the face direction using the following command.

G79 X/U\_ Z/W\_ F\_ ;

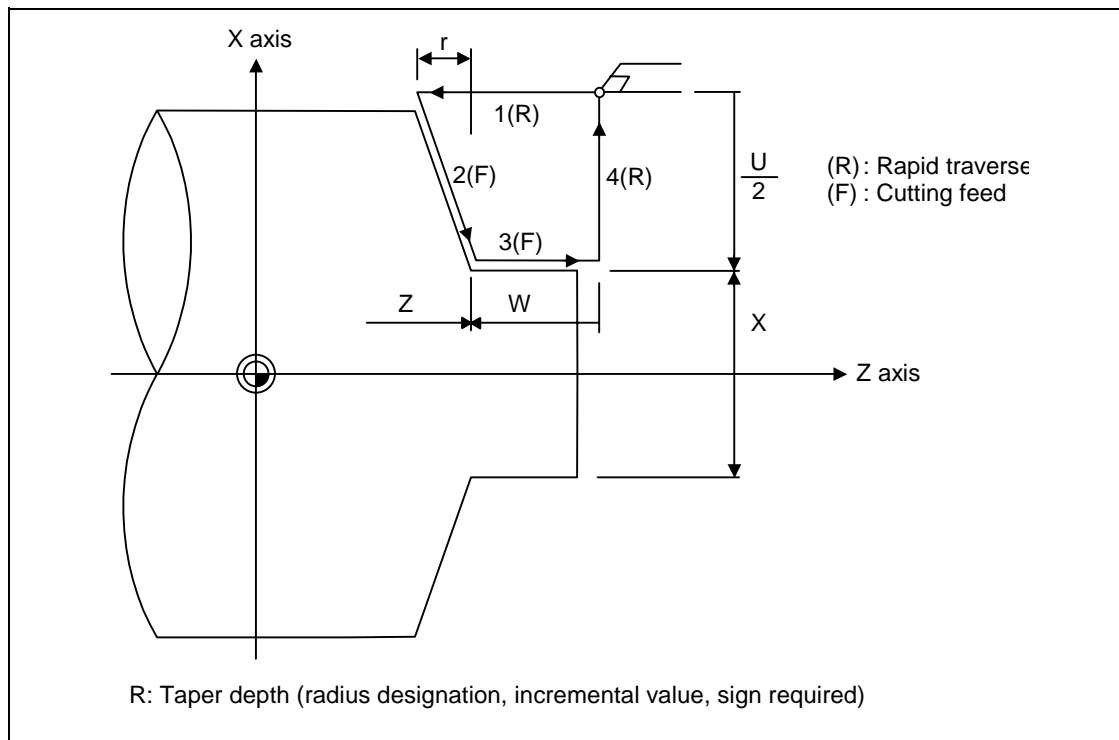




### Taper cutting

This function enables continuous taper thread cutting using the following command.

G79 X/U\_Z/W\_R\_F\_;

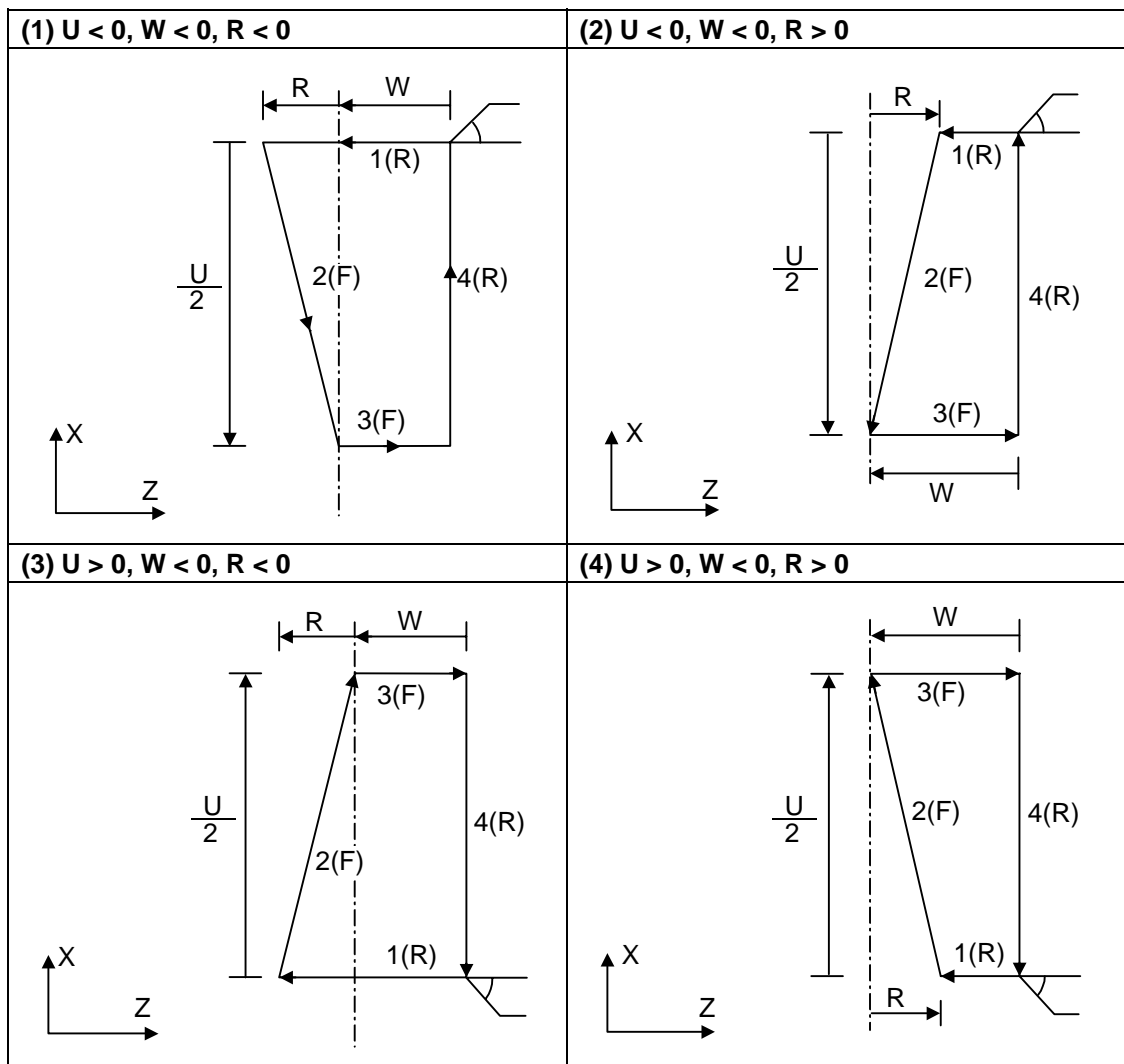


With a single block, the tool stops at the end points of operations 1, 2, 3 and 4.

# 13. Program Support Functions

## 13.1 Fixed cycles for turning

Depending on the U, W and R signs, the following shapes are created.



Program error (P191) results with shapes (2) and (3) unless the following condition is satisfied.

$$|w| \geq |r|$$

## 13. Program Support Functions

### 13.2 Fixed Cycle for Turning (MELDAS Special Format)

#### 13.2 Fixed Cycle for Turning (MELDAS Special Format)

[E68]



#### Function and purpose

When performing rough cutting and other such operations during turning, these functions enable shapes normally commanded in several blocks to be commanded in a single block. In other words, they simplify the machining program.

When using the MELDAS special format (#1265 ext01/bit1 ON), some addresses differ from the normal format. The sections that differ from the normal format are shown below.

Refer to section "13.1 Fixed cycle for turning " for details on the fixed cycles for turning.



#### Command format

G77 X(U)\_ Z(W)\_ I\_ F\_ ;  
(The same for G78)

G79 X(U)\_ Z(W)\_ K\_ F\_ ;



#### Detailed description

##### (1) Comparison of MELDAS special format and normal format

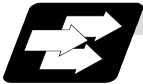
Some addresses in the MELDAS special format differ from the normal format.

Function	MELDAS special format	Normal format	Difference from normal format
Longitudinal turning cycle	G77 X Z I F ; or G77 U W I F ;	G77 X Z R F ; or G77 U W R F ;	Taper depth Designate with R → I
Thread cutting cycle	G78 X Z I F ; or G78 U W I F ;	G78 X Z R F ; or G78 U W R F ;	Taper depth Designate with R → I
Face turning cycle	G79 X Z K F ; or G79 U W K F ;	G79 X Z R F ; or G79 U W R F ;	Taper depth Designate with R → K

## 13. Program Support Functions

### 13.3 Compound Type Fixed Cycle for Turning Machining

#### 13.3 Compound Type Fixed Cycle for Turning Machining



##### Function and purpose

These functions enable prepared fixed cycle to be executed by commanding the corresponding program in a block.

The types of fixed cycles are listed below.

G code	Function	
G70	Finishing cycle	Compound type fixed cycle for turning machining I
G71	Longitudinal rough cutting cycle (finished shape chamfering)	
G72	Face rough cutting cycle (finished shape chamfering)	
G73	Formed material rough cutting cycle	
G74	Face cut-off cycle	Compound type fixed cycle for turning machining II
G75	Longitudinal cut-off cycle	
G76	Compound thread cutting cycle	

If, with any of the above functions for the compound type fixed cycle for turning machining I (G70 ~ G73), the finished shape program has not been entered in the memory, the function cannot be used.

G71 and G72 have the type which machines pocket shape (compound type fixed cycle for turning machining type II). This is valid for only E68 specification.



##### Command format

G70	A_P_Q_;
G71	U_R_;
G71	A_P_Q_U_W_F_S_T_;
G72	W_R_;
G72	A_P_Q_U_W_F_S_T_;
G73	U_W_R_;
G73	A_P_Q_U_W_F_S_T_;
G74	R_;
G74	X(U)_Z(W)_P_Q_R_F_;
	(Same for G75)
G76	P_R_;
G76	X(U)_Z(W)_P_Q_R_F_;



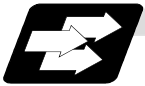
##### Detailed description

- (1) The A, P and Q commands of the compound type fixed cycle for turning machining I are described below.
  - (a) When the A command is not present, P and Q in the program now being executed are called.  
When the A command is present and the P command is not present, the head block of the program designated by the A command is treated as the P command.
  - (b) When the Q command is not present, operation continues until the M99 command is located.  
When both the Q and M99 commands are not present, operation continues until the final block in the finished shape program.

# 13. Program Support Functions

## 13.3 Compound Type Fixed Cycle for Turning Machining

### 13.3.1 Longitudinal rough cutting cycle; G71



#### Function and purpose

This function calls the finished shape program and, while automatically calculating the tool path, performs rough cutting in the longitudinal direction.



#### Command format

G71 Ud Re Hh ; (May be omitted when using value set in parameters) G71 Aa Pp Qq Uu Ww Ff Ss Tt ;	
<H0: Used for finished shape with no pockets>	<H1: Mainly used for finished shape with pocket>
<p>G71 Ud Re H0 ; G71 Pp Qq · · ;</p>	<p>G71 Ud Re H1 ; G71 Pp Qq · · ;</p>
Ud	Cut amount (modal) ······ Reversible parameter ("G71 THICK") Unit ···· μm/ 1/10000 inch radius value command
Re	Retract amount (modal) ······ Reversible parameter ("G71 PULL UP") Unit ···· μm/ 1/10000 inch radius value command
Hh	Pocket machining (modal) ··· Reversible parameter ("G71 POCKET") 0: Select this only for finished shapes that have no indent section (pocket). * The cutting section will retract 45° per cycle, and finally will trace the finished shape. 1: This can be selected whether or not there is an indent (pocket) in the finished shape. * The cutting section will trace the finished shape each cycle. Parameter cutting method [Control parameter] "#8110 G71/G72 POCKET" 0: Do not machine pocket 1: Machine pocket Pocket machining is selected with the parameter "#8110 G71/G72 POCKET" or with Hh. However, if the following parameters are ON, pocket machining will be automatically selected according to the number of axes in the finished shape start block. <ext08/bit0=0> Follows parameter "#8110 G71/G72 POCKET" or Hh designation. <ext08/bit0=1> When X and Z axes are commanded in finished shape start block: Pocket machining (equivalent to H1) When only X or Z axis is commanded in finished shape start block: Equivalent to H0 <b>(Note)</b> When ext08/bit0 = 1, the tool will retract in the X axis direction during cutting. (Refer to section "Cutting method and retract amount" for details.)

## 13. Program Support Functions

### 13.3 Compound Type Fixed Cycle for Turning Machining

Aa	Finished shape program No. (Program being executed when omitted) If the A command is omitted, the P and Q commands in the program being executed will be used. If A is omitted, the program will be executed in the order of G71 and then the program following Qq (finished shape end sequence No.)
Pp	Finished shape start sequence No. (Head of program when omitted)
Qq	Finished shape end sequence No. (To end of program when omitted) Even when Q is commanded, if M99 is commanded first the program will continue to M99.
Uu	X axis direction finishing allowance (When omitted, X axis direction finishing allowance is handled as 0.) Unit ··· μm/ Follows 1/10000ich diameter/radius value command (#1019 dia).
Ww	Z axis direction finishing allowance (When omitted, Z axis direction finishing allowance is handled as 0.) Unit ··· μm/ Follows 1/10000ich radius value command
Ff	Cutting speed (Follows cutting speed (modal) before G71 when omitted)
Ss,Tt	Spindle command, tool command



# 13. Program Support Functions

## 13.3 Compound Type Fixed Cycle for Turning Machining



### Cutting shape

#### (1) Finished shape start block and finished shape end block

G71 A<sub>a</sub> P<sub>p</sub> Q<sub>q</sub>

The finished shape start block and finished shape end block are designated with the above command.

There may be up to 50 finished shape blocks, including the blocks inserted with corner chamfer, corner R command and nose R command. A program error (P202) will occur if the number of blocks is exceeded.

<p>&lt;When pocket machining is OFF&gt;</p>	<p>&lt;When pocket machining is ON&gt;</p>
<p>&lt;When creating finished shape program in machining program other than that being currently executed&gt;</p> <p>G71 A<sub>a</sub> P<sub>p</sub> Q<sub>q</sub></p> <p>a: Finished shape program No.  p: Finished shape start sequence No.  (From head of finished shape program when omitted)  q: Finished shape end sequence No.  (To end of finished shape program or M99 when omitted)</p>	<p>&lt;When creating finished shape program in program currently being executed&gt;</p> <p>G71 P<sub>p</sub> Q<sub>q</sub></p> <p>p: Finished shape start sequence No.  q: Finished shape end sequence No.</p>
<p>Program currently being executed</p> <p>G0 X80.0 Z75.0 T0101 ; ← (a)  G71 U10. R3. ; ← (b)  G71 <b>A100 P10 Q20</b> U3. W1.5 F500 S1500 ; ←(c)  G70 A100 P10 Q20 ; ← (d)</p> <p>Shape program</p> <p><b>O100</b>  G28 XZ ;  <b>N10</b> G0 X15.0 Z65.0 ;  G1 Z55. F450 ;  G1 X30.0 ;  G3 X40.0 Z50.0 R5.0 ;  :  G1 Z42.0 ;  G1 X50.0 Z35.0 ;  <b>N20</b> G1 X60.0 ;  N30 G0 X13.0 Z68.0 ;  :  Program execution order (a) (b) (c) (d)</p> <p>Finished shape program</p>	<p>Program currently being executed</p> <p>G0 X80.0 Z75.0 T0101 ; ← (a)  G71 U10. R3. ; ← (b)  G71 <b>P10 Q20</b> U3. W1.5 F500 S1500 ; ←(c)</p> <p><b>N10</b> G0 X15.0 Z65.0 ;  G1 Z55. F450 ;  G1 X30.0 ;  G3 X40.0 Z50.0 R5.0 ;  G1 Z42.0 ;  G1 X50.0 ;  G1 X55.0 Z35.0 ;  <b>N20</b> G1 X60.0 ;  G70 A100 P10 Q20 ; ← (d)</p> <p>Finished shape program</p> <p>Program execution order (a) (b) (c) (d)</p> <p>* G70 command: Finishing cycle  The workpiece is finished following the finishing program, and then the axis returns to the G70 command position.</p>

## 13. Program Support Functions

### 13.3 Compound Type Fixed Cycle for Turning Machining

#### (2) Machining of open section and machining of mid-section

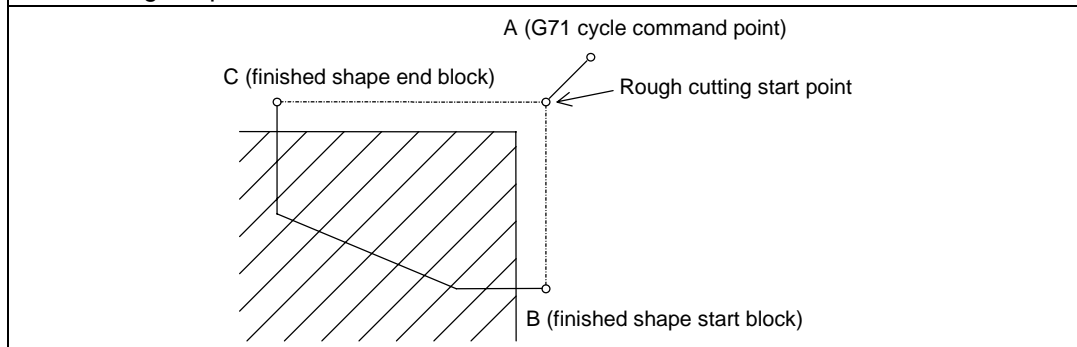
The rough cutting cycle machining includes that in which one end of the workpiece is open, and that in which machining starts at a mid-section of the workpiece.

Machining of open section ..... Possible when pocket machining is either OFF or ON

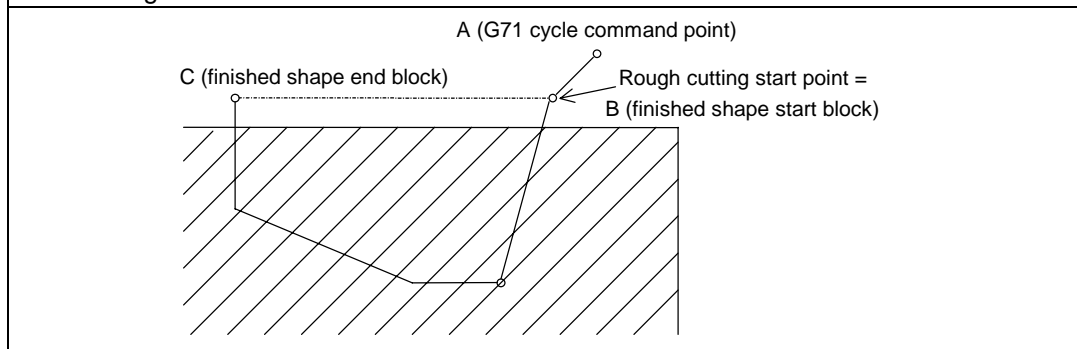
Machining of mid-section ..... Possible only when pocket machining is ON

(Entire rough cutting cycle is handled as one pocket.)

#### <Machining of open section>



#### <Machining of mid-section>



# 13. Program Support Functions

## 13.3 Compound Type Fixed Cycle for Turning Machining

### (3) Rough cutting direction

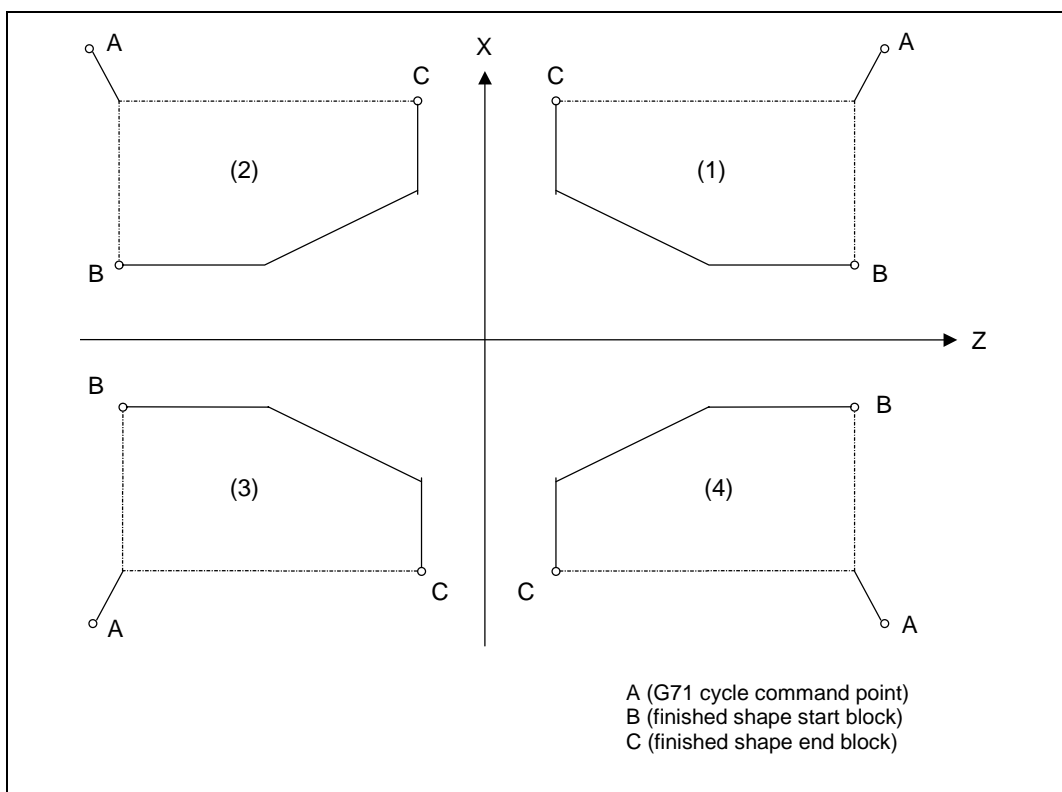
<Rough cutting direction when pocket machining is OFF>

Select one of the following.

- Automatically determine according to finished shape (#1273 ext09/bit2=0)  
The rough cutting direction is determined in the following manner according to the finished shape. The finished shape program is A → B → C.

C (finished shape end block) X axis > B (finished shape start block) Z axis	(1) or (2)
C (finished shape end block) X axis < B (finished shape start block) Z axis	(2) or (4)
C (finished shape end block) Z axis > B (finished shape start block) Z axis	(2) or (3)
C (finished shape end block) Z axis < B (finished shape start block) Z axis	(1) or (4)

\* A program error (P203) will occur if C (finished shape end block) X axis = B (finished shape start block) X axis, or C (finished shape end block) Z axis = B (finished shape start block) Z axis.



- Designate with program (#1273 ext09/bit2=1)  
The finishing allowance direction follows the sign of the finishing allowance designated in the program.

# 13. Program Support Functions

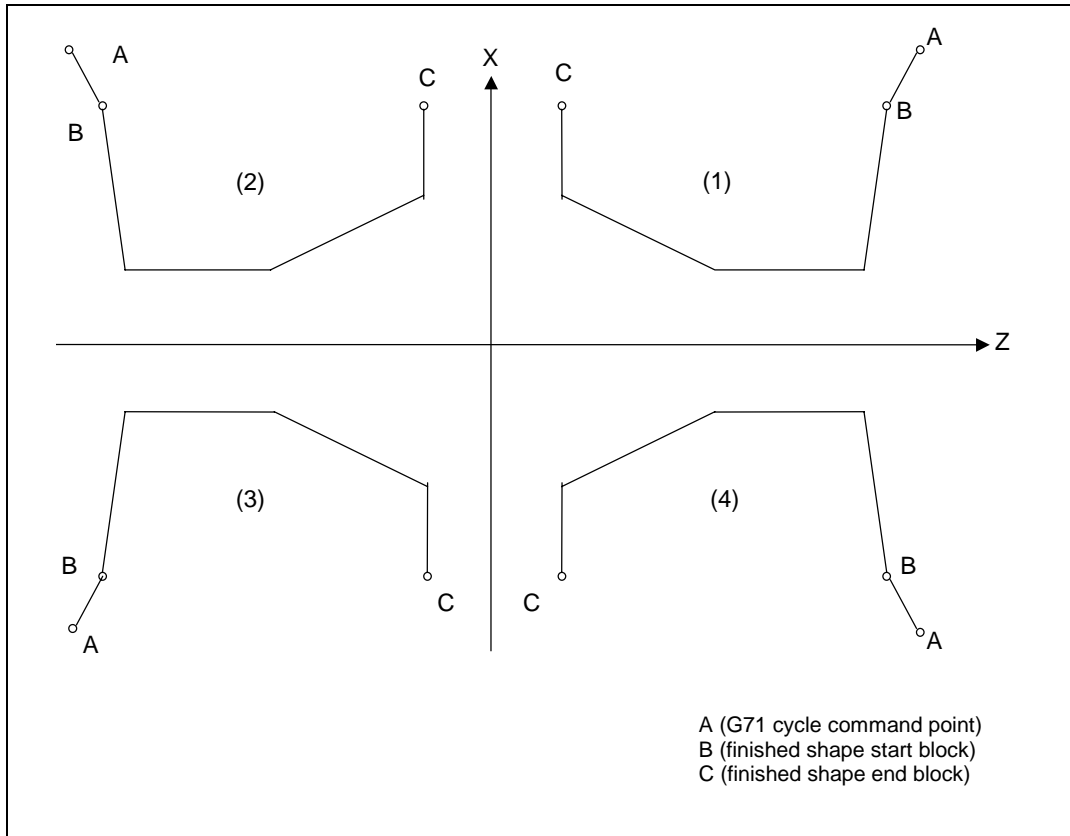
## 13.3 Compound Type Fixed Cycle for Turning Machining

<Rough cutting direction when pocket machining is ON>  
 Select one of the following.

- Automatically determine according to finished shape (ext09/bit2=0)  
 The rough cutting direction is determined in the following manner according to the finished shape. The finished shape program is A → B → C.

A (G71 cycle command point) X axis > B (finished shape start block) X axis, and B (finished shape start block) X axis ≤ C (finished shape end block) X axis	(1) or (2)
A (G71 cycle command point) X axis < B (finished shape start block) X axis, and B (finished shape start block) X axis ≥ C (finished shape end block) X axis	(3) or (4)
C (finished shape end block) Z axis > B (finished shape start block) Z axis	(2) or (3)
C (finished shape end block) Z axis < B (finished shape start block) Z axis	(1) or (4)

\* A program error (P203) will occur if C (finished shape end block) Z axis = B (finished shape start block) Z axis.



\* When setting the B point X axis at the base, the method is the same as the "<Rough cutting direction when pocket machining is OFF>".

- Designate with program (ext09/bit2=1)  
 The finishing allowance direction follows the sign of the finishing allowance designated in the program.

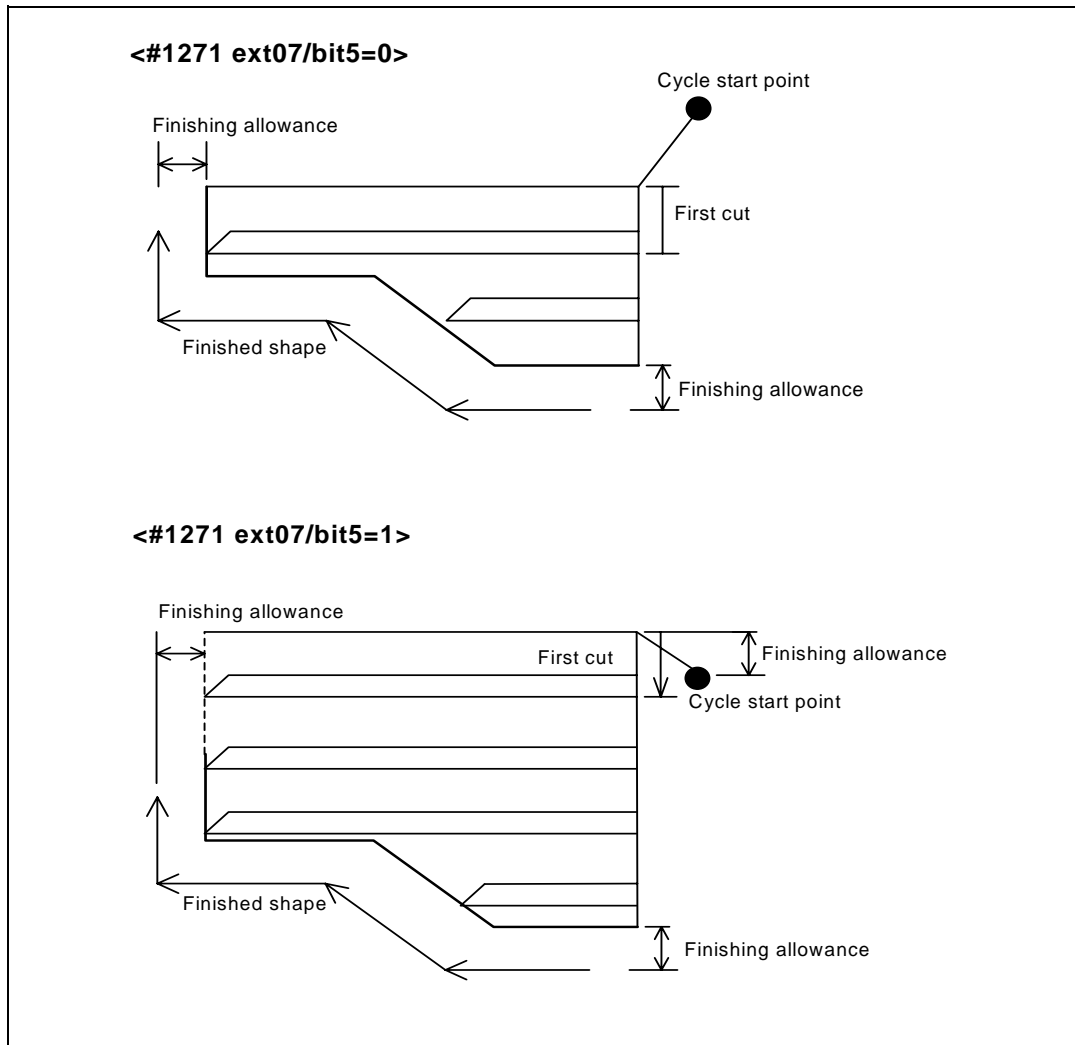
## 13. Program Support Functions

### 13.3 Compound Type Fixed Cycle for Turning Machining

#### (4) Finished shape

Selection of cut start position (#1271 ext07/bit5)

The cut start position is obtained from the final position of the finished shape program, but this can be changed to the cycle start point.



# 13. Program Support Functions

## 13.3 Compound Type Fixed Cycle for Turning Machining

<Finished shape in Z axis direction when pocket machining is OFF>

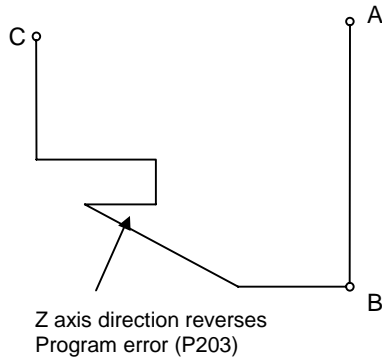
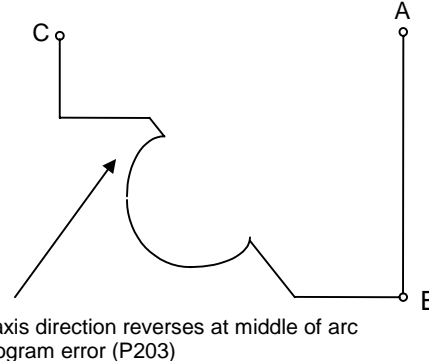
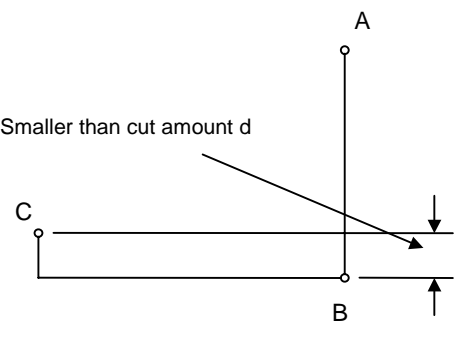
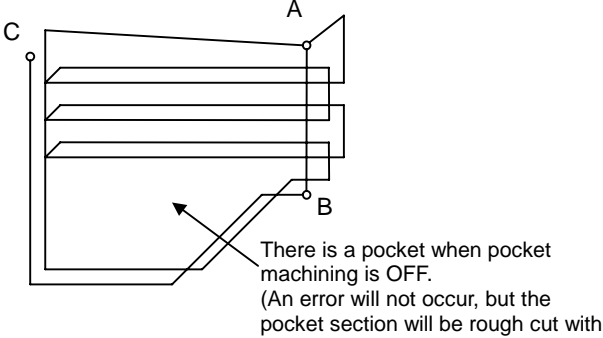
The Z axis direction finishing allowance must always change monotonously (only increment, or only decrement).

A program error (P203) will occur if the shape is illegal.

<Finished shape in X axis direction when pocket machining is OFF>

As a principle, the X axis direction finishing allowance must change monotonously (increment or decrement).

At sections that do not change monotonously, the rough cutting cycle will not be executed at the indent section (pocket), but the finishing allowance will be traced as commanded. Thus, if the pocket's X axis depth is not less than the cut amount, the cutting load will increase at the pocket section. Take care when programming.

<Example of illegal Z axis direction when pocket machining is OFF>	
 <p style="text-align: center;">Z axis direction reverses Program error (P203)</p>	 <p style="text-align: center;">Z axis direction reverses at middle of arc Program error (P203)</p>
<Example of illegal X axis direction when pocket machining is OFF>	
 <p style="text-align: center;">Smaller than cut amount d</p> <p style="text-align: center;">Program error (P203)</p>	 <p style="text-align: center;">There is a pocket when pocket machining is OFF. (An error will not occur, but the pocket section will be rough cut with one cut.)</p>

# 13. Program Support Functions

## 13.3 Compound Type Fixed Cycle for Turning Machining

<Finished shape in Z axis direction when pocket machining is ON>

The finished shape in the Z axis direction will always change monotonously (only increment, or only decrement).

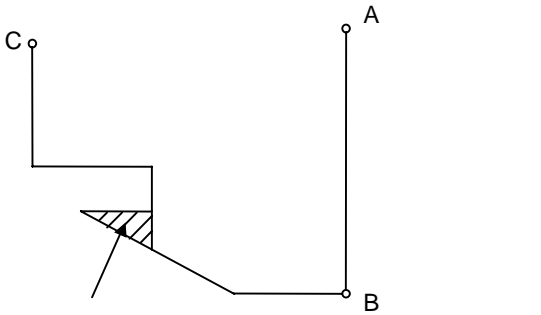
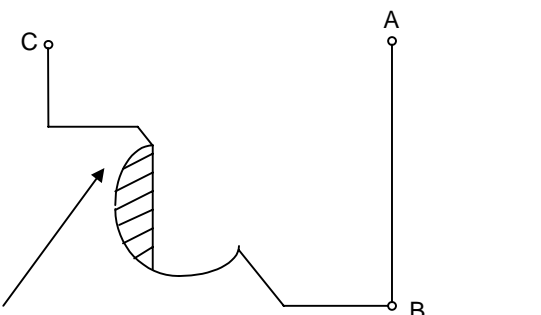
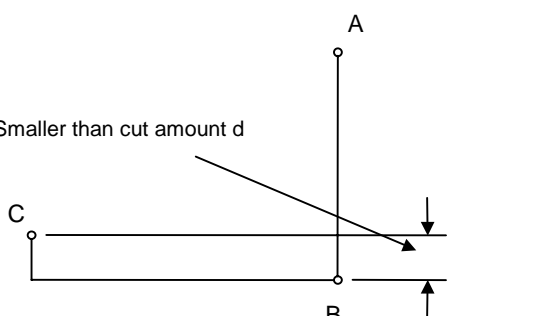
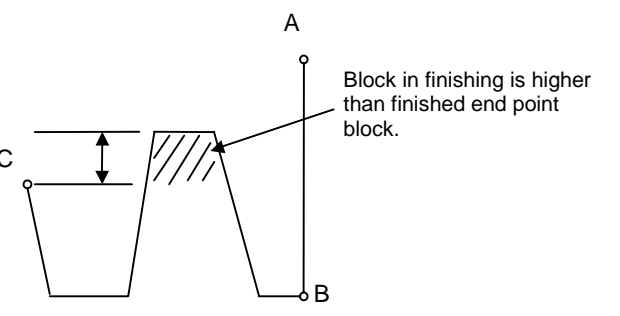
Sections that do not change monotonously will have a cover.

<Finished shape in X axis direction when pocket machining is ON>

As a principle, the finished shape in the X axis direction must change monotonously (increment or decrement). The direction must reverse only at the indent shape (pocket).

Make sure that the blocks in the finished shape do not exceed the C (finished shape end block) X axis position during pocket machining.

A program error (P203) if the interim blocks exceed the C (finished shape end block) X axis position.

<Example of section where Z axis direction does not change monotonously when pocket machining is ON>	
 <p>The Z axis direction reverses, so a finished shape with cover at shaded section is created. (The shaded section is not cut)</p>	 <p>The Z axis direction reverses in the middle of the arc, so a finished shape with cover at the shaded section is created. (The shaded section is not cut)</p>
<Example of illegal X axis direction when pocket machining is ON>	
 <p>Smaller than cut amount d</p> <p>Program error (P203)</p>	 <p>Block in finishing is higher than finished end point block.</p> <p>Program error (P203)</p>

## 13. Program Support Functions

### 13.3 Compound Type Fixed Cycle for Turning Machining

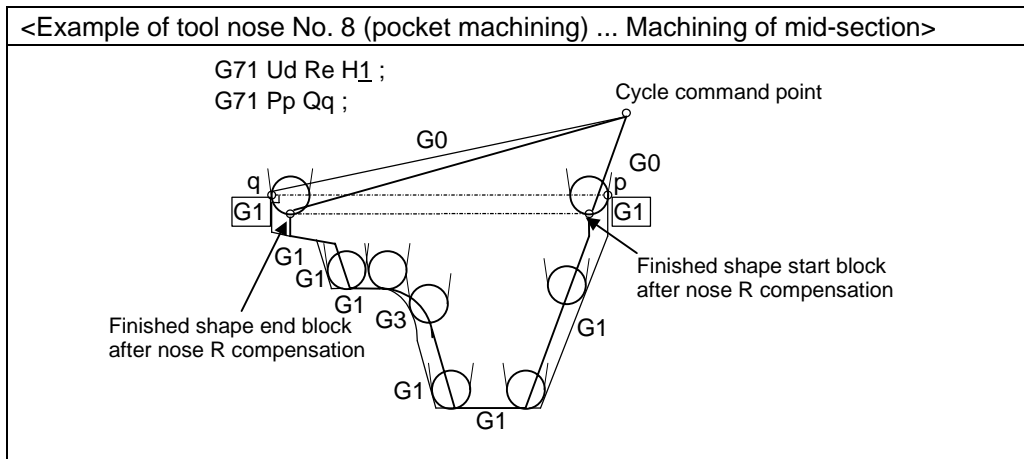
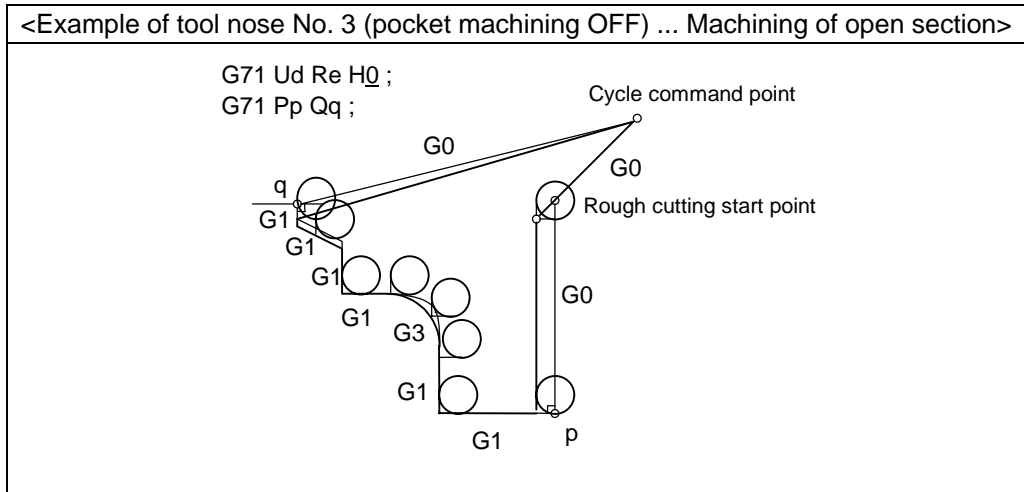
#### (5) Finished shape during nose R compensation

Selection of nose R compensation (#1271 ext07/bit6)

<#1271 ext07/bit6=0>

If there is a G71 command in nose R compensation, the G71 cycle command point will be at the position where nose R compensation is temporarily canceled. The finished shape will be created while nose R compensation is applied.

**(Note)** To machine a mid-section while pocket machining is ON, program so that the finished shape start block's X axis position and finished shape end block's X axis position are the same after nose R compensation is applied.



#### Precautions for machining mid-section

The mid-section will be machined when the finished shape start block's X axis position and finished shape end block's X axis position are the same after nose R compensation. Program so that the blocks enclosed with the square above move only the X axis (distance more than nose R diameter).

<#1271 ext07/bit6=1>

The shape for which nose R compensation is not executed in the finished shape program will be the rough cutting shape.



# 13. Program Support Functions

## 13.3 Compound Type Fixed Cycle for Turning Machining



### Cut amount

- (1) The cut amount is designated by Ud. However, it is possible to change the cut amount with each cutting pass by setting the cut change amount ( $\Delta d$ ) using a parameter.

Cut amount designation (ext07/bit7)

<ext07/bit7=0>

If the amount of one cut commanded in the program is deeper than the cutting depth of the finished shape ( $d < \Delta d$ ), a program error (P204) occurred.

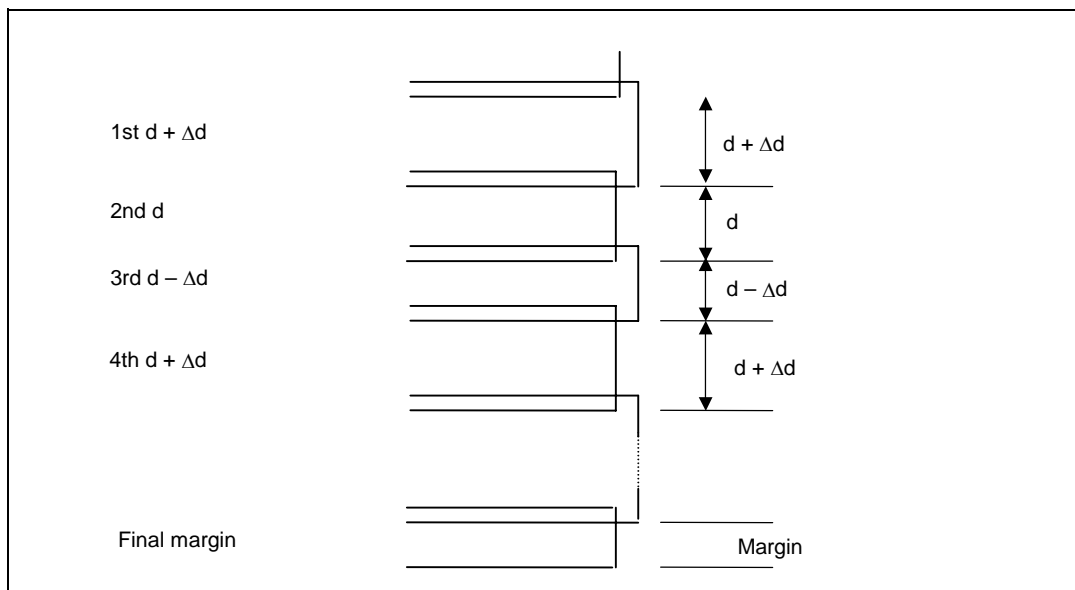
<ext07/bit7=1>

A program error will not occur even if the amount designated for one cut is deeper than the cutting depth of the finished shape, and the command will be executed as one cut.

Note that the amount of one cut must be commanded between 0 and 99.999mm. A program error will occur if the value is not within this range.

- (2) The end of the cutting is the cut amount, but if this value is smaller than the value set in the parameters, cutting will not be carried out, and instead finishing rough cutting will be executed.

Parameter	Cut amount (d)	[Machining parameter] "#8051 G71 THICK" 0 to 99.999mm
	Cutting change amount ( $\Delta d$ )	[Machining parameter] "#8017 G71 DELTA-D" 0 to 99.999mm
	Minimum cut amount for final cut	[Machining parameter] "#8016 G71 MINIMUM" 0 to 99.999mm



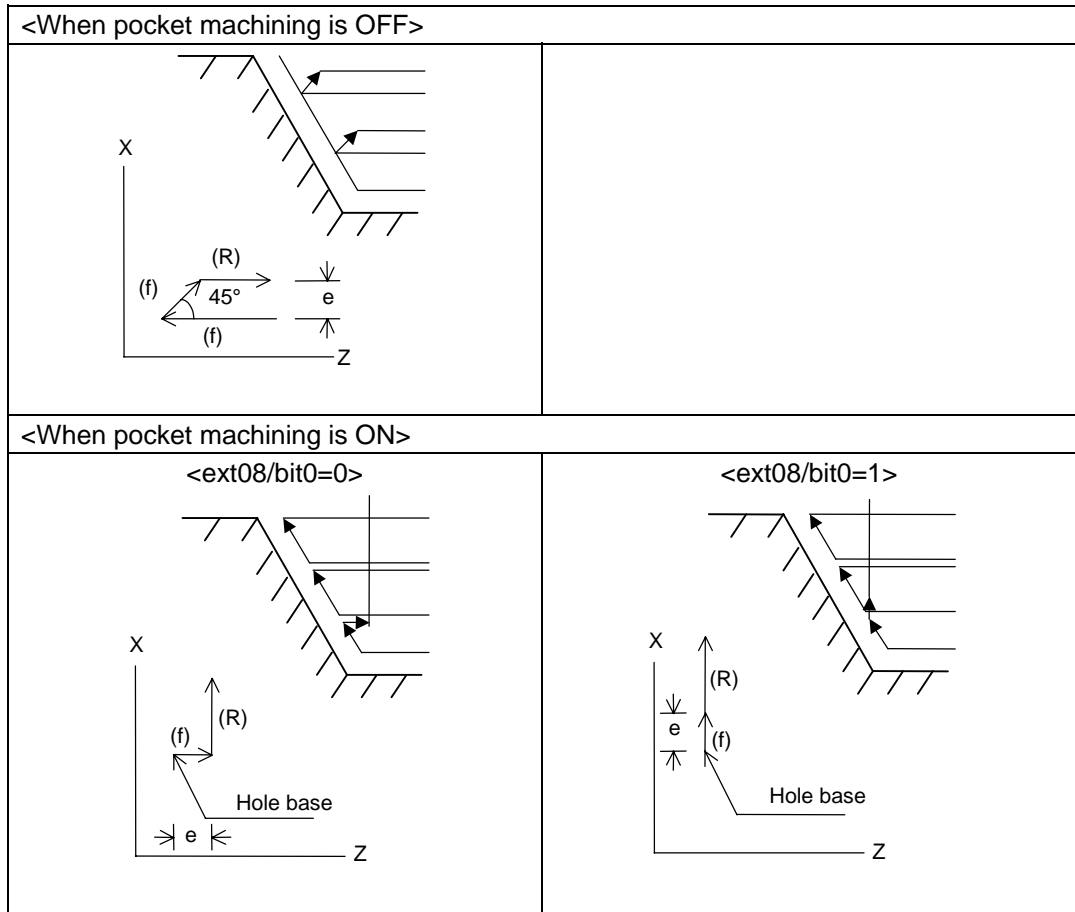
# 13. Program Support Functions

## 13.3 Compound Type Fixed Cycle for Turning Machining



### Cutting method and retract amount

- (1) The cutting method differs according to whether pocket machining is ON or OFF.  
 Pocket machining OFF ..... Retracts in 45° direction in respect to workpiece  
 Pocket machining ON ..... Traces shape
- (2) The retract amount is designated with Re, but the retract amount can also be set by designating the amount in the parameters.  
 Pocket machining OFF ..... Retracts in 45° direction in respect to workpiece  
 Parameter retract amount (e) [Machining parameter] "#8052 G71 PULL UP" 0 to 99.999mm



**(Note)** ext08/bit0 is also used to select whether to automatically determine pocket machining. (Refer to the explanation of "Hh" in Command format.)

# 13. Program Support Functions

## 13.3 Compound Type Fixed Cycle for Turning Machining



### Finishing margin

When the finishing allowance is designated, the Uu/Ww section will be left uncut from the finished shape.

- X axis finishing allowance . . . The finishing allowance is left uncut in the rough cutting start point direction.
- Z axis finishing allowance . . . The finishing allowance is left uncut in the rough cutting start point direction.

Note that if down cutting (block commanded in direction in which X axis faces hole base) is executed when pocket machining is ON and then cutting is started from the middle of the longitudinal direction, the Z axis finishing allowance direction will be reversed.

<p>&lt;Finishing allowance when pocket machining is OFF or finishing allowance of open section when pocket machining is ON (finished shape start block X axis is at base)&gt;</p>	
<p>G71 Ud Re H0 (or H1) ; G71 Pp Qq ... ; Finished shape end block</p>	<p>Cycle command point Rough cutting start point Finished shape start block</p>
<p>&lt;Finishing allowance of mid-section when pocket machining is ON (Finished shape start block X axis = finished shape end block X axis)&gt;</p>	
<p>G71 Ud Re H1; (Valid when pocket machining is ON) G71 Pp Qq ... ; Finished shape end block</p>	<p>The block that starts cutting from the middle of the longitudinal direction will have a reversed Z axis finishing allowance direction. <b>Note)</b> During nose R compensation, program so that the p and q X axis positions after nose R compensation have the same value. (Refer to "Cutting shape" (5).)</p>
<p>Rough cutting start point Finished shape start block</p>	<p>The block that executes down cut has a reversed Z axis finishing allowance direction.</p>

# 13. Program Support Functions

## 13.3 Compound Type Fixed Cycle for Turning Machining



### Example of program

Machining of open section (example with pocket machining OFF)	
<p>G71 U<math>\underline{d}</math> R<math>\underline{e}</math> H<math>\underline{0}</math> ;</p>	<pre> G0 X80.0 Z75.0 T0101 ; ← (1) G71 U10. R3. H0 ; ← (2) G71 P10 Q20 U3.W1.5 F500 S1500 ; ← (3) N10 G0 X15.0 Z65.0 ; G1 Z55. F450 ; G1 X30.0 ; G3 X40.0 Z50.0 R5.0 ; G1 Z42.0 ; G1 X50.0 ; G1 X55.0 Z35.0 ; N20 G1 X60.0 ; G70 P10 Q20 ; ← (4)                     </pre> <p>Finished program</p> <p>Program execution order (1) (2) (3) (4)</p>
Machining of open section (example with pocket machining ON)	
<p>G71 U<math>\underline{d}</math> R<math>\underline{3}</math> H<math>\underline{1}</math> ;</p>	<pre> G0 X80.0 Z75.0 T0101 ; ← (1) G71 U10. R3. H1 ; ← (2) G71 P10 Q20 U3.W1.5 F500 S1500 ; ← (3) N10 G0 X15.0 Z65.0 ; G1 Z55. F450 ; G1 X30.0 ; G3 X40.0 Z50.0 R5.0 ; G1 Z42.0 ; G1 X50.0 ; G1 X55.0 Z35.0 ; N20 G1 X60.0 ; G70 P10 Q20 ; ← (4)                     </pre> <p>Finished program</p> <p>Program execution order (1) (2) (3) (4)</p>
Machining of mid-section (example with pocket machining ON)	
<p>G71 U<math>\underline{d}</math> R<math>\underline{e}</math> H<math>\underline{1}</math> ;</p>	<pre> G0 X80.0 Z75.0 T0101 ; ← (1) G71 U10. R3. H1 ; ← (2) G71 P10 Q20 U3.W1.5 F500 S1500 ; ← (3) N10 G0 X60.0 Z73.0 ; G1 X15.0 Z65.0 ; G1 Z55. F450 ; G1 X30.0 ; G3 X40.0 Z50.0 R5.0 ; G1 Z42.0 ; G1 X50.0 ; G1 X55.0 Z35.0 ; N20 G1 X60.0 ; G70 P10 Q20 ; ← (4)                     </pre> <p>Finished program</p> <p>Program execution order (1) (2) (3) (4)</p>

# 13. Program Support Functions

## 13.3 Compound Type Fixed Cycle for Turning Machining

### 13.3.2 Face rough cutting cycle; G72



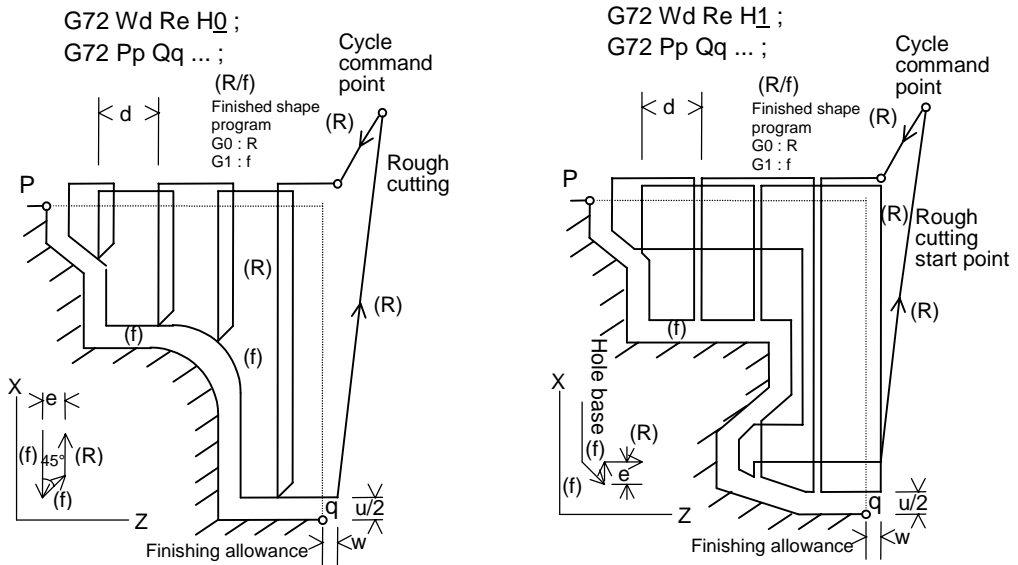
#### Function and purpose

This function calls the finished shape program and, while automatically calculating the path, performs rough cutting in the face direction.



#### Command format

**G72 Wd Re Hh ; (May be omitted when using value set in parameters)**  
**G72 Ae Pp Qq Uu Ww Ff Ss Tt ;**



Wd	Cut amount (modal) Unit ···· μm/ 1/10000 inch radius value command
Re	Retract amount (modal) ····· Reversible parameter ("G71 PULL UP") Unit ···· μm/ 1/10000 inch radius value command
Hh	Pocket machining (modal) ··· Reversible parameter ("G71 POCKET") 0: Select this only for finished shapes that have no indent section (pocket). * The cutting section will retract 45° per cycle, and finally will trace the finished shape. 1: This can be selected whether or not there is an indent (pocket) in the finished shape. * The cutting section will trace the finished shape each cycle.
Aa	Finished shape program No. (Program being executed when omitted) If the A command is omitted, the P and Q commands in the program being executed will be used. If A is omitted, the program will be executed in the order of G72 and then the program following Qq (finished shape end sequence No.)
Pp	Finished shape start sequence No. (Head of program when omitted)
Qq	Finished shape end sequence No. (To end of program when omitted) Even when Q is commanded, if M99 is commanded first the program will continue to M99.

## 13. Program Support Functions

### 13.3 Compound Type Fixed Cycle for Turning Machining

Uu	X axis direction finishing allowance (When omitted, X axis direction finishing allowance is handled as 0.) Unit ··· μm/ Follows 1/10000ich diameter/radius value command (#1019 dia).
Ww	Z axis direction finishing allowance (When omitted, Z axis direction finishing allowance is handled as 0.) Unit ··· μm/ Follows 1/10000ich radius value command
Ff	Cutting speed (Follows cutting speed (modal) before G72 when omitted)
Ss,Tt	Spindle command, tool command

\* The other detailed functions are the same as "13.3.1 Longitudinal rough cutting cycle".

## 13. Program Support Functions

### 13.3 Compound Type Fixed Cycle for Turning Machining

#### 13.3.3 Formed material rough cutting cycle; G73



#### Function and purpose

This calls the finished shape program, automatically calculates the path and performs rough cutting while cutting the workpiece into the finished shape.



#### Command format

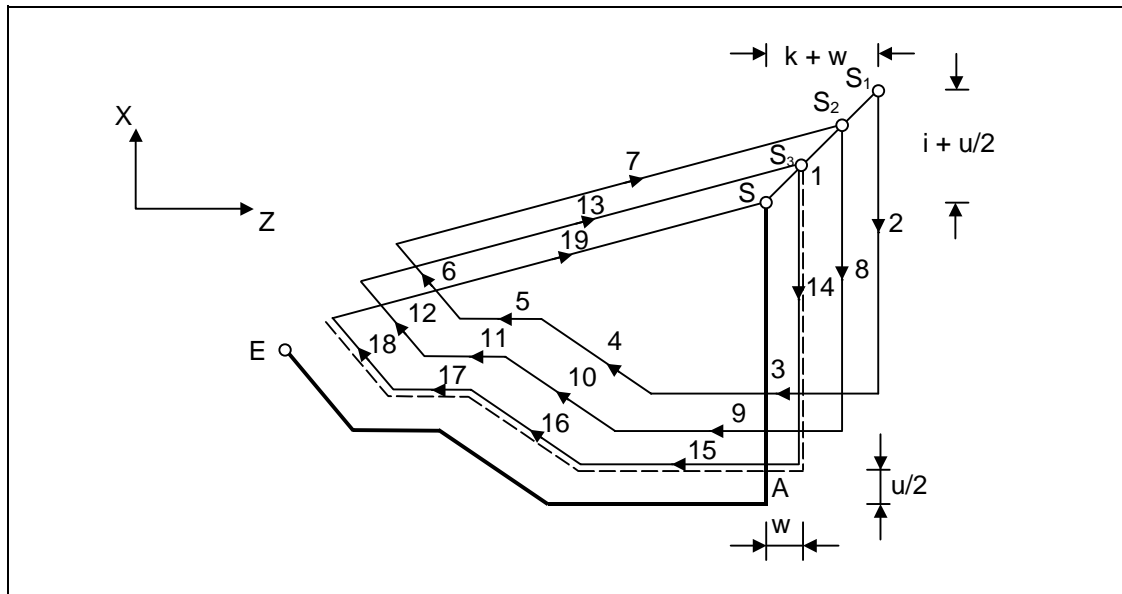
**G73 Ui Wk Rd ;**

**G73 Aa Pp Qq Uu Ww Ff Ss Tt ;**

Ui	X-axis direction cutting allowance	i	}	• Cutting allowances when P, Q commands are not present
Wk	Z-axis direction cutting allowance	k		
Rd	Number of divisions	d		
				• Modal data
				• Sign is ignored
				• Radius designation applies to the cutting allowance.
Aa	Finished shape program number			(program being executed when omitted)
Pp	Finished shape start sequence number			(program head when omitted)
Qq	Finished shape end sequence number			(up to end of program or M99 when omitted)
				Note that this is up to M99 when M99 comes first, even when there is a Qq command.
Uu	X-axis direction finishing allowance	u	}	• Cutting allowance when P, Q commands are present
Ww	Z-axis direction finishing allowance	w		
				• Sign is ignored.
				• Diameter/radius designation changes in accordance with the parameters (#1019 dia).
				• The shift direction is determined by the shape. For details, reference should be made to the "finishing allowance direction" for G71.
Ff	Cutting feedrate (F function)		}	The F, S and T commands in the finished shape program are ignored and the value in the rough cutting cycle command or previous value is valid.
Ss	Spindle speed (S function)			
Tt	Tool command (T function)			

# 13. Program Support Functions

## 13.3 Compound Type Fixed Cycle for Turning Machining



**(Note)** With a single block, operation stops at the end point of each block.



## 13. Program Support Functions

### 13.3 Compound Type Fixed Cycle for Turning Machining



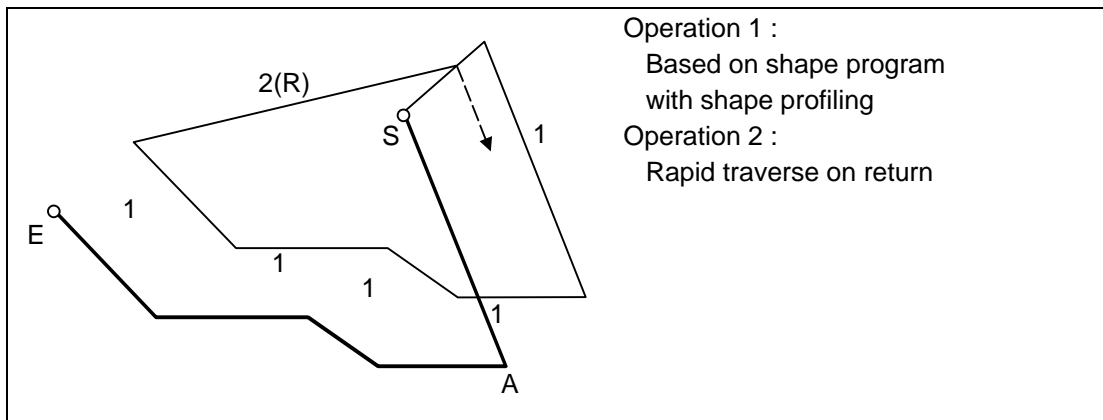
#### Finished shape

In the program,  $S \rightarrow A \rightarrow E$  in the previous program are commanded.  
The section between A and E must be a shape with monotonous changes in both the X axis and Z axis directions.



#### 1 cycle configuration

A cycle is configured as shown below.



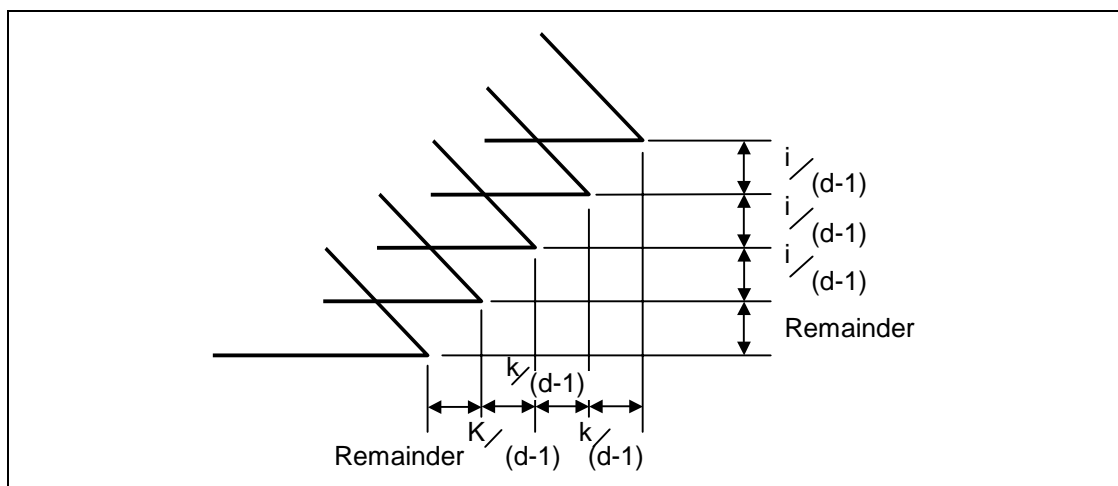
#### Cut amount

The cut amount is the value produced by dividing the cutting allowances (i, k) by the number of divisions (d-1).

X axis direction  $i/(d-1)$

Z axis direction  $k/(d-1)$

When divisions cannot be made, chamfering is performed and adjustment is made at the final pass.



## 13. Program Support Functions

### 13.3 Compound Type Fixed Cycle for Turning Machining



#### Tool nose R compensation

When this cycle is commanded with the tool nose R compensation mode still in force, tool nose R compensation is applied to the finished shape program covered by this cycle and the cycle is executed for this shape.

However, when this cycle is commanded with the tool nose R compensation mode still in force, the compensation is temporarily canceled immediately before this cycle and started at the head block of the finished shape program.



#### Others

##### (1) Cutting direction

- Designate with program (#1273 ext09/bit2=0)

The shift direction for the cutting is determined by the shape in the finishing program, as shown in the table below.

	1	2	3	4
Drawing				
Initial X axis	"-" direction	-	+	+
Complete Z axis	"-" direction	+	+	-
X axis cutting	"+" direction	+	-	-
Z axis cutting	"+" direction	-	-	+

- Determined according to commanded finish margin/cutting margin (#1273 ext09/bit2=1)

The shift direction for cutting follows the sign of the cutting allowance, etc., designated in the program.

## 13. Program Support Functions

### 13.3 Compound Type Fixed Cycle for Turning Machining

#### 13.3.4 Finishing cycle; G70



##### Function and purpose

After rough cutting have been carried out by the G71 ~ G73 commands, finishing cutting can be performed by the following command.



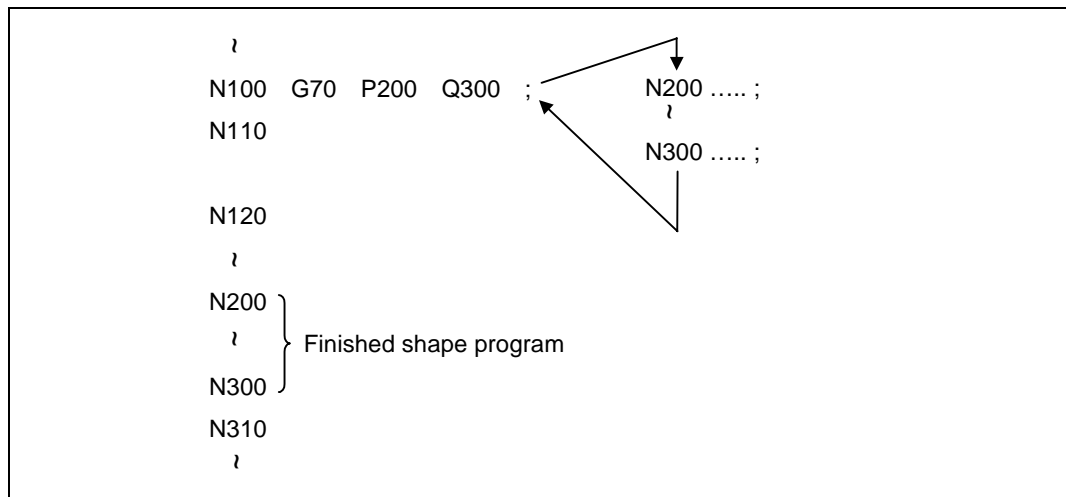
##### Command format

**G70 A\_ P\_ Q\_ ;**

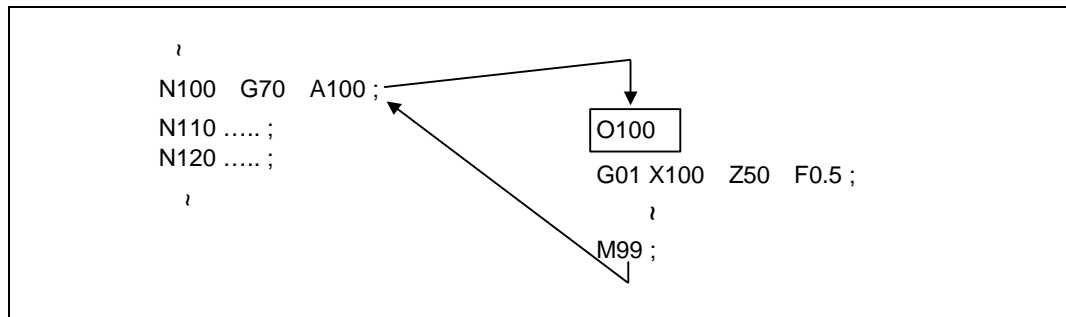
- A Finished shape program number (program being executed when omitted)
- P Finished shape start sequence number (program head when omitted)
- Q Finished shape end sequence number (up to end of program when omitted)  
Note that this is up to M99 when M99 comes first, even when there is a Q command.

- (1) The F, S and T commands in the finished shape program are valid during the finishing cycle.
- (2) When the G70 cycle is completed, the tool returns to the start point by rapid traverse and the next block is read.

##### (Example 1) When a sequence number is designated



##### (Example 2) When a program number is designated



If the N100 cycle is executed in either Example 1 or Example 2, the N110 block is executed next.

## 13. Program Support Functions

### 13.3 Compound Type Fixed Cycle for Turning Machining

#### 13.3.5 Face cut-off cycle; G74



#### Function and purpose

The G74 fixed cycle automatically enters a groove in the face direction of the workpiece by commanding the coordinates of the groove end point, cut amount, cutter shift amount and cutter escape at the bottom of the cut.

The machining program commands are as follows.



#### Command format

**G74 Re ;**

**C74 X(U)x Z(W)z Pi Qk Rd Ff ;**

Re Return amount (no X/U, P commands) (modal)

X/Ux B point X coordinate (absolute/incremental value)

Z/Wz B point Z coordinate (absolute/incremental value)

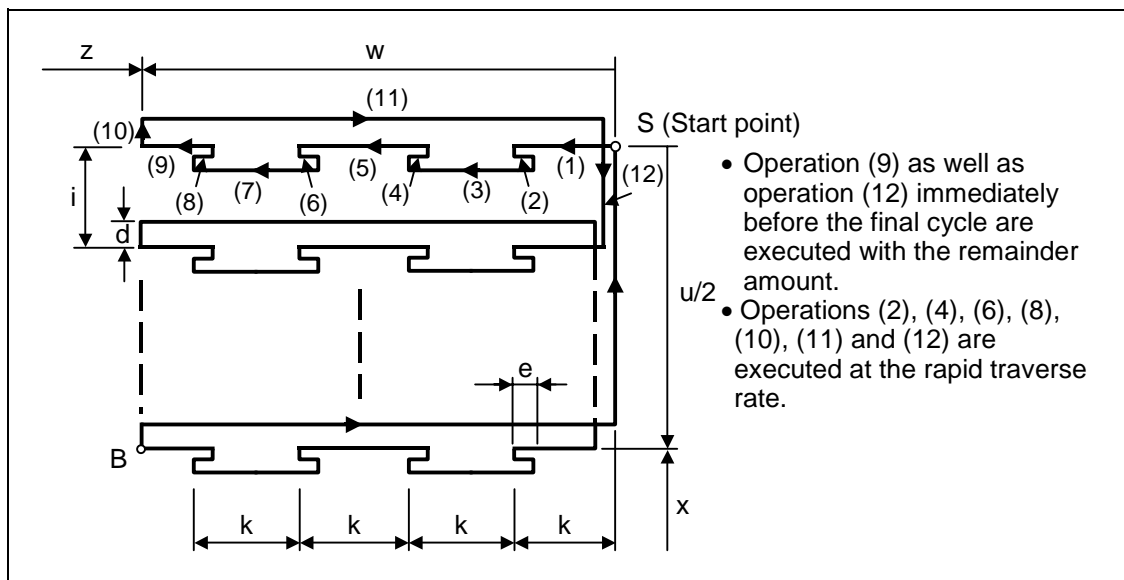
Pi Tool shift amount (radius designation, incremental value, sign not required)

Qk Cut amount (radius designation, incremental value, sign not required)

Rd Escape at the bottom of the cut

When there is no sign, escapes even at the bottom of the first cut. When a - sign is attached, the tool escapes from the second cut without escaping the first cut.

Ff Feedrate



## 13. Program Support Functions

### 13.3 Compound Type Fixed Cycle for Turning Machining



#### Single block stop

Operation stops at each block from (1) to (12).



#### Others

- (1) When X/U and P are omitted or when the values of "x" and "i" are zero, operation will apply to the Z axis only. Note that when there is an Rd command and no sign, the tool will escape at the bottom of the cut.
- (2) A case where the X/U or Z/W command is not present is treated as the assignment of a parameter setting command (G74 Re). Even when G74 Pi Qk Rd ; is commanded, Rd is regarded as Re, and the return amount is set.
- (3) The escape direction does not change when a - sign is attached to the Rd command, or when there is no sign.
- (4) A program error (P204) results in the following cases.
  - (a) When " i " is zero or P has not been commanded even though X/U has been commanded
  - (b) When tool shift amount " i " is greater than the "x" movement amount
  - (c) When the escape amount "d" is greater than the shift amount " i "
  - (d) When the return amount "e" is greater than the cut amount "k"
  - (e) When the cut in amount "k" is larger than the hole depth "w".

## 13. Program Support Functions

### 13.3 Compound Type Fixed Cycle for Turning Machining

#### 13.3.6 Longitudinal cut-off cycle; G75



#### Function and purpose

The G75 fixed cycle automatically enters a groove in the longitudinal direction of the workpiece by commanding the coordinates of the groove end point, cut amount, cutter shift amount and cutter escape at the bottom of the cut.



#### Command format

**G75 Re ;**

**C75 X(U)x Z(w)z Pi Qk Rd Ff ;**

Re Return amount (no X/U, P commands) (modal)

X/Ux B point X coordinate (absolute/incremental value)

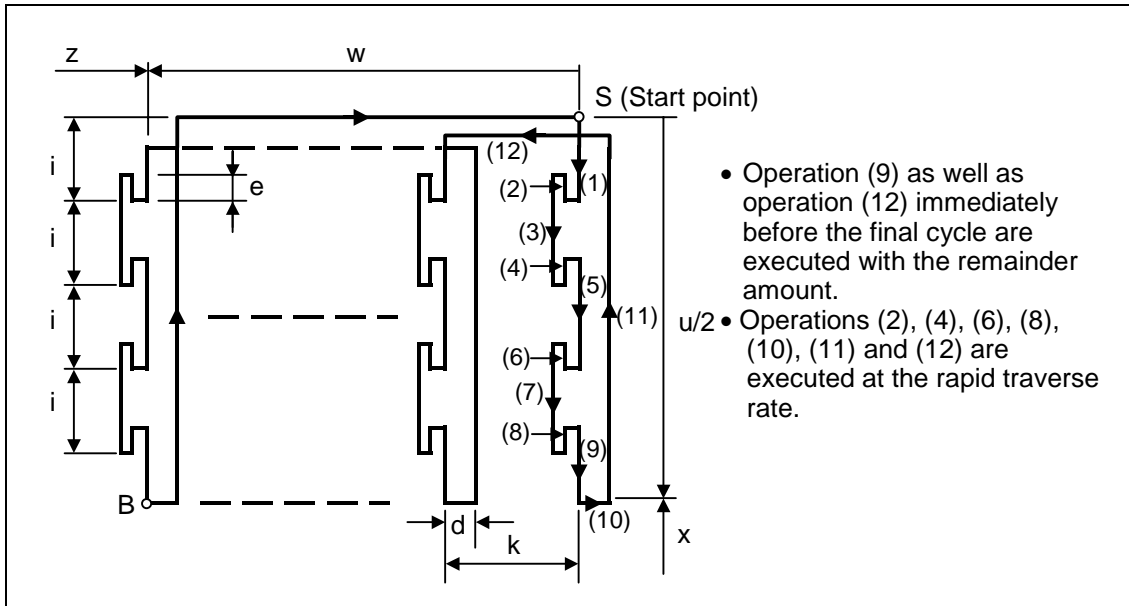
Z/Wz B point Z coordinate (absolute/incremental value)

Pi Cut amount (radius designation, incremental value, sign not required)

Qk Tool shift amount (radius designation, incremental value, sign not required)

Rd Escape at the bottom of the cut } When there is no sign, escapes even at the bottom of the first cut. When a - sign is attached, the tool escapes from the second cut without escaping the first cut.

Ff Feedrate



## 13. Program Support Functions

### 13.3 Compound Type Fixed Cycle for Turning Machining



#### Single block stop

Operation stops at each block from (1) to (12).



#### Others

- (1) When Z/W and Q are omitted or when the values of "z" and "k" are zero, operation will apply to the X axis only (slitting). Note that when there is an Rd command and no sign, the tool will escape at the bottom of the cut.
- (2) A case where both the X/U and Z/W commands are not present is treated as the assignment of a parameter setting command (G75 Re). Even when G75 Pi Qk Rd ; is commanded, Rd is regarded as Re, and the return amount is set.
- (3) The escape direction does not change when a - sign is attached to the Rd command, or when there is no sign.
- (4) A program error (P204) results in the following cases.
  - (a) When "k" is zero or Q has not been commanded even though Z/W has been commanded
  - (b) When tool shift amount "k" is greater than the "z" movement amount
  - (c) When the escape amount "d" is greater than the shift amount "k"
  - (d) When the return amount "e" is greater than the cut amount " i "
  - (e) When the cut in amount "i" is larger than the hole depth "u/2".

## 13. Program Support Functions

### 13.3 Compound Type Fixed Cycle for Turning Machining

#### 13.3.7 Compound thread cutting cycle; G76



##### Function and purpose

The G76 fixed cycle enables the workpiece to be cut at the desired angle by designating the thread cutting start point and end point, and it automatically cuts so that the cutting cross section (cutting torque) per cutting pass is constant.

Various longitudinal direction threads can be cut by bearing in mind the command value for the thread end point coordinate and taper height component.



##### Command format

**G76 Pmra Q $\Delta$ admin Rd ;**

**G75 X/U Z/W Ri Pk Q $\Delta$ d FI ;**

m	Number of cutting passes for finishing: 00 ~ 99 (modal)
r	Chamfering amount: 00 ~ 99 (modal) The chamfering width based on thread lead "l" is designated by a 2-digit integer without decimal point from 0.0 l to 9.9 l.
a	Tool nose angle (thread angle): 00 ~ 99 (modal) The angle from 0° to 99° is assigned in 1° units. "m", "r" and "a" are commanded in succession in address P. <b>(Example)</b> When m=5, r=1.5 and a=0° P is 051500 and the leading and trailing zeroes cannot be omitted.
$\Delta$ admin	Minimum cut amount If the calculated cut amount is smaller than $\Delta$ admin, it is clamped by $\Delta$ admin. (When omitting, the operation depends on #1222/bit4 setting.)
d	Finishing allowance (modal)
X/U	X-axis end point coordinate of thread The X coordinate of the end point for the thread is commanded by an absolute or incremental value.
Z/W	Z-axis end point coordinate of thread The Z coordinate of the end point for the thread is commanded by an absolute or incremental value.
i	Taper height component (radius value) for thread straight thread when "i" is zero
k	Thread height This is commanded by a positive radial value.
$\Delta$ d	Cut amount The cut amount of the first cutting pass is commanded by a positive radius value.
l	Thread lead

**(Note 1)** The two above G76 commands cannot be placed in the same block.  
The data commanded by P, Q and R are automatically identified according to the presence or absence of the X/U and Z/W axis addresses.

**(Note 2)** Parameter settings can be used for the above "r" modal data but these parameter (#8014 chamfer amount) settings are rewritten by the program commands.

**(Note 3)** The chamfering amount designation is valid even for thread cutting fixed cycles.



## 13. Program Support Functions

### 13.3 Compound Type Fixed Cycle for Turning Machining

(Note 4) Program error (P204) results in the following cases.

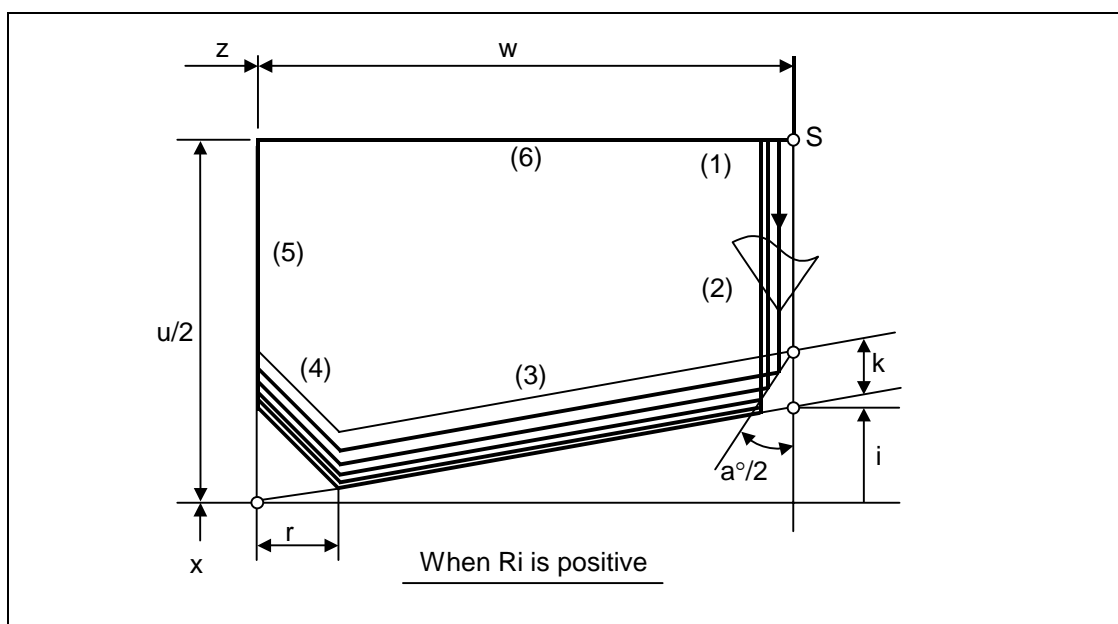
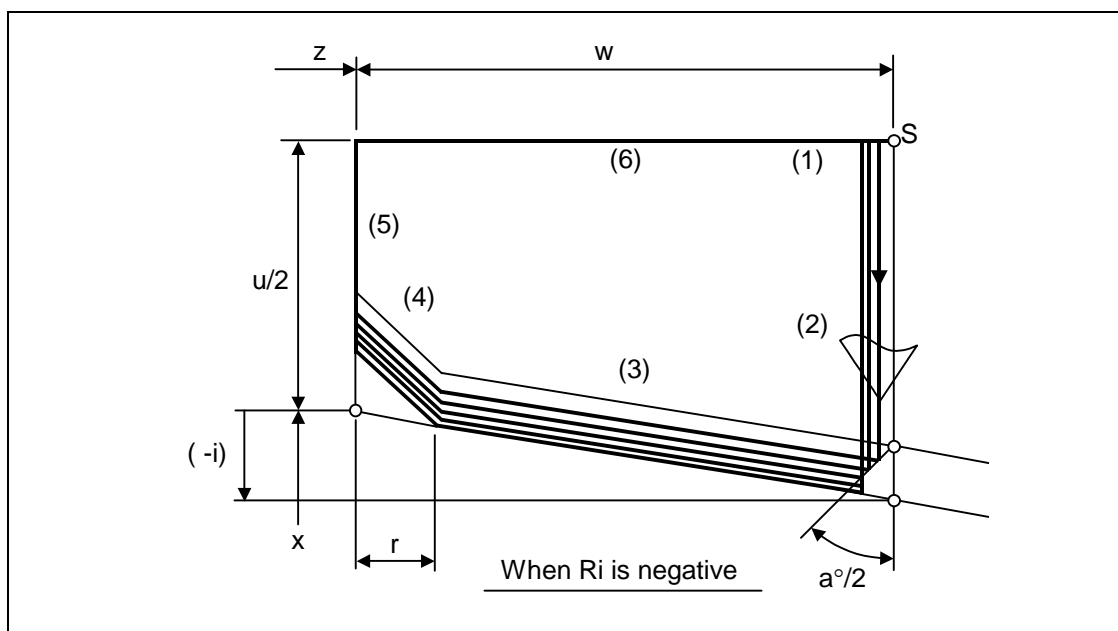
- When "a" is outside the rating
- When both the X and Z commands have not been issued or when the start and end point coordinates are the same for either the X or Z command.
- When the thread is greater than the movement of the X axis at the thread bottom

(Note 5) The precautions for the thread cutting command (G33) and thread cutting cycle (G78) should be observed.



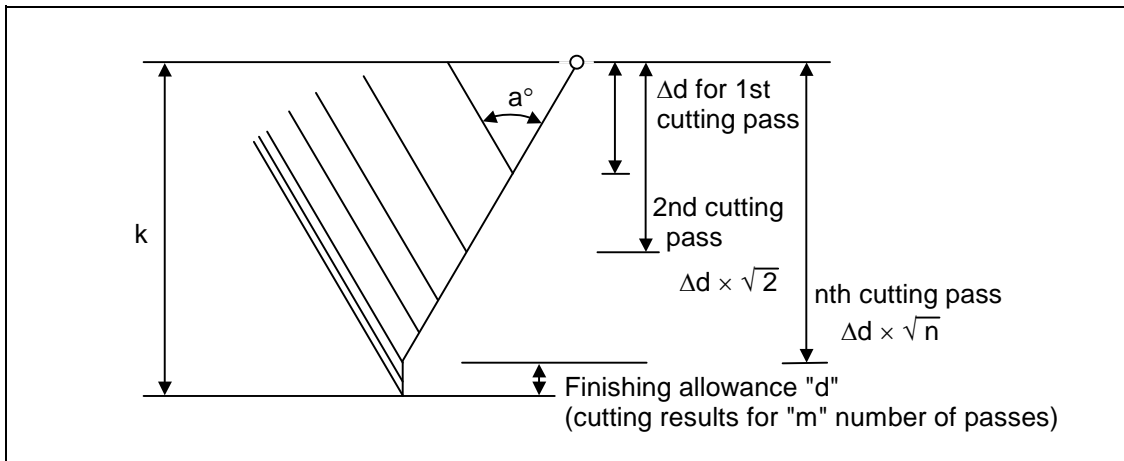
#### 1 cycle configuration

The tool moves at rapid traverse for operations (1), (2), (5) and (6) in the cycle and at the cutting feed based on the F designation for operations (3) and (4).

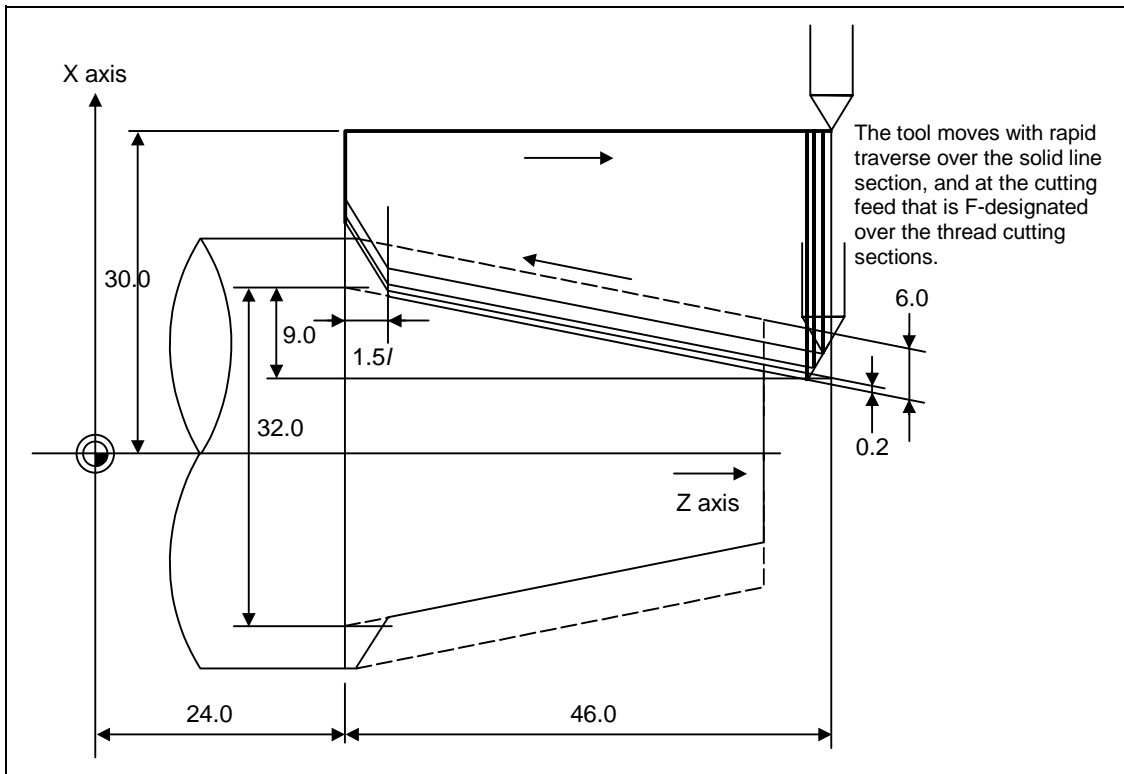


# 13. Program Support Functions

## 13.3 Compound Type Fixed Cycle for Turning Machining



### Example of program



G76 P011560 R0.2 ;

G76 U-28.0 W-46.0 R9.0 P6.0 Q3.5 F4.0 ;

## 13. Program Support Functions

### 13.3 Compound Type Fixed Cycle for Turning Machining



#### Interrupt operation

- (1) When the feed hold button is pressed while G76 is being executed, automatic operation will stop upon completion of a block without thread cutting if thread cutting is ongoing. (The automatic operation pause lamp lights immediately in the feed hold mode and it goes off when automatic operation stops.)  
If feed hold is applied when no thread is then being cut, or if applied in the interval from the execution of the thread cutting command to the axis movement, the automatic operation pause lamp will turn ON, and the automatic operation will pause.
- (2) The tool stops upon completion of operations (1), (4) and (5) when the mode is switched to another automatic operation mode during the G76 command execution, when automatic operation is changed to manual operations or when single block operation is conducted.
- (3) The dry run valid/invalid status during G76 execution does not change during thread cutting.

## 13. Program Support Functions

### 13.3 Compound Type Fixed Cycle for Turning Machining

#### 13.3.8 Precautions for Compound Type Fixed Cycle for Turning Machining; G70 ~ G76



#### Precautions

- (1) Command all required parameters in a compound type fixed cycle for turning machining command block.
- (2) Provided that the finished shape program has been entered in the memory, compound type fixed cycle for turning machining I commands can be executed in the memory, MDI or tape operation mode.
- (3) When executing a G70 ~ G73 command, ensure that the sequence number of the finished shape program which is stop with P and Q is not duplicated in that program.
- (4) The finished shape program specified by P and Q in the G71 ~ G73 blocks should be prepared so that the maximum number of blocks is 200 for all the commands for corner chamfering, corner rounding and other commands including the automatic insertion blocks based on tool nose R compensation. If this number is exceeded, program error (P202) results.  
If the cutting start position is the cycle start position (#1271 ext07/bit5 =1), the finished shape program can have up to 199 blocks. (When the cycle start point is on the other side of the finished shape program's final position.)
- (5) The finished shape program which is designated by the G71 ~ G73 blocks should be a program with monotonous changes (increases or reductions only) for both the X and Z axes.
- (6) Blocks without movement in the finished shape program are ignored.
- (7) N, F, S, M, and T commands in the finished shape program are ignored.
- (8) When any of the following commands are present in a finished shape program, program error (P201) results.
  - (a) Commands related to reference point return (G27, G28, G29, G30)
  - (b) Thread cutting (G33)
  - (c) Fixed cycles
  - (d) Skip functions (G31, G37)
- (9) If subprogram call or macro call commands are present in the finished shape program, these commands will also be executed.
- (10) Except for thread cutting cycles, operation stops at the end (start) point of each block in the single block mode.

## 13. Program Support Functions

### 13.3 Compound Type Fixed Cycle for Turning Machining

- (11) Remember that, depending on whether the sequence or program number is designated, the next block upon completion of the G71, G72 or G73 command will differ.

(a) When the sequence number is designated	(b) When the program number is designated
<p>The next block is the next block designated by Q.</p> <pre> } N100 G71P200 Q500 U_W_ ..... ; N200 N300 N400 } Finished shape N500 } program N600  Operation moves to the N600 block upon completion of the cycle.                     </pre>	<p>The next block is the next block of the cycle command.</p> <pre> } N100 G71 A100 U_W_..... ; N200 N300 N400 O100 N10 X100. Z50. ; N20  Operation moves to the N200 block upon completion of the cycle.                     </pre>

- (12) The next block applying upon completion of the G70 command is the next block of the command block.

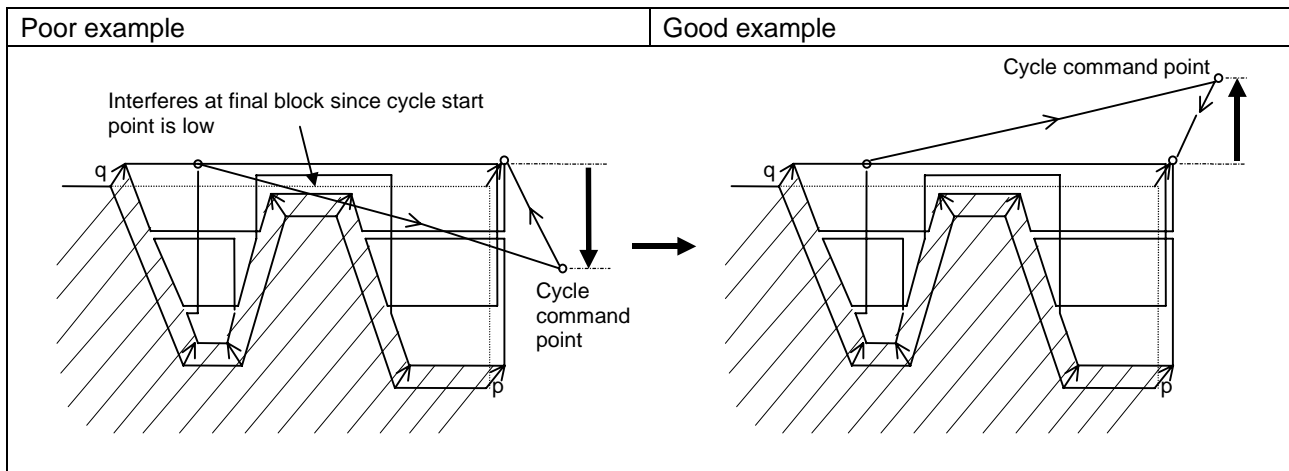
<pre> } N100 ..... ; N200 ; N300 ; N400 ; N500 ; } N1000 G70 P200 Q500 ; (or G70 A100 ;) N1100 ..... ; }  Operation moves to the N1100 block upon completion of the G70 command.                     </pre>
---

- (13) It is possible to apply a manual interrupt while a compound type fixed cycle for turning machining command (G70 ~ G76) is being executed. However, upon completion of the interrupt, the tool must first be returned to the position where the interrupt was applied and then the compound type fixed cycle for turning machining must be restarted. If it is restarted without the tool having been returned, all subsequent movements will deviate by an amount equivalent to the manual interrupt amount.
- (14) Compound type fixed cycle for turning machining commands are unmodal commands and so they must be issued every time they are required.
- (15) Program error (P203) results with the G71 and G72 commands even when, because of tool nose R compensation, there is no further movement of the Z axis in the second block or the Z axis has moved in the opposite direction.
- (16) A program error (P204) will occur if the finished shape program is issued in the same program with G70 to G73 and both P and Q are not designated. Note that if A, P and Q are all not designated in G71 to G73, this will be interpreted as the first block for the compound fixed cycle I, and an error will not occur.

## 13. Program Support Functions

### 13.3 Compound Type Fixed Cycle for Turning Machining

- (17) Set the cycle start point in the final block of the compound turning fixed cycle I so that the tool retracts from the workpiece cutting face. This is necessary to prevent the cutter from interfering with the convex sections of the workpiece.



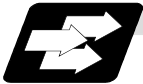
- (18) The compound type fixed cycle for turning machining type II is E68 system specification. H1 will operate as H0 even if designated in E60.

## 13. Program Support Functions

### 13.4 Compound Fixed Cycle (MELDAS Special Format)

#### 13.4 Compound Fixed Cycle (MELDAS Special Format)

[E68]



#### Function and purpose

This function allows a prepared fixed cycle to be executed by commanding the program in one block.

When using the MELDAS special format (#1265 ext01/bit0 ON), some of the addresses differ from the normal format.

With the normal format, G71 to G76 are commanded in two blocks. However with the MELDAS special format, these are commanded with one block. The sections that differ from the normal format are explained in this section.

Refer to section "13.3 Compound fixed cycle" for details on the compound fixed cycle.



#### Command format

**G71 P\_ Q\_ U\_ W\_ D\_ F\_ S\_ T\_ ;**  
(Same for G72)

P	Finished shape start sequence No.
Q	Finished shape end sequence No.
U	X axis direction finishing allowance
W	Z axis direction finishing allowance
D	Cut amount
F	Cutting speed
S	Spindle speed
T	Tool command

**G73 P\_ Q\_ U\_ W\_ I\_ K\_ D\_ F\_ S\_ T\_ ;**

P	Finished shape start sequence No.
Q	Finished shape end sequence No.
U	X axis direction finishing allowance
W	Z axis direction finishing allowance
I	X axis direction cutting allowance
K	Z axis direction cutting allowance
D	Number of divisions
F	Cutting speed
S	Spindle speed
T	Tool command

**G74 X(U)\_ Z(W)\_ I\_ K\_ F\_ D\_ ;**  
(Same for G75)

X(U)	X axis slot insertion end point coordinate
Z(W)	Z axis slot insertion end point coordinate
I	Tool shift amount (X/U axis direction movement amount)
K	Cut amount (Z/W axis direction cut amount)
F	Cutting speed
D	Tool escape amount at cutting base

## 13. Program Support Functions

### 13.4 Compound Fixed Cycle (MELDAS Special Format)

G76 X(U)_ Z(W)_ I_ K_ D_ F_ A_ Q_ P_;	
X(U)	Thread section X axis end point coordinate
Z(W)	Thread section Z axis end point coordinate
I	Radius value at thread section
K	Thread height
D	1st cut amount
F	Thread lead
A	Thread angle
Q	Thread cutting start shift amount (0.001 to 360.000 [°])
P	Zigzag thread cutting at P2 command



#### Detailed description

##### (1) Check of command format

A check is performed to check whether the normal command format is being used with the MELDAS special format.

[When normal format is selected (#1265 ext01/bit0 OFF)]

###### (a) Check of command format

If the format check selection parameter (#1222 aux06/bit5) is set to 0, a program error (P33) will occur in the following cases.

- There is no 1st block command.
- A MELDAS special format is commanded.

###### (b) Check of command address

A program error (P32) will occur in the following cases.

- When address I, K or D is commanded in the G71, G72, G73, G74 or G75 block.
- When address I, K, D or A is commanded in the G76 block.

[When MELDAS special format is selected (#1265 ext01/bit0 ON)]

###### (a) Check of command format

A program error (P33) will occur if a normal format is commanded.

###### (b) Check of command address

A program error (P32) will occur in the following cases.

- When address R or A is commanded in the G71, G72 or G73 block.
- When address P, Q or R is commanded in the G74 or G75 block.
- When address R is commanded in G76 the block.



## 13. Program Support Functions

### 13.4 Compound Fixed Cycle (MELDAS Special Format)

#### (2) Comparison of MELDAS special format and normal format

While the normal format is commanded with two blocks, the MELDAS special format is commanded with one block. Some addresses differ, and some are designated with the parameters.

Function	MELDAS special format	Normal format	Difference from normal format
Longitudinal rough cutting cycle I	G71 P Q U W D F S T ;	(1) G71 U R ; (2) G71 A P Q U W F S T ;	Cut amount Designate with U (1) → D Retract amount R (1) → Parameter (#8052 G71 PULL UP) No A command (finished shape program No.)
Face rough cutting cycle I	G72 P Q U W D F S T ;	(1) G72 W R ; (2) G72 A P Q U W F S T ;	Cut amount Designate with W (1) → D Retract amount R (1) → Parameter (#8052 G71 PULL UP) No A command (finished shape program No.)
Formed material rough cutting cycle	G73 P Q U W I K D F S T ;	(1) G73 U W R ; (2) G73 A P Q U W F S T ;	X axis cutting allowance U (1) → I Z axis cutting allowance W (1) → K Number of divisions R (1) → D No A command (finished shape program No.)
Face cut-off cycle	G74 X Z I K F D ; or G74 U W I K F D ;	(1) G74 R ; (2) G74 X Z P Q R F ; or (1) G74 R ; (2) G74 U W P Q R F ;	X axis direction movement amount P (2) → I Z axis direction cut amount Q (2) → K Escape amount at cutting base R (2) → D Return amount R (1) → parameter (#8056 G74 RETRACT)
Outer diameter cut-off cycle	G75 X Z I K F D ; or G75 U W I K F D ;	(1) G75 R ; (2) G75 X Z P Q R F ; or (1) G75 R ; (2) G75 U W P Q R F ;	X axis direction cut amount P (2) → I Z axis direction movement amount Q (2) → K Escape amount at cutting base R (2) → D Return amount R (1) → parameter (#8056 G74 RETRACT)
Compound thread cutting cycle	G76 X_ Z_ I_ K_ D_ F_ A_ Q_ ;	(1) G76 Pmra R_ ; (2) G76 X_ Z_ R_ P_ Q_ F_ ;	Radius at thread section R (2) → I Thread height P (2) → K 1st cut amount Q (2) → D (Q in the MELDAS special format is the thread cutting start shift angle) Thread angle P (1) a → A Thread cutting P (1) r → parameter (#8014 CDZ-VALE) Number of finish cuts P (1) m → parameter (#8058 TIMES) Finishing allowance R (1) → parameter (#8057 G76 LAST-D)

## 13. Program Support Functions

### 13.4 Compound Fixed Cycle (MELDAS Special Format)



#### Precautions and restrictions

- (1) Always designate P and Q in G71 to G73. A program error (P204) will occur if either is not designated.
- (2) The thread cutting start shift angle is to a modal. If there is no Q command at G76, this will be handled as "Q0".
- (3) If a value exceeding 360.000 is commanded for Q in G76, it will be handled as "Q360.000".
- (4) G76 cuts one row with one cycle. To cut two rows, change the Q value, and issue the same command.

## 13. Program Support Functions

### 13.5 Hole drilling fixed cycles

#### 13.5 Hole drilling fixed cycles; G80 ~ G89



#### Function and purpose

These fixed cycles are used for predetermined sequences of machining operations such as normal positioning, hole drilling, boring and tapping which are specified in a block. The various sequences available using these functions are listed in the table below.

The type II is E68 system specification. Thus, this cannot be used in E60.

G code	Hole drilling axis	Hole drilling start	Operation at hole bottom	Return operation	Application	Type II	
						Longitudinal hole drilling axis selection signal ON/OFF	Hole drilling axis
G80	-	-	-	-	Cancel	-	-
G83	Z	Cutting feed, intermittent feed	In-position check Dwell	Rapid traverse	Deep hole drilling cycle 1	ON	Z
						OFF	
G84 (G84.1)	Z	Cutting feed	In-position check Dwell, spindle reverse rotation	Cutting feed	Tapping cycle (Reverse tap cycle)	ON	Z
						OFF	
G85	Z	Cutting feed	In-position check Dwell	Cutting feed	Boring cycle	ON	Z
						OFF	
G87	X	Cutting feed, intermittent feed	In-position check Dwell	Rapid traverse	Deep hole drilling cycle 1	ON	Y
						OFF	X
G88 (G88.1)	X	Cutting feed	In-position check Dwell, spindle reverse rotation	Cutting feed	Tapping cycle (Reverse tap cycle)	ON	Y
						OFF	X
G89	X	Cutting feed	In-position check Dwell	Cutting feed	Boring cycle	ON	Y
						OFF	X
G83.2	Z/X	Cutting feed, intermittent feed	In-position check Dwell	Rapid traverse	Deep hole drilling cycle 2	ON	Z/X
						OFF	

A fixed cycle mode is canceled when the G80 or any G command in the 01 group is issued. The various data will also be cleared simultaneously to zero.



## Command format

## (1) Face hole drilling

<b>G8Δ X/U_ C/H_ Z/W_ R_ Q_ P_ F_ K_ D_ S_ ,S_ M_ ;</b>	
G8Δ	Hole machining mode (G83, G84, G85)
X/U_ C/H_	Hole positioning data
Z/W_ R_ Q_ P_ F_	Hole machining data
K_	Number of repetitions
D_	Tool spindle No.
S_	Spindle rotation speed
,S_	Spindle rotation speed at return
M_	Miscellaneous command

## (2) Longitudinal hole drilling

<b>G8* Z/W_ C/H_ X/U_ R_ Q_ P_ F_ K_ D_ S_ ,S_ M_ ;</b>	
G8*	Hole machining mode (G87, G88, G89)
Z/W_ C/H_	Hole positioning data
X/U_ R_ Q_ P_ F_	Hole machining data
K_	Number of repetitions
D_	Tool spindle No.
S_	Spindle rotation speed
,S_	Spindle rotation speed at return
M_	Miscellaneous command

## (3) Cancel

<b>G80 ;</b>
--------------

### (4) Data outline and corresponding addresses

- (a) Hole machining modes : These are the fixed cycle modes for drilling (G83, G87), tapping (G84, G88) and boring (G85, G89).  
These are modal commands and once they have been issued, they will remain valid until another hole machining mode command, the cancel command for the hole drilling fixed cycle or a G command in the 01 group is issued.
- (b) Hole positioning data : These are for the positioning of the X (Z) and C axes.  
These are unmodal data, and they are commanded block by block when the same hole machining mode is to be executed continuously.
- (c) Hole machining data : These are the actual machining mode.  
Except for Q, they are modal. Q in the G83 or G87 command is unmodal and is commanded block by block as required.
- (d) Number of repetitions : This number is designated for machining holes at equal intervals when the same cycle is to be repeated.  
The setting range is from 0 to 9999; the decimal point is not valid.  
The number is unmodal and is valid only in the block in which it has been assigned.  
When this number is not designated, it is treated as K1. When K0 is designated, the hole machining data are stored in the memory but no holes will be machined.

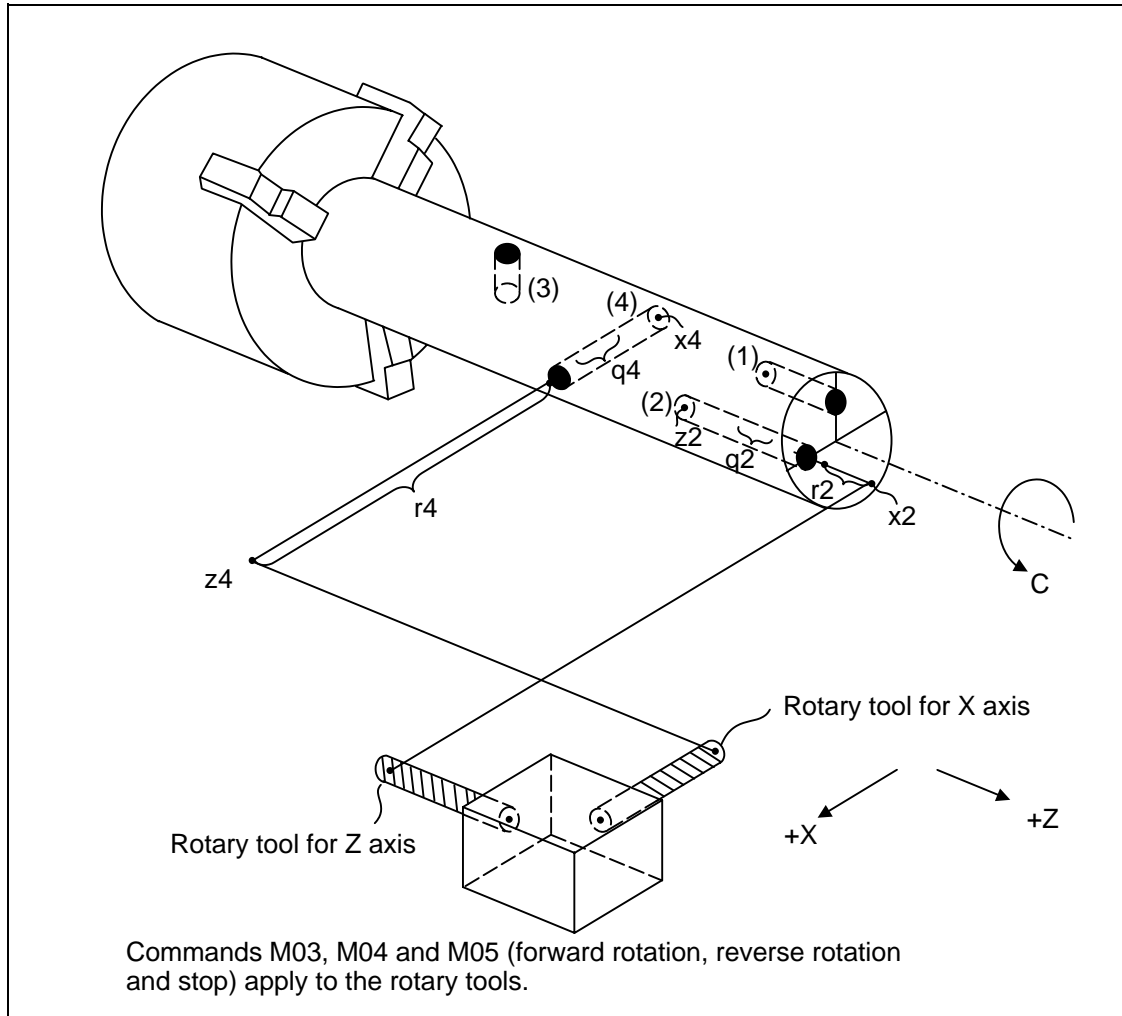
Address	Significance
G	Selection of hole machining cycle sequence (G80, G83, G84, G85, G87, G88, G89)
* X/U (Z/W), C/H	Designation of hole position initial point (absolute/incremental value)
* Z/W (X/U)	Designation of hole bottom position (absolute/incremental value from R point)
R	Designation of R point position (incremental value from initial point) (sign ignored)
Q	Designation of cut amount for each cutting pass with G83 (G87); always incremental value, radius value (sign ignored)
P	Designation of dwell time at hole bottom point; relationship between time and designated value is same as for G04 designation
F	Designation of feedrate for cutting feed
K	Designation of number of repetitions, 0 ~ 9999 (standard value = 1)
D	Designate of spindle used in tap cycle (not used)
S	Designation of spindle rotation speed
,S	Designation of spindle rotation speed at return <b>(Note)</b> Valid only during synchronous tap mode. This setting is ignored during other modes.
,R	Designation of synchronous tapping/asynchronous tapping <b>(Note)</b> The spindle rotation direction can be reversed using the synchronous tap command polarity reversal signal.
M	Designation of miscellaneous command

\* : Addresses in parentheses apply for commands G87, G88 and G89.



### Detailed description

The hole drilling axes for the hole drilling fixed cycle and the positioning are shown in the outline drawing below.

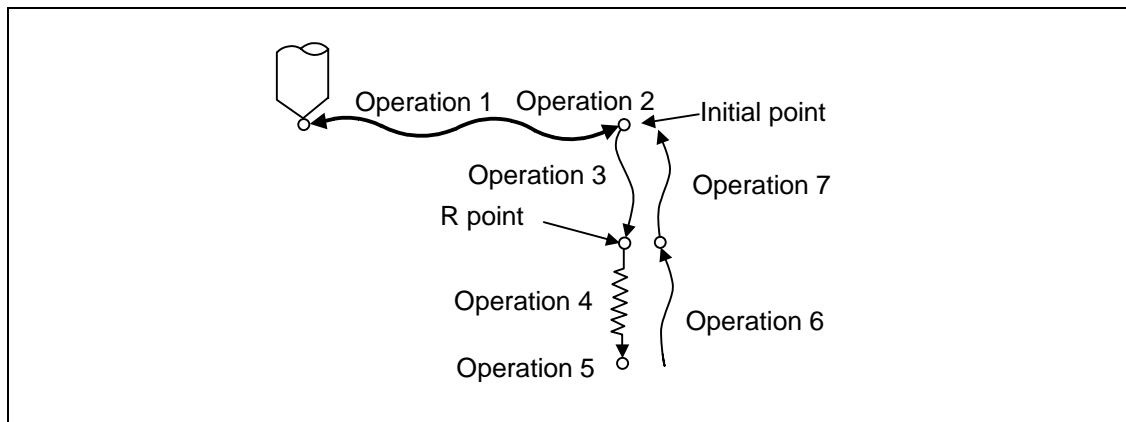


- (1) G83 Xx<sub>1</sub> Cc<sub>1</sub> Zz<sub>1</sub> Rr<sub>1</sub> Qq<sub>1</sub> Pp<sub>1</sub> Ff<sub>1</sub> Kk<sub>1</sub> ;
- (2) G83 Xx<sub>2</sub> Cc<sub>2</sub> Zz<sub>2</sub> Rr<sub>2</sub> Qq<sub>2</sub> Pp<sub>2</sub> Ff<sub>2</sub> Kk<sub>2</sub> ;
- (3) G87 Zz<sub>3</sub> Cc<sub>3</sub> Xx<sub>3</sub> Rr<sub>3</sub> Qq<sub>3</sub> Pp<sub>3</sub> Ff<sub>3</sub> Kk<sub>3</sub> ;
- (4) G87 Zz<sub>4</sub> Cc<sub>4</sub> Xx<sub>4</sub> Rr<sub>4</sub> Qq<sub>4</sub> Pp<sub>4</sub> Ff<sub>4</sub> Kk<sub>4</sub> ;



## Example of operation

There are 7 actual operations which are each described in turn below.



- Operation 1 : This denotes the positioning (by rapid traverse) of the tool to the X (Z) and C axis initial point. If "I" (Positioning axis in-position width) is commanded, in-position check will be carried out after the block completes.
- Operation 2 : This is output provided that the M code for C-axis clamping is issued.
- Operation 3 : The tool is positioned as far as the R point by rapid traverse.
- Operation 4 : Hole machining is conducted by cutting feed.  
If "J" (Hole drilling axis in-position width) is commanded, in-position check will be carried out after the block completes.  
Note that in case of deep drilling cycle 1 or 2, in-position check will not be carried out halfway for the hole drilling. It will be carried out at designated hole bottom position (the last hole drilling).
- Operation 5 : This operation takes place at the hole bottom position and it differs according to the fixed cycle mode. Possible actions include rotary tool reverse rotation (M04), rotary tool forward rotation (M03) and dwell.
- Operation 6 : The tool is retracted as far as the R point.
- Operation 7 : The tool is returned as far as the initial point at the rapid traverse rate.

(Operations 6 and 7 may be a single operation depending on the fixed cycle mode.)

**(Note)** As for the synchronous tapping command, in-position check will be carried out depending on the parameter #1223 (Synchronous tap improvement). (Only some models)

Whether the fixed cycle is to be completed at operation 6 or 7 can be selected by the following G commands.

- G98 ..... Initial level return
- G99 ..... R point level return

These G commands are modal. Once, for instance, G98 is designated, the G98 mode will remain valid until the G99 command is issued. The G98 mode is established in the initialized state when the NC unit is ready to operate.

## 13. Program Support Functions

### 13.5 Hole drilling fixed cycles



#### In-position check at synchronous tapping

Refer to "Example of operation" for in-position check of the other hole drilling cycle (including asynchronous tapping).

#### Setting values of synchronous tap in-position check parameters and tap axis movement

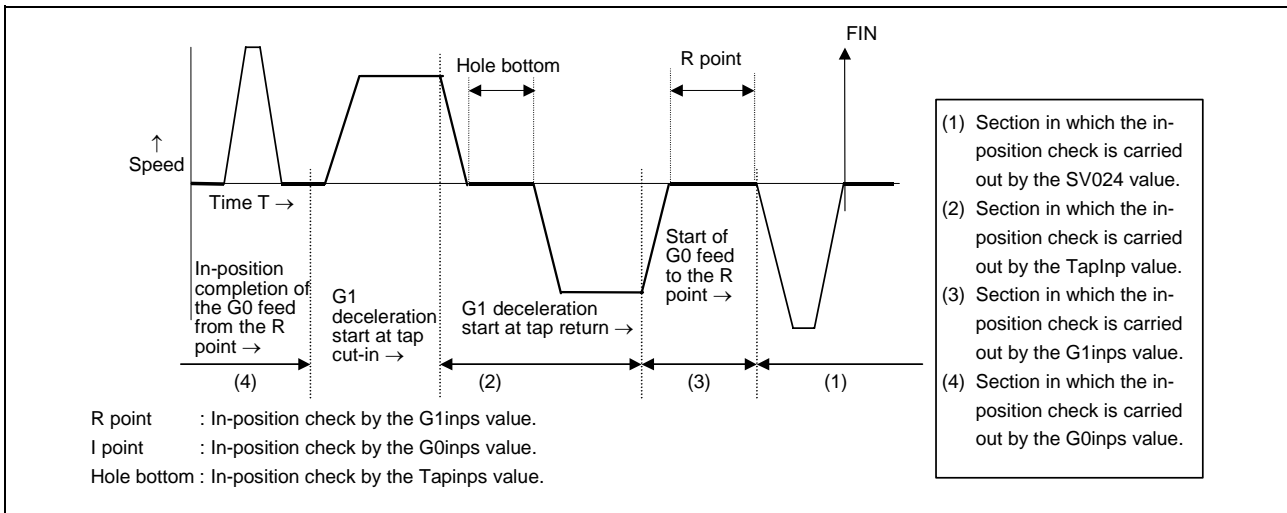
##### (1) Synchronous tap in-position check settings

#1223 aux07				"P" designation of G84/G88 command	In-position check during synchronous tap
bit3	bit4	bit5	bit2		
Synchronous tap in-position check	Hole bottom	R point	I point → R point		
0	-	-	-	-	Carry out in-position check at I point → R point/R point/hole bottom.
1	-	-	-	No "P" designation <b>Example:</b> G84 F1. Z-5. S1000 R-5.	Hole bottom: Do not carry out in-position check. R point: Do not carry out in-position check. I point → R point: Do not carry out in-position check.
1	1	1	1	"P" designation <b>Example:</b> G84 F1. Z-5. S1000 <b>PO</b> R-5.	Hole bottom: Carry out in-position check by tap in-position width. R point: Carry out in-position check. I point → R point: Carry out in-position check.
1	1	0	1	"P" designation <b>Example:</b> G84 F1. Z-5. S1000 <b>PO</b> R-5.	Hole bottom: Carry out in-position check by tap in-position width. R point: Do not carry out in-position check. I point → R point: Carry out in-position check.
1	0	1	1	"P" designation <b>Example:</b> G84 F1. Z-5. S1000 <b>PO</b> R-5.	Hole bottom: Do not carry out in-position check. R point: Carry out in-position check. I point → R point: Carry out in-position check.
1	0	0	1	"P" designation <b>Example:</b> G84 F1. Z-5. S1000 <b>PO</b> R-5.	Hole bottom: Do not carry out in-position check. R point: Do not carry out in-position check. I point → R point: Carry out in-position check.
1	1	1	0	"P" designation <b>Example:</b> G84 F1. Z-5. S1000 <b>PO</b> R-5.	Hole bottom: Carry out in-position check by tap in-position width. R point: Carry out in-position check. I point → R point: Do not carry out in-position check.
1	1	0	0	"P" designation <b>Example:</b> G84 F1. Z-5. S1000 <b>PO</b> R-5.	Hole bottom: Carry out in-position check by tap in-position width. R point: Do not carry out in-position check. I point → R point: Do not carry out in-position check.
1	0	1	0	"P" designation <b>Example:</b> G84 F1. Z-5. S1000 <b>PO</b> R-5.	Hole bottom: Do not carry out in-position check. R point: Carry out in-position check. I point → R point: Do not carry out in-position check.
1	0	0	0	"P" designation <b>Example:</b> G84 F1. Z-5. S1000 <b>PO</b> R-5.	Hole bottom: Do not carry out in-position check. R point: Do not carry out in-position check. I point → R point: Do not carry out in-position check.

**(Note 1)** The I point refers to the initial point.



### (2) In-position width and tap axis movement for a synchronous tap in-position check



## 13. Program Support Functions

### 13.5 Hole drilling fixed cycles

(3) Relation between the parameter setting values and tap axis movement for a synchronous tap in-position check

#1223 aux07				Hole bottom wait time	Operation at hole bottom	Operation at R point	Operation at I point → R point
bit3	bit4	bit5	bit2				
Synchronous tap in-position check	Hole bottom	R point	I point → R point				
0	-	-	-	Time designated by "P". Processing time is several 10ms when no "P".	Operation determined by setting of inpos (#1193) and aux 07 (#1223): bit 1 parameters.	Operation determined by setting of inpos (#1193) and aux 07 (#1223): bit 1 parameters.	Operation determined by setting of inpos (#1193) and aux 07 (#1223): bit 1 parameters.
1	0	0	1	Larger value of "P" and TapDwl (#1313) is valid. No dwell executed if both values are 0.	Wait until time in left column has elapsed.		Wait until completion of in-position check by G0inps value.
1	0	1	1	Larger value of "P" and TapDwl (#1313) is valid. No dwell executed if both values are 0.	Wait until time in left column has elapsed.	Wait until completion of in-position check by G1inps value.	Wait until completion of in-position check by G0inps value.
1	1	0	1	Larger value of "P" and TapDwl (#1313) is valid. No dwell executed if both values are 0.	Wait until dwell time in left column has elapsed after completion of in-position check.		Wait until completion of in-position check by G0inps value.
1	1	1	1	Larger value of "P" and TapDwl (#1313) is valid. Processing time is several 10ms if both values are 0.	Wait until dwell time in left column has elapsed after completion of in-position check.	Wait until completion of in-position check by G1inps value.	Wait until completion of in-position check by G0inps value.
1	0	0	0	Larger value of "P" and TapDwl (#1313) is valid. No dwell executed if both values are 0.	Wait until time in left column has elapsed.		
1	0	1	0	Larger value of "P" and TapDwl (#1313) is valid. No dwell executed if both values are 0.	Wait until time in left column has elapsed.	Wait until completion of in-position check by G1inps value.	
1	1	0	0	Larger value of "P" and TapDwl (#1313) is valid. No dwell executed if both values are 0.	Wait until dwell time in left column has elapsed after completion of in-position check.		
1	1	1	0	Larger value of "P" and TapDwl (#1313) is valid. Processing time is several 10ms if both values are 0.	Wait until dwell time in left column has elapsed after completion of in-position check.	Wait until completion of in-position check by G1inps value.	

**(Note 1)** The I point refers to the initial point.

**(Note 2)** Keep in mind that vibration or deterioration in accuracy may occur when having the in-position check invalidated at R point.

# 13. Program Support Functions

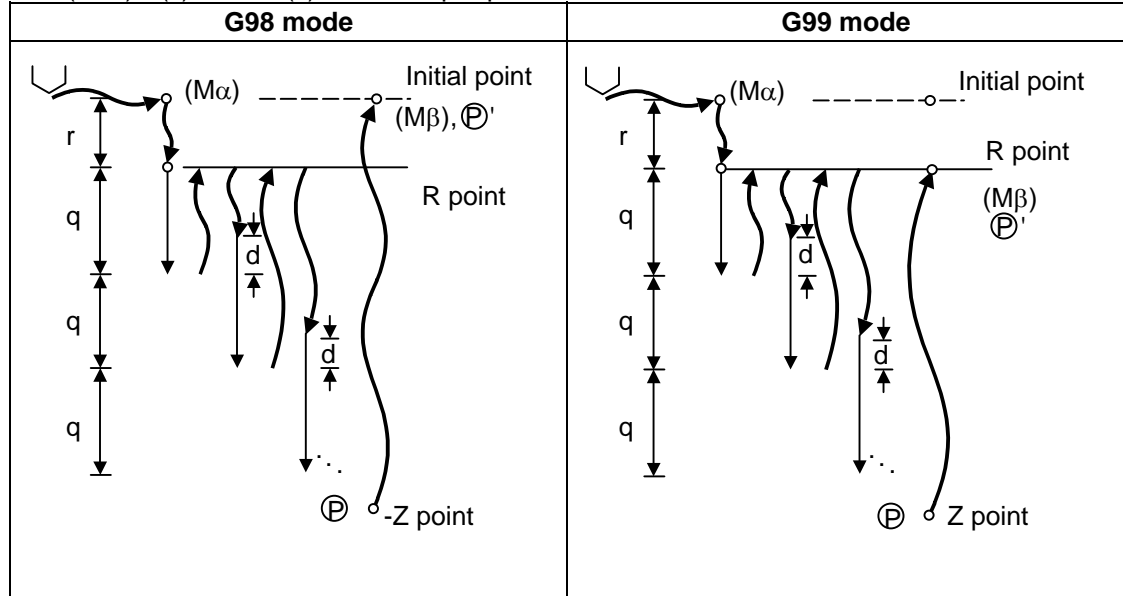
## 13.5 Hole drilling fixed cycles

### 13.5.1 Face deep hole drilling cycle 1; G83 (Longitudinal deep hole drilling cycle 1; G87)



When the Q command is present (deep hole drilling)

G83 (G87) X(z)\_ C\_ Z(x)\_ R\_r Q\_q P\_p F\_f K\_k M\_m ;

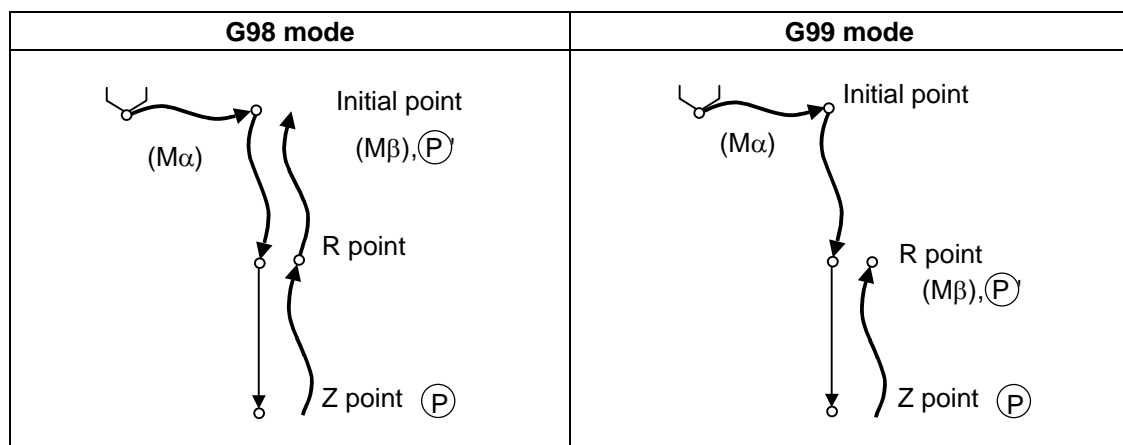


- (1) Return amount "d" is set by the setup parameter (#8013 G83 return). The tool returns at rapid traverse.
- (2)  $(M\alpha)$  : The M code (Mm) is output when there is a C-axis clamping M code command (Mm).
- (3)  $(M\beta)$  : The C-axis unclamping M code (C-axis clamp M code + 1 = Mm + 1) is output when there is a C-axis clamping M code command (Mm).
- (4)  $(P)$  : Dwell is performed for the duration equivalent to the time designated by P.
- (5)  $(P')$  : After the C-axis unclamping M code (Mm+2) has been output, dwell is performed for the duration equivalent to the time set by the parameter (#1184 clmp\_D).



When the Q command is not present (drilling)

G83 (G87) X(z)\_ C\_ Z(x)\_ R\_r P\_p F\_f K\_k M\_m ;



See "When the Q command is present (deep hole drilling)" for details on  $M\alpha$ ,  $M\beta$ ,  $(P)$  and  $(P')$ .

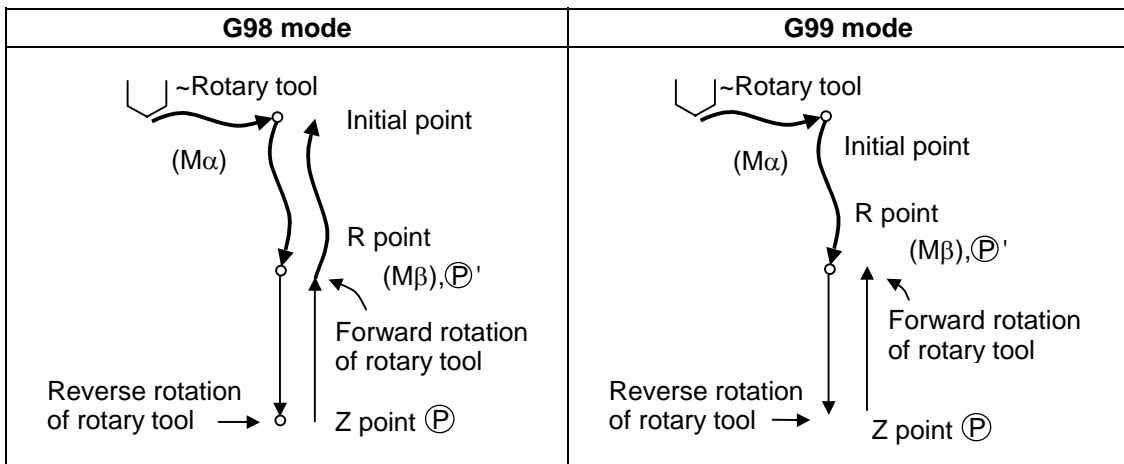
### 13.5.2 Face tapping cycle; G84 (Longitudinal tapping cycle; G88) / Face reverse tapping cycle; G84.1 (Longitudinal reverse tapping cycle; G88.1)



#### Detailed description

G84 (G88) is explained below. With the reverse tapping cycle G84.1 (G88.1), the tap rotation direction is reversed.

G84 (G88)  $Xx1$   $Cc1$   $Zz1$   $Rr1$   $Pp1$   $Ff1$   $Kk1$   $Dd1$   $Ss1$  , $Ss2$  , $Rr2$   $Mm1$  ;



- (1) See Section 13.5.1 "When the Q command is present (deep hole drilling)" for details on  $M\alpha$ ,  $M\beta$ ,  $(P)$  and  $(P)'$ .
- (2) When G84 (G88) is executed, the override will be canceled and the override will automatically be set to 100%.
- (3) Dry run is valid when the control parameter "G00 DRY RUN" is ON and is valid for the positioning command. If the feed hold button is pressed during G84 (G88) execution, block stop occurs after returning completes.
- (4) During single block operation, the axis will not stop at the turning point of tap cycle.
- (5) During the G84 (G88) modal, the "Tapping" NC output signal will be output.
- (6) During the G84 (G88) synchronous tapping modal, the M3, M4, S code, etc. will not be output.
- (7) The tool stops at the reference point and the rotary tool forward rotation signal is output.
- (8) The rotation of the rotary tool is reversed at the hole bottom and tapping is performed.
- (9) Before the rotary tool reverse (M04) or forward rotation (M03) signal is output, the fixed cycle subprograms should be edited if the rotary tool stop (M05) command is required.



### Synchronous/asynchronous tap selection

#### (1) Selecting with a program command

Tap cycle ",R0/1" command

G84 (G88) Xx1 Cc1 Zz1 Rr1 Pp1 Ff1 Kk1 Dd1 Ss1 ,Ss2 ,Rr2 Mm1 ;

When r2 = 1, the synchronous tap mode is entered, and when r2 = 0, the asynchronous tap mode is entered.

#### (2) Selecting with parameters

[Base specifications parameters]

#	Items	Details	Setting range (unit)
1229	set01 bit4	0: Handles tap cycles as the tap cycles with a floating tap chuck. 1: Handles tap cycles as the tap cycles without a floating tap chuck.	0/1

The tap command will be the synchronous tap cycle when this parameter is turned ON.

#### (3) Selecting with M function

[Base specifications parameters]

#	Items	Details	Setting range (unit)
1272 (PR)	ext08 bit1	M function synchronous tap cycle valid	0: Invalid 1: Valid

Synchronous tap cannot be selected with the M function when this parameter is OFF.

[Base specifications parameters]

#	Items	Details	Setting range (unit)
1513	stapM	M code for synchronous tap selection	0 to 99999999

The synchronous tap mode is selected with the M function code set with this parameter. The M function can be commanded in the same block before the tap command.

**(Note)** Do not use M00, 01, 02, 30, 96, 97, 98 or 99.

#### (Example)

M29 ; (M command synchronous tap)

G84 Z50. R20. F2. ; or G84 Z50. R20. F2. M29 ;

The synchronous and asynchronous tap will follow the combination shown below.

	Combination											
	0	0	0	0	1	1	1	1	No command			
Program command (,R0/1)	0	0	1	1	0	0	1	1	0	0	1	1
#1229 (bit4)	x	○	x	○	x	○	x	○	x	○	x	○
M function code (M**)	A	A	A	A	S	S	S	S	A	S	S	S

x : Does not command    A : Asynchronous tap  
○ : Commands            S : Synchronous tap



### Canceling synchronous tap selection

The synchronous tap selection is canceled when the reset, G80 (hole drilling fixed cycle cancel), 01 group G code or other fixed cycle G code is commanded.



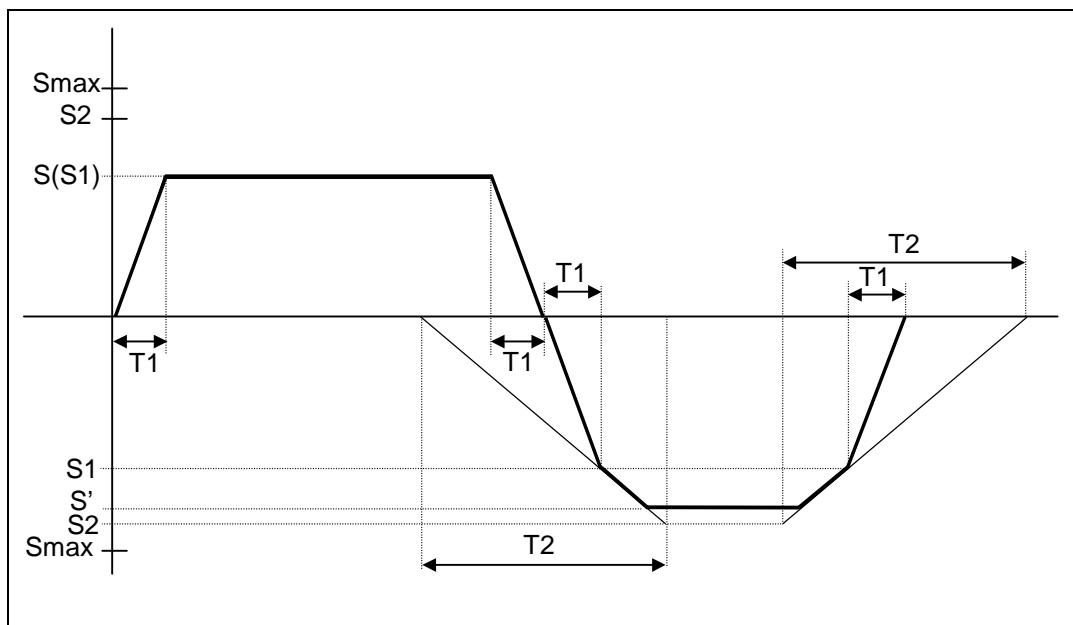
### Spindle acceleration/deceleration pattern during synchronous tap

This function allows spindle acceleration/deceleration pattern to be approached to the speed loop acceleration/deceleration pattern by dividing the spindle and drilling axis acceleration/ deceleration pattern into up to three stages during synchronous tapping.

The acceleration/deceleration pattern can be set up to three stages for each gear.

When returning from the hole bottom, rapid return is possible depending on the spindle rotation speed during return. The spindle rotation speed during return is held as modal information.

- (1) When tap rotation speed < spindle rotation speed during return ≤ synchronous tap changeover spindle rotation speed 2

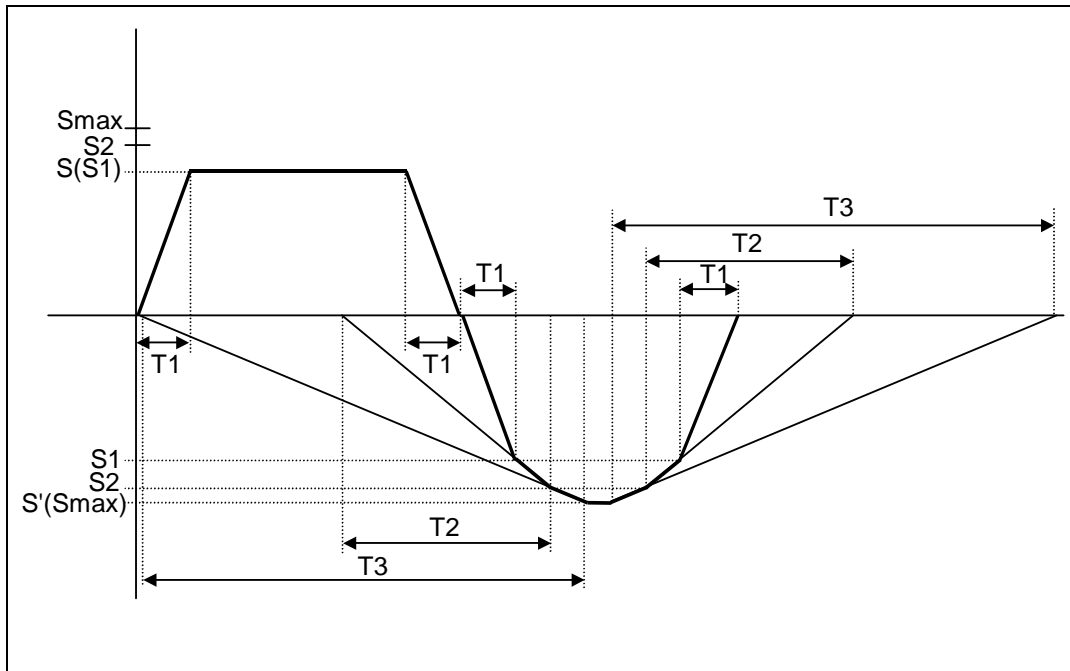


S	: Command spindle rotation speed
S'	: Spindle rotation speed during return
S1	: Tap rotation speed (spindle base specification parameters #3013 to #3016)
S2	: Synchronous tap changeover spindle rotation speed 2 (spindle base specification parameters #3037 to #3040)
Smax	: Maximum rotation speed (spindle base specification parameters #3005 to #3008)
T1	: Tap time constant (spindle base specification parameters #3017 to #3020)
T2	: Synchronous tap changeover time constant 2 (spindle base specification parameters #3041 to #3044)

## 13. Program Support Functions

### 13.5 Hole drilling fixed cycles

- (2) When synchronous tap changeover spindle rotation speed  $2 <$  spindle rotation speed during return



S	: Command spindle rotation speed
S'	: Spindle rotation speed during return
S1	: Tap rotation speed (spindle base specification parameters #3013 to #3016)
S2	: Synchronous tap changeover spindle rotation speed 2 (spindle base specification parameters #3037 to #3040)
Smax	: Maximum rotation speed (spindle base specification parameters #3005 to #3008)
T1	: Tap time constant (spindle base specification parameters #3017 to #3020)
T2	: Synchronous tap changeover time constant 2 (spindle base specification parameters #3041 to #3044)
T3	: Synchronous tap changeover time constant 3 (spindle base specification parameters #3045 to #3048)



#### Precautions for commanding tool spindle No. (Dd1)

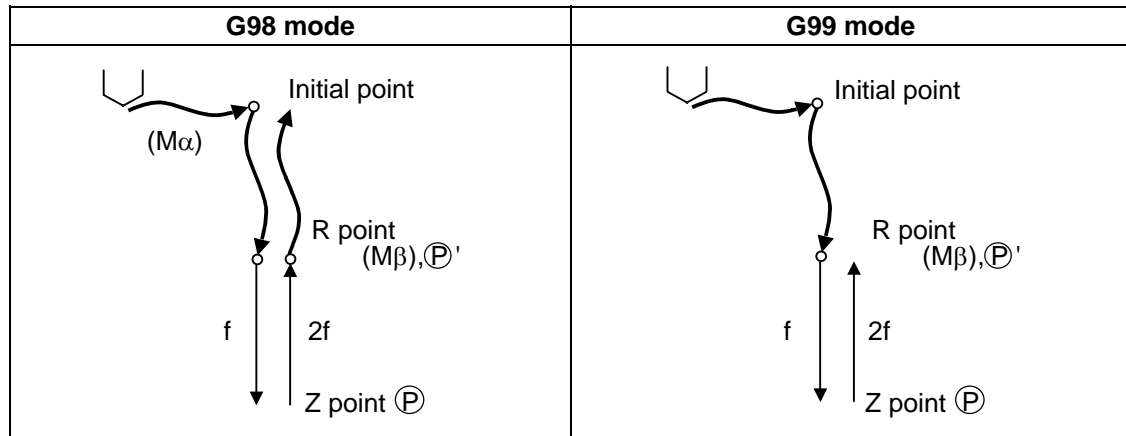
- (1) The command value range is "1 to the number of spindles". A program error (P35) will occur if a value exceeding the range is set.
- (2) When multiple spindle control II is valid, the spindle is selected from the PLC so do not command Dd1. A program error (P32) will occur if commanded.
- (3) When Dd1 is omitted, the spindle designated with the spindle selection command will be selected.

### 13.5.3 Face boring cycle; G85 (Longitudinal boring cycle; G89)



#### Detailed description

G85 (G89) X (z)\_ C\_ Z (x)\_ R\_ P\_ F\_ K\_ M\_ ;



- (1) See Section 13.5.1 "When the Q command is present (deep hole drilling)" for details on  $M\alpha$ ,  $M\beta$ ,  $(P)$  and  $(P)'$ .
- (2) The tool returns to the R point at a cutting feedrate which is double the designated feedrate command. However, it does not exceed the maximum cutting feedrate.

### 13.5.4 Deep hole drilling cycle 2; G83.2



#### Function and purpose

The deep hole drilling cycle 2 drills deep holes in the X-axis or Z-axis direction by commanding the X or Z coordinate of the end point and the cut amount with cutting feed.

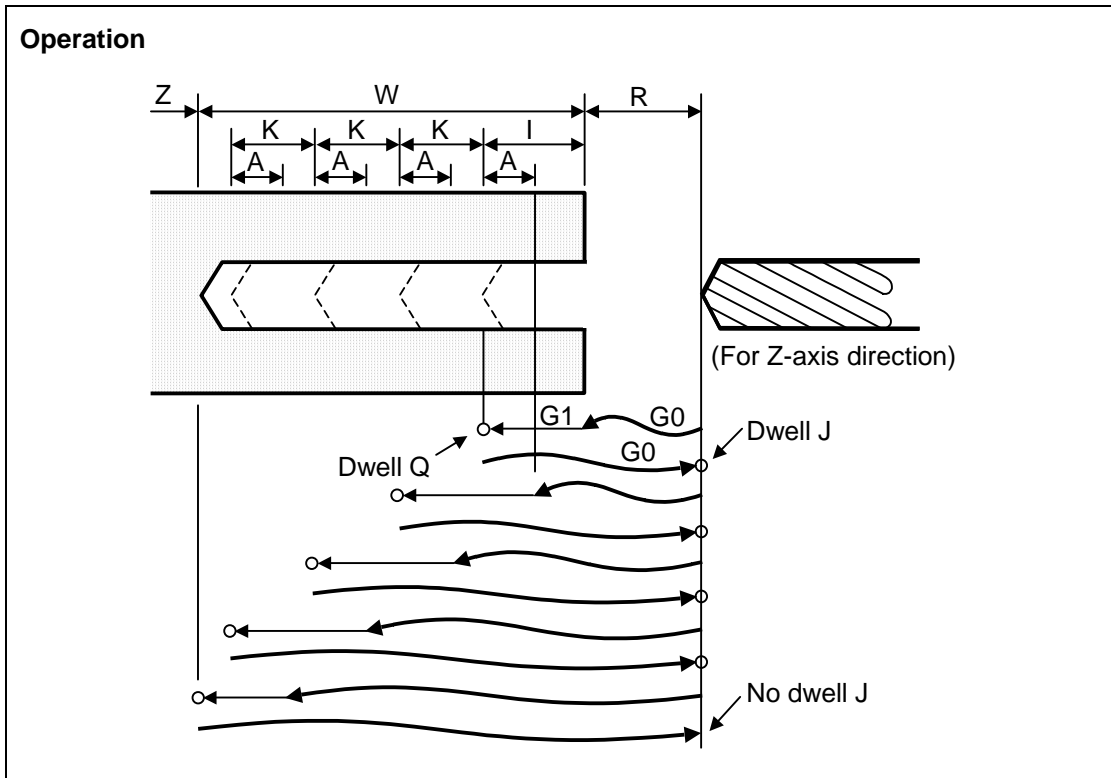


#### Command format

**G83.2 W/Z/U/X\_ R\_ I\_ K\_ A\_ Q\_ J\_ F\_ ;**

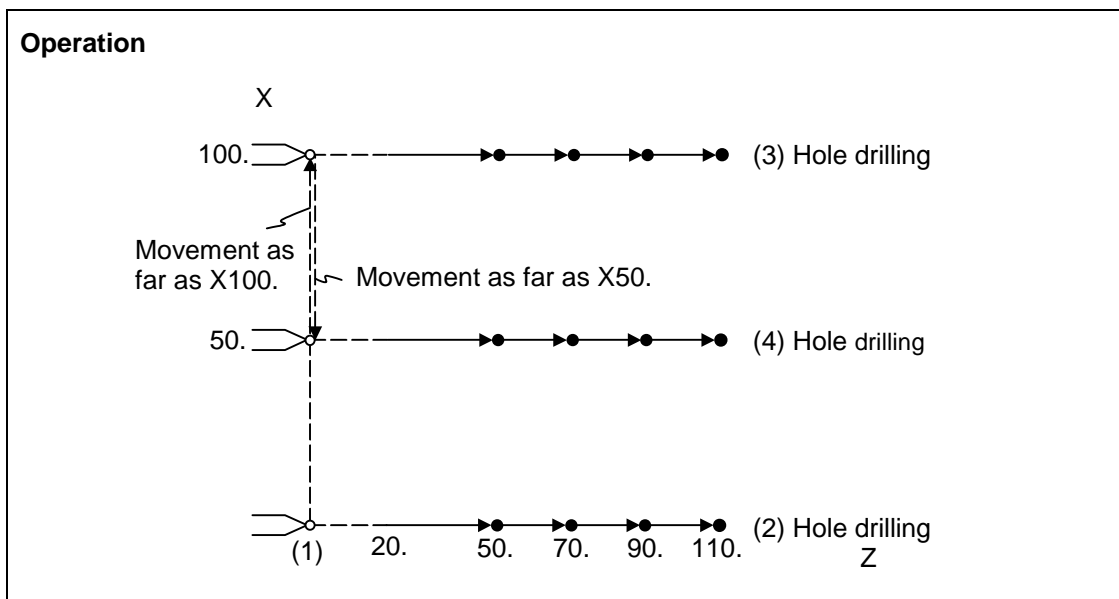
W/Z/U/X	Incremental value from hole drilling start point/coordinates of hole bottom (with sign)
R	Incremental value (sign ignored) from present position up to hole drilling start point, always radius value with incremental value.
I	Cut amount of first cutting pass (sign ignored), always radius value with incremental value.
K	Cut amount of second and subsequent cutting passes (sign ignored), always radial value with incremental value.
A	Drill stop safety distance for second and subsequent cutting passes (sign ignored), always radial value with incremental value.
Q	Dwell time at cut point (sign ignored, decimal point invalid)
J	Dwell time at return point (sign ignored, decimal point invalid)
F	Cutting feedrate





Example of program (when deep hole drilling cycle 2 is used as a modal command)

G28 XZ ;
G0 X0. Z0. ;..... (1)
G83.2 Z110. R20. I30. K20. A5. Q1000 J500. F300. ; ..... (2)
X100. ;..... (3)
X50. ;..... (4)
M02 ;





### Detailed description

- (1) When the drill stop safety distance (address "A") command is not given, the parameter #8013 return parameter setting value is used.
- (2) The deep hole drilling cycle 2 is a modal command and so it will remain valid until a command in the same modal group or the cancel command (G80) is issued.
- (3) If the command either for the cut amount (address "I") for the first cutting pass or the cut amount (address "K") for the second and subsequent passes is not present (including a command value of zero), the command value which is present will be used and the operation will be executed with both I and K equal to the command value.  
If both commands are not present, hole drilling is conducted once as far as the hole bottom.
- (4) When the axis address of the hole drilling axes has been commanded a multiple number of times in a block, the address commanded last is valid.
- (5) The hole drilling fixed cycle 2 is also canceled by any G code in the 01 group except G80. If it is commanded in the same block as the fixed cycle, the fixed cycle will be ignored.  
m = 01 group code, n = hole drilling fixed cycle code

(a)  $\underbrace{G_m}_{\text{Executed}} \underbrace{G_n}_{\text{Ignored}} \underbrace{X(z) \quad C \quad Z(x)}_{\text{Executed}} \underbrace{R \quad I \quad K \quad A \quad Q \quad J}_{\text{Ignored}} \underbrace{F}_{\text{Memorized}} ;$

(b)  $\underbrace{G_n}_{\text{Ignored}} \underbrace{G_m \quad X(z) \quad C \quad Z(x)}_{\text{Executed}} \underbrace{R \quad I \quad K \quad A \quad Q \quad J}_{\text{Ignored}} \underbrace{F}_{\text{Memorized}} ;$

**(Examples)** G01 G83.2 Z50. R-10. I8. K10. A3. Q1000 J500 F100. ;  
G83.2 G01 Z50. R-10. I8. K10. A3. Q1000 J500 F100. ;  
In both cases, G01 Z50. F100. is executed.

- (6) When a miscellaneous command is issued in the same block as the deep hole drilling cycle 2 command, it is output at the same time as the initial positioning.
- (7) When a tool length offset command (T function) is issued in the deep hole drilling cycle 2 mode, execution will follow the tool length offset function.
- (8) If the basic axis, additional axis or R data are in the block during the deep hole drilling cycle 2 mode, hole drilling is performed; if the data are not present, no holes are machined.  
Even if the X-axis data are present, no holes will be machined if the dwell (G04) time command is present in the block.
- (9) Command the hole machining data (A, I, K, Q, J) in the block (including the basic axis, additional axis or R data) in which the hole drilling operation is conducted.  
Even if they are commanded in a block with no hole drilling operation, the modal data will not be updated.
- (10) A program error (P33) results with the following commands.
  - (a) When both the X hole drilling axis (command address X or U) and the Z hole drilling axis (command address Z or W) have been commanded.
  - (b) When any axis except X or Z (any command address except X, U, Z and W) has been commanded.

- (11) When the feed hold button is pressed while the deep hole drilling cycle 2 is being executed, feed hold results at that point, and when automatic operation is restarted, the remainder is executed.
- (12) When an interrupt based on manual operation is performed during the feed hold (manual absolute switch ON), the operation in the deep hole drilling cycle 2 modal is shifted from the automatic operation restart by an amount equivalent to the movement caused by the interrupt.
- (13) With single block operation, block stop results upon completion of the deep hole drilling cycle 2 command.

#### 13.5.5 Hole drilling fixed cycle cancel; G80



##### Detailed description

This cancels the hole drilling fixed cycle (G83, G84, G85, G87, G88, G89). The hole machining mode and hole machining data are both canceled.

#### 13.5.6 Precautions when using a hole drilling fixed cycle



##### Precautions

- (1) When a G84 or G88 fixed cycle is commanded, the rotary tool must be rotated in the specified direction beforehand by the previous miscellaneous functions (M3, M4).
- (2) If there is data for the basic axis, additional axis or R in the same block during fixed cycle mode, hole drilling is carried out. If there is no data, the hole drilling is not carried out. Note that even with X axis data, if that block is a dwell (G04) time command, the hole drilling is not carried out.
- (3) Command hole machining data (Q, P) in the block in which hole drilling is carried out (block containing data for the basic axis, additional axis or R). The modal data is not updated in blocks without hole drilling, even if these data are commanded.
- (4) The F modal may change when a reset is applied during execution of G85 (G89).

## 13. Program Support Functions

### 13.5 Hole drilling fixed cycles

- (5) The hole drilling fixed cycle is also canceled by 01 group G codes other than G80. If these are commanded in the same block as the fixed cycle, the fixed cycle will be ignored.  
 m = 01 group code    n = hole drilling fixed cycle code

(a)  $\underbrace{G_m}_{\text{Executed}} \underbrace{G_n}_{\text{Ignored}} \underbrace{X(z)\_ \ C\_ \ Z(x)\_}_{\text{Executed}} \underbrace{R\_ \ Q\_ \ P\_ \ K\_}_{\text{Ignored}} \underbrace{F\_}_{\text{Memorized}};$

(b)  $\underbrace{G_n}_{\text{Ignored}} \underbrace{G_m \ X(z)\_ \ C\_ \ Z(x)\_}_{\text{Executed}} \underbrace{R\_ \ Q\_ \ P\_ \ K\_}_{\text{Ignored}} \underbrace{F\_}_{\text{Memorized}};$

**(Examples)** G01 G83 X100. C30. Z50. R-10. Q10. P1 F100. ;  
 G83 G01 X100. C30. Z50. R-10. Q10. P1 F100. ;  
 In both cases, G01X100. C30. Z50. F100 is executed..

- (6) When miscellaneous functions are commanded in the same block as fixed cycle commands, they are output simultaneously with the initial positioning.  
 Note that when the M code of the C axis clamp set in the parameters (#1183clmp\_M) is commanded in the same block, the M code is output after positioning (Operation 2).  
 Once the axis has returned to the return point (G98 mode: initial point/G99 mode: R point) after hole drilling, the M code of the C axis clamp (clamp M + 1) is output, and the axis dwells for the time set in the parameters (#1184 clmp\_D).  
 When the No. of rotations is designated, M codes other than that of the C axis clamp carry out the control above for the first rotation only.  
 C axis clamp/unclamp M commands work in modal, and are output every rotation until canceled by the fixed cycle cancel command.
- (7) When tool length offset commands (T functions) are commanded in the hole drilling fixed cycle, they are executed according to the tool length offset function.
- (8) A program error (P155) will result when a hole drilling fixed cycle is commanded during nose R compensation.
- (9) For the G code list 1, the initial point level return is fixed. The return level cannot be changed with a G98/G99 command. Note that a separate function will be executed if G98/G99 is commanded.
- (10) In a block where the movement direction of any axis reverses as shown below, the servo system load will greatly increase so do not command the in-position width in the machining program.  
 G0 X100., I10.0 ;  
 X-200. ;

## 13. Program Support Functions

### 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)

#### 13.6 Hole Drilling Fixed Cycle; G80~G89 (MELDAS Special Format)

[E68]



#### Function and purpose

These fixed cycles are used for predetermined sequences of machining operations such as normal positioning, hole drilling, boring and tapping which are specified in a block. The various sequences available using these functions are listed in the table below.

This function is valid when the MELDAS special format is added (#1265 ext01/bit2 is ON).

G code	Hole machining start (-Z direction)	Operation at hole bottom		Return operation (+Z direction)	Application
		Dwell	Spindle		
G80	-	-	-	-	Cancel
G81	Cutting feed	-	-	Rapid traverse	Drill, spot drilling cycle
G82	Cutting feed	Yes	-	Rapid traverse	Drill, counter boring cycle
G83	Intermittent feed	Yes	-	Rapid traverse	Deep hole drilling cycle
G83.1	Intermittent feed	Yes	-	Rapid traverse	Stepping cycle
G84	Cutting feed	Yes	Reverse rotation	Cutting feed	Tapping cycle
G84.2	Cutting feed	Yes	Reverse rotation	Cutting feed	Synchronous tap cycle
G85	Cutting feed	-	-	Cutting feed	Boring cycle
G89	Cutting feed	Yes	-	Cutting feed	Boring cycle

A fixed cycle mode is canceled when the G80 or any G command in (G00, G01, G02, G03) is issued. The various data will also be cleared simultaneously to zero.



#### Command format

**G8Δ X\_ Y\_ Z\_ R\_ Q\_ P\_ F\_ L\_ S\_ , S\_ ,I\_ ,J\_**

G8Δ	Hole machining mode
X_ Y_ Z_	Hole positioning data
R_ Q_ P_ F_	Hole machining data
L_	Number of repetitions
S_	Spindle rotation speed
,S_	Spindle rotation speed during return
,I_	Positioning axis in-position width
,J_	Drilling axis in-position width

As shown above, the format is divided into the hole machining mode, hole positioning data, hole machining data, No. of repetitions, spindle rotation speed, spindle rotation speed during return, positioning axis in-position width and drilling axis in-position width.

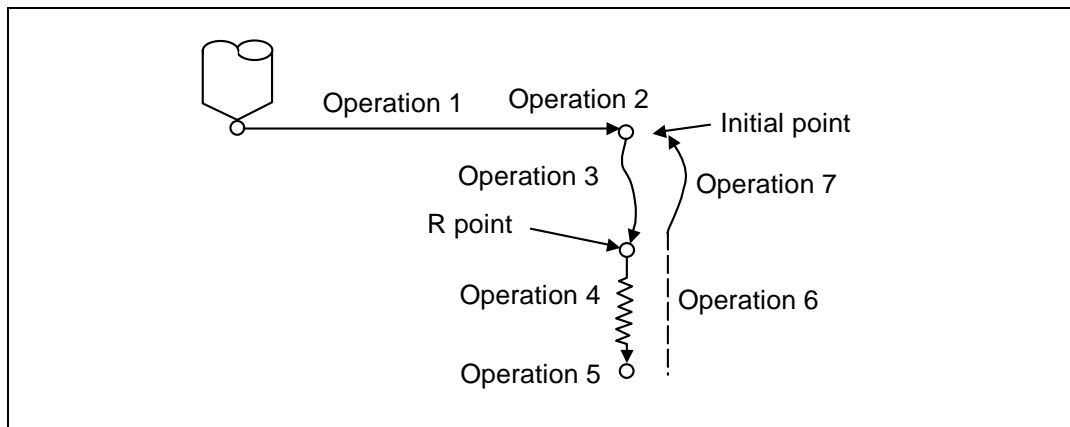
## 13. Program Support Functions

### 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)



#### Detailed description

- (1) The hole machining mode refers to the fixed cycle modes such as drilling, counter boring, tapping and boring. The hole positioning data is the data used to position the X and Y axes. The hole machining data is the machining data actually used for machining. The hole positioning data and No. of repetitions are unmodals and the hole machining data and spindle rotation speed during return are modals.
- (2) If M00 or M01 is commanded in the same block as the fixed cycle or during the fixed cycle mode, the fixed cycle will be ignored. Instead, M00 and M01 will be output after positioning. The fixed cycle will be executed if X, Y, Z or R is commanded.
- (3) There are 7 actual operations which are each described in turn below.  
Operation 1 : This indicates the X and Y axes positioning, and executes positioning with G00.  
Operation 2 : This is an operation done after positioning is completed (at the initial point), and when G87 is commanded, the M19 command is output from the control unit to the machine. When this M command is executed and the finish signal (FIN) is received by the control unit, the next operation will start. If the single block stop switch is ON, the block will stop after positioning.



Operation 3 : The tool is positioned to the R point by rapid traverse.

Operation 4 : Hole machining is conducted by cutting feed.

Operation 5 : This operation takes place at the hole bottom position and it differs according to the fixed cycle mode. Possible actions include spindle stop (M05) spindle reverse rotation (M04), spindle forward rotation (M03), dwell and tool shift.

Operation 6 : Escaping to the R point may take place with either cutting feed or rapid traverse depending on the fixed cycle mode.

Operation 7 : The tool is returned to the initial point at the rapid traverse rate.

Whether the fixed cycle is to be completed at operation 6 or 7 can be selected by the following G commands.

G98 ..... Initial level return  
G99 ..... R point level return

These are modal commands, and for example, if G98 is commanded once, the G98 mode will be entered until G99 is designated. The initial state when the NC is ready is the G98 mode. The hole machining data will be ignored if X, Y, Z or R is not commanded. This function is mainly used with the special fixed cycle.

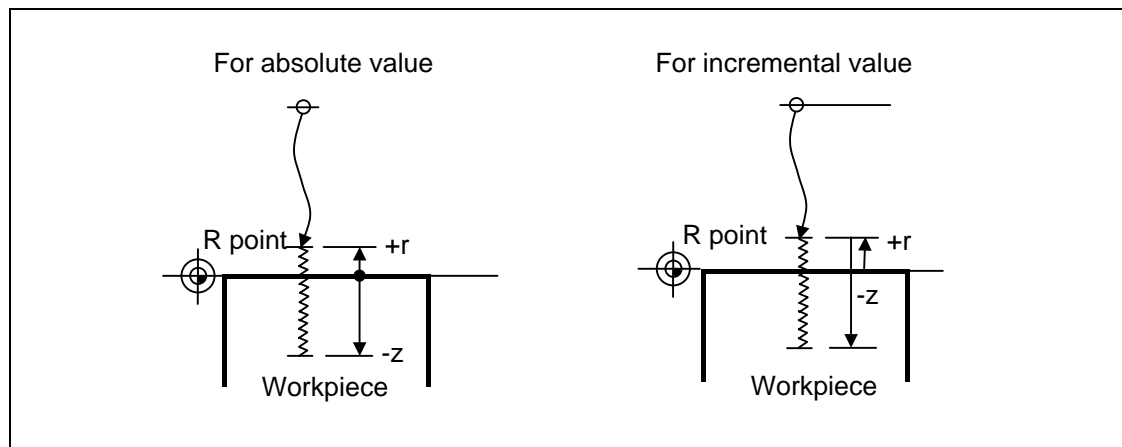
## 13. Program Support Functions

### 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)

(4) Fixed cycle addresses and meanings

Address	Significance
G	Selection of hole drilling fixed cycle sequence (G80 to G89)
X	Designation of hole drilling position (absolute value or incremental value)
Y	Designation of hole drilling position (absolute value or incremental value)
Z	Designation of hole bottom position (absolute value or incremental value)
P	Designation of dwell time at hole bottom position (decimal points will be ignored)
Q	Designation of cut amount for each cutting pass with G83, or designation of the shift amount at G87 (incremental value)
R	Designation of R point position (absolute value)
F	Designation of feedrate for cutting feed
L	Designation of number of repetitions. 0 to 9999
S	Designation of spindle rotation speed
,S	Designation of spindle rotation speed at return <b>(Note)</b> Valid only during synchronous tap mode. This setting is ignored during other modes.
,I	In-position width (position error amount)
,J	In-position width for hole drilling axis in fixed cycle (position error amount)

(5) Difference between absolute value command and incremental value command



(6) Feedrate for tapping cycle and tapping return

The feedrate for the tapping cycle and tapping return are as shown below.

(a) Selection of asynchronous tapping cycle feedrate

G94/G95	Control parameter F1-digit value	F command value	Feed designation
G94	OFF	F designation	Per-minute feed
	ON	Other than F0 to F8	
			F0 to F8 (no decimal point)
G95	-	F designation	Per-rotation feed

- is irrelevant to the setting

## 13. Program Support Functions

### 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)

(b) Spindle rotation speed during return of synchronous tapping cycle

Address	Meaning of address	Command range (unit)	Remarks
,S	Spindle rotation speed during return	0 to 99999 (r/min)	The data is held as modal information. If the value is smaller than the spindle rotation speed, the spindle rotation speed value will be valid even during return. If the spindle rotation speed is not 0 during return, the tap return override value will be invalid.



#### Positioning plane and hole drilling axis

The fixed cycle has basic control elements for the positioning plane and hole drilling axis. The positioning plane is determined by the G17, G18 and G19 plane selection command, and the hole drilling axis is the axis perpendicular (X, Y, Z or parallel axis) to the above plane.

Plane selection	Positioning plane	Hole drilling axis
G17 (X - Y)	Xp - Yp	Zp
G18 (Z - X)	Zp - Xp	Yp
G19 (Y - Z)	Yp - Zp	Xp

Xp, Yp and Zp indicate the basic axes X, Y and Z or an axis parallel to the basic axis.

A random axis other than the hole drilling axis can be commanded for positioning. The hole drilling axis is determined by the axis address of the hole drilling axis commanded in the same block as G81 to G89. The basic axis will be the hole drilling axis if there is no designation.

**(Example 1)** When G17 (XY plane) is selected, and the axis parallel to the Z axis is set as the W axis.

G81 ... Z\_\_ ;      The Z axis is used as the hole drilling axis.  
G81 ... W\_\_ ;      The W axis is used as the hole drilling axis.  
G81 ... ;          (No Z or W) The Z axis is used as the hole drilling axis.

**(Note 1)** The hole drilling axis can be fixed to the Z axis with parameter #1080 Dril\_Z.

**(Note 2)** Change over the hole drilling axis in the fixed cycle canceled state.

**(Note 3)** A program error will occur if the hole drilling axis does not exist.

In the following explanations on the movement in each fixed cycle mode, the XY plane is used for the positioning plane and the Z axis for the hole drilling axis.

Note that all command values will be incremental values, the positioning plane will be the XY plane and the hole drilling axis will be the Z axis.



# 13. Program Support Functions

## 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)



### Programmable in-position width command in fixed cycle

This command commands the in-position width for the fixed cycle from the machining program. The commanded in-position width is valid only in the G81 (drill, spot drill), G82 (drill, counter boring), G83 (deep drill cycle), G84 (tap cycle), G85 (boring), G89 (boring), G83.1 (step cycle) and G84.2 (synchronous tap cycle) fixed cycles. The ",I" address is commanded in respect to the positioning axis, and the ",J" address is commanded in respect to the hole drilling axis.

Address	Meaning of address	Command range (unit)	Remarks
,I	In-position width (position error amount)	1 to 999.999 (mm) 1 $\mu$ m unit	When a value exceeding the command range is issued, a program error will occur. Program error (P35)
,J	In-position width for hole drilling axis in fixed cycle (position error amount)		



### In-position check in fixed cycle

When the number of repetitions L is designated as two or more times in the fixed cycle, the commanded in-position width will be valid in the repetition block (operation 5 to operation 8).

:  
G81 X-50. Z-50. R-50. L2 F2000 ,I0.2 ,J0.3 ;  
:

Operation pattern	,I	,J
Operation 1	Valid	–
Operation 2	–	Invalid
Operation 3	–	Invalid
Operation 4	–	Valid
Operation 5	Valid	–
Operation 6	–	Invalid
Operation 7	–	Invalid
Operation 8	–	Valid

**Operation when number of repetitions L is designated**

In the following machining program, the commanded in-position width is valid for the following Fig. block.

In the (B) block, the in-position width (,I) commanded for positioning in the previous block (A) is invalid (operation 5). However, when returning from the hole bottom, the in-position width (,J) commanded in the previous block (A) is valid (operation 8).

# 13. Program Support Functions

## 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)

In the following machining program, the commanded in-position width is valid for the following Fig. block.

In the (B) block, the in-position width (, I) commanded for positioning in the previous block (A) is invalid (operation 5). However, when returning from the hole bottom, the in-position width (, J) commanded in the previous block (A) is valid (operation 8).

To validate the in-position width for positioning, command again as shown in block (C) (Operation 9).

```

:
G81 X-50. Z-50. R-50. F2000 ,I0.2 ,J0.3 ;..... (A)
X-10. ; ..... (B)
X-10. ,I0.2 ; ..... (C)
:

```

Operation pattern	,I	,J
Operation 1	Valid	–
Operation 2	–	Invalid
Operation 3	–	Invalid
Operation 4	–	Valid
Operation 5	Invalid	–
Operation 6	–	Invalid
Operation 7	–	Invalid
Operation 8	–	Valid
Operation 9	Valid	–
Operation 10	–	Invalid
Operation 11	–	Invalid
Operation 12	–	Valid

Operation in fixed cycle modal

# 13. Program Support Functions

## 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)



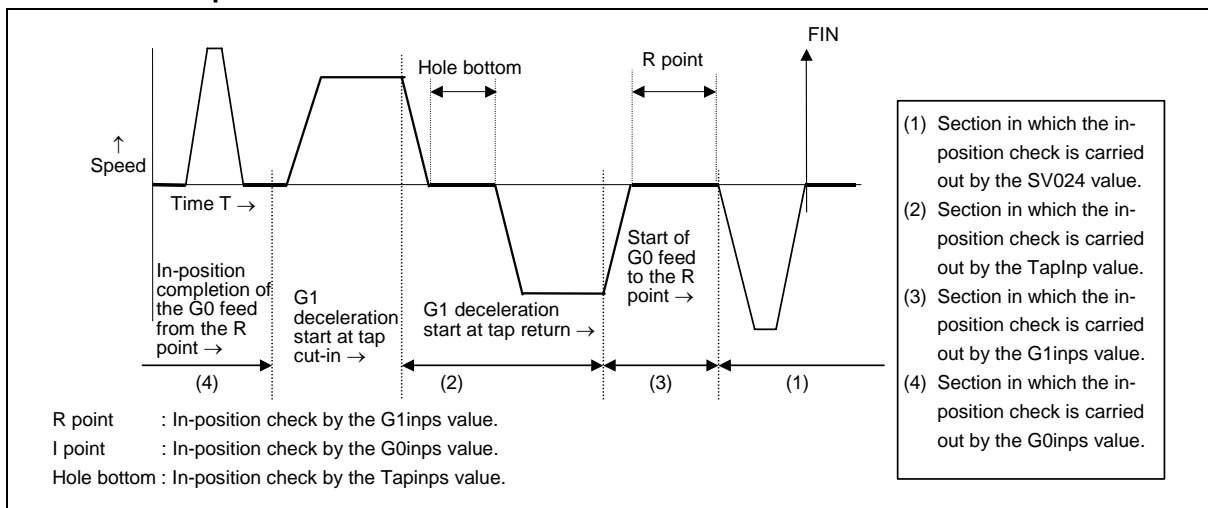
### Setting values of synchronous tap in-position check parameters and tap axis movement

#### (1) Synchronous tap in-position check settings

#1223 aux07				"P" designation of G84/G74 command	In-position check during synchronous tap
bit3	bit4	bit5	bit2		
Synchronous tap in-position check	Hole bottom	R point	I point → R point		
0	—	—	—	—	Carry out in-position check at I point → R point/R point/hole bottom.
1	—	—	—	No "P" designation Example: G84 F1. Z-5. S1000 R-5.	Hole bottom: Do not carry out in-position check. R point: Do not carry out in-position check. I point → R point: Do not carry out in-position check.
1	1	1	1	"P" designation Example: G84 F1. Z-5. S1000 PO R-5.	Hole bottom: Carry out in-position check by tap in-position width. R point: Carry out in-position check. I point → R point: Carry out in-position check.
1	1	0	1	"P" designation Example: G84 F1. Z-5. S1000 PO R-5.	Hole bottom: Carry out in-position check by tap in-position width. R point: Do not carry out in-position check. I point → R point: Carry out in-position check.
1	0	1	1	"P" designation Example: G84 F1. Z-5. S1000 PO R-5.	Hole bottom: Do not carry out in-position check. R point: Carry out in-position check. I point → R point: Carry out in-position check.
1	0	0	1	"P" designation Example: G84 F1. Z-5. S1000 PO R-5.	Hole bottom: Do not carry out in-position check. R point: Do not carry out in-position check. I point → R point: Carry out in-position check.
1	1	1	0	"P" designation Example: G84 F1. Z-5. S1000 PO R-5.	Hole bottom: Carry out in-position check by tap in-position width. R point: Carry out in-position check. I point → R point: Do not carry out in-position check.
1	1	0	0	"P" designation Example: G84 F1. Z-5. S1000 PO R-5.	Hole bottom: Carry out in-position check by tap in-position width. R point: Do not carry out in-position check. I point → R point: Do not carry out in-position check.
1	0	1	0	"P" designation Example: G84 F1. Z-5. S1000 PO R-5.	Hole bottom: Do not carry out in-position check. R point: Carry out in-position check. I point → R point: Do not carry out in-position check.
1	0	0	0	"P" designation Example: G84 F1. Z-5. S1000 PO R-5.	Hole bottom: Do not carry out in-position check. R point: Do not carry out in-position check. I point → R point: Do not carry out in-position check.

(Note 1) The I point refers to the initial point.

#### (2) Relation between the in-position width and tap axis movement for a synchronous tap in-position check



## 13. Program Support Functions

### 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)

#### (3) Relation between the parameter setting values and tap axis movement for a synchronous tap in-position check

#1223 aux07				Hole bottom wait time	Operation at hole bottom	Operation at R point	Operation at I point → R point
bit3	bit4	bit5	bit2				
Synchronous tap in-position check	Hole bottom	R point	I point → R point				
0	—	—	—	Time designated by "P". Processing time is several 10ms when no "P".	Operation determined by setting of inpos (#1193) and aux 07 (#1223): bit 1 parameters.	Operation determined by setting of inpos (#1193) and aux 07 (#1223): bit 1 parameters.	Operation determined by setting of inpos (#1193) and aux 07 (#1223): bit 1 parameters.
1	0	0	1	Larger value of "P" and TapDwl (#1313) is valid. No dwell executed if both values are 0.	Wait until time in left column has elapsed.		Wait until completion of in-position check by G0inps value.
1	0	1	1	Larger value of "P" and TapDwl (#1313) is valid. No dwell executed if both values are 0.	Wait until time in left column has elapsed.	Wait until completion of in-position check by G1inps value.	Wait until completion of in-position check by G0inps value.
1	1	0	1	Larger value of "P" and TapDwl (#1313) is valid. No dwell executed if both values are 0.	Wait until dwell time in left column has elapsed after completion of in-position check.		Wait until completion of in-position check by G0inps value.
1	1	1	1	Larger value of "P" and TapDwl (#1313) is valid. Processing time is several 10ms if both values are 0.	Wait until dwell time in left column has elapsed after completion of in-position check.	Wait until completion of in-position check by G1inps value.	Wait until completion of in-position check by G0inps value.
1	0	0	0	Larger value of "P" and TapDwl (#1313) is valid. No dwell executed if both values are 0.	Wait until time in left column has elapsed.		
1	0	1	0	Larger value of "P" and TapDwl (#1313) is valid. No dwell executed if both values are 0.	Wait until time in left column has elapsed.	Wait until completion of in-position check by G1inps value.	
1	1	0	0	Larger value of "P" and TapDwl (#1313) is valid. No dwell executed if both values are 0.	Wait until dwell time in left column has elapsed after completion of in-position check.		
1	1	1	0	Larger value of "P" and TapDwl (#1313) is valid. Processing time is several 10ms if both values are 0.	Wait until dwell time in left column has elapsed after completion of in-position check.	Wait until completion of in-position check by G1inps value.	

**(Note 1)** The I point refers to the initial point.

**(Note 2)** Keep in mind that vibration or deterioration in accuracy may occur when having the in-position check invalidated at R point.

## 13. Program Support Functions

### 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)

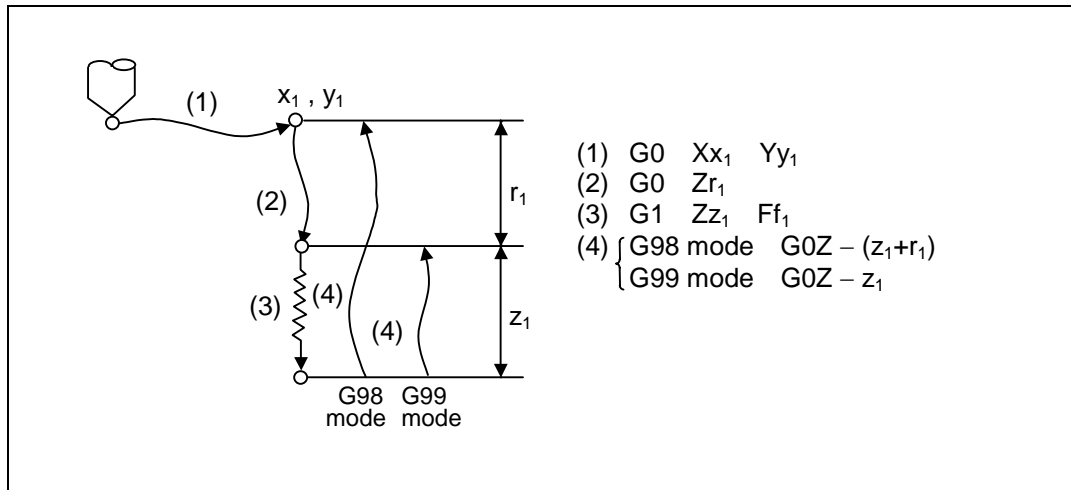


Operations when each fixed cycle is executed

(a) G81 (Drilling, spot drilling)

Program

G81  $Xx_1$   $Yy_1$   $Zz_1$   $Rr_1$   $Ff_1$   $,li_1$   $,Jj_1$  ;



The operation stops at after the (1), (2) and (4) commands during single block operation.

Operation pattern	i1	j1
Operation (1)	Valid	-
Operation (2)	-	Invalid
Operation (3)	-	Invalid
Operation (4)	-	Valid

## 13. Program Support Functions

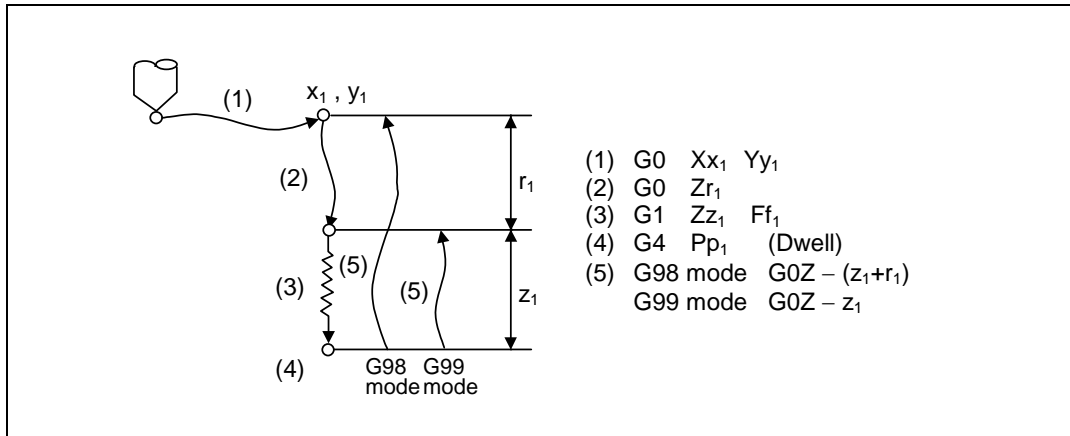
### 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)

(b) G82 (Drilling, counter boring)

Program

G82  $Xx_1 Yy_1 Zz_1 Rr_1 Ff_1 Pp_1 ,li_1 ,Jj_1 ;$

P : Dwell designation



Operation pattern	i1	j1
Operation (1)	Valid	-
Operation (2)	-	Invalid
Operation (3)	-	Invalid
Operation (4)	-	-
Operation (5)	-	Valid

The operation stops at after the (1), (2) and (5) commands during single block operation.

# 13. Program Support Functions

## 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)

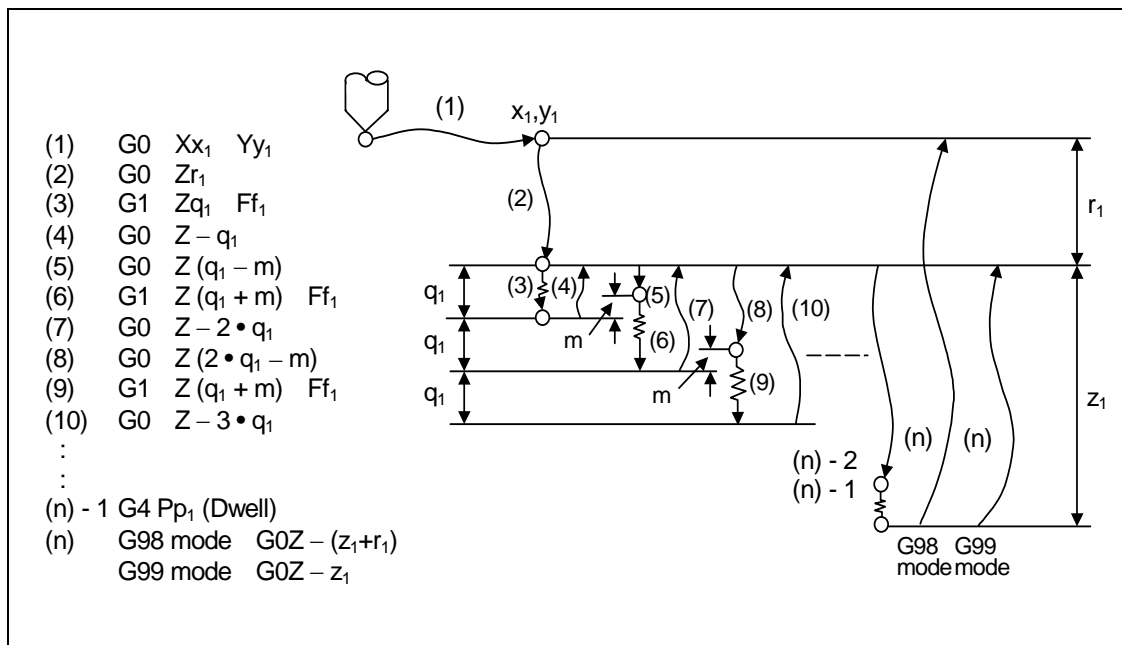
(c) G83 (Deep hole drilling cycle)

Program

G83 Xx<sub>1</sub> Yy<sub>1</sub> Zz<sub>1</sub> Rr<sub>1</sub> Qq<sub>1</sub> Pp<sub>1</sub> Ff<sub>1</sub> ,li<sub>1</sub> ,Jj<sub>1</sub> ;

Q : This designates the cutting amount per pass, and is always designated with an incremental value.

P : Dwell time at hole bottom



Operation pattern	i1	j1
Operation (1)	Valid	-
Operation (2)	-	Invalid
Operation (3)	-	Invalid
Operation (4)	-	Invalid
Operation (5)	-	Invalid
Operation (6)	-	Invalid
Operation (7)	-	Invalid
Operation (8)	-	Invalid
Operation (9)	-	Invalid
Operation (10)	-	Invalid

:  
:

Operation (n-2)	-	Invalid
Operation (n-1)	-	Invalid
Operation (n)	-	Valid

When executing a second and following cutting in the G83 as shown above, the movement will change from rapid traverse to cutting feed several mm before the position machined last. When the hole bottom is reached, the axis will return according to the G98 or G99 mode. m will differ according to the parameter #8013 G83 return. Program so that q<sub>1</sub>>m. The operation stops at after the (1), (2) and (n) commands during single block operation.

# 13. Program Support Functions

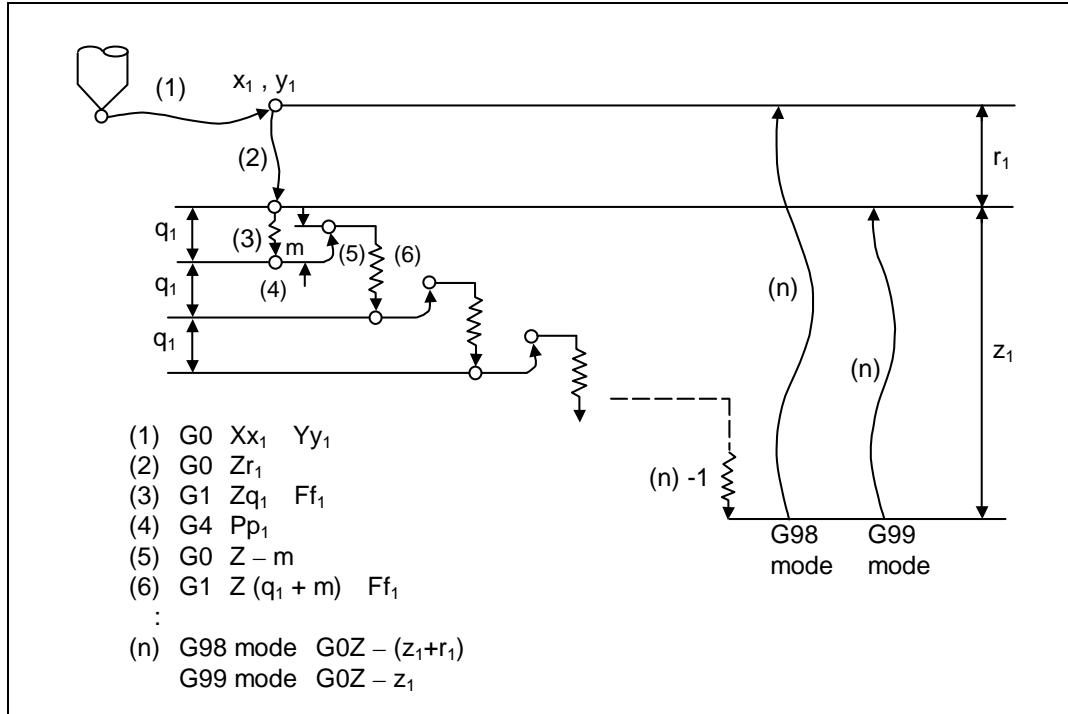
## 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)

(d) G83.1 (Step cycle)

Program

G83.1  $Xx_1$   $Yy_1$   $Zz_1$   $Qq_1$   $Rr_1$   $Ff_1$   $Pp_1$  , $li_1$  , $Jj_1$  ;

P : Dwell designation



Operation pattern	i1	j1
Operation (1)	Valid	-
Operation (2)	-	Invalid
Operation (3)	-	Invalid
Operation (4)	-	-
Operation (5)	-	Invalid
Operation (6)	-	Invalid

:  
:

Operation (n-1)	-	Invalid
Operation (n)	-	Valid

When executing a second and following cutting in the G83.1 as shown above, the movement will return several  $m$  mm with rapid traverse and then will change to cutting feed.

The return amount  $m$  will differ according to the parameter #8012 G73 return.

The operation stops at after the (1), (2) and (n) commands during single block operation.



## 13. Program Support Functions

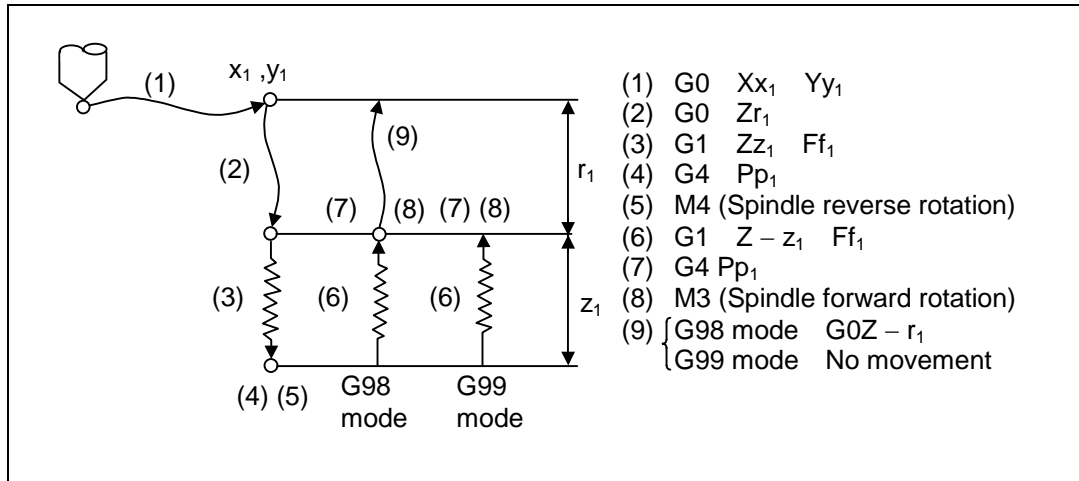
### 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)

(e) G84 (Tapping cycle)

Program

G84  $Xx_1 Yy_1 Zz_1 Rr_1 Ff_1 Pp_1 Rr_2 ,li_1 ,Jj_1 ;$

P : Dwell designation



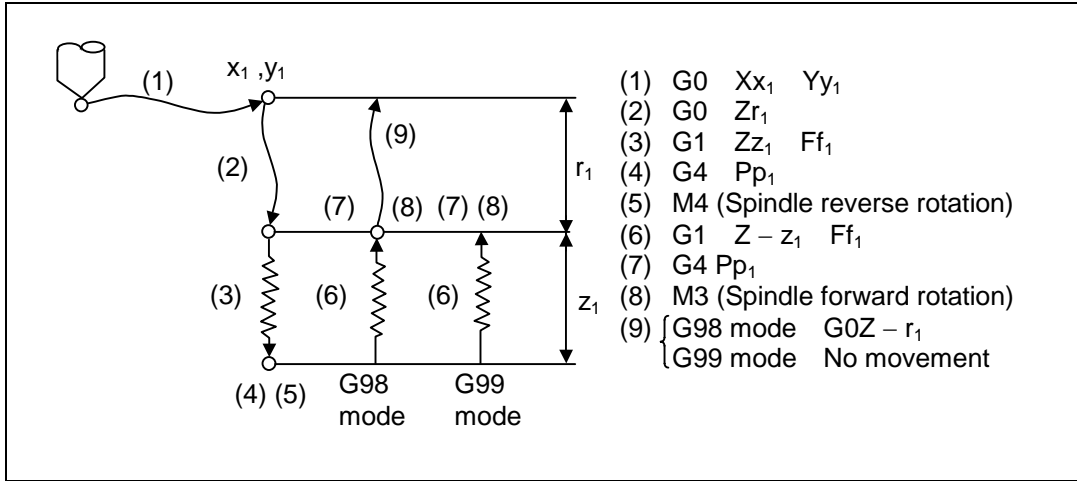
Operation pattern	i1	j1
Operation (1)	Valid	-
Operation (2)	-	Invalid
Operation (3)	-	Invalid
Operation (4)	-	-
Operation (5)	-	-
Operation (6)	-	Invalid
Operation (7)	-	-
Operation (8)	-	-
Operation (9)	-	Valid

When G84 is executed, the override will be canceled and the override will automatically be set to 100%. Dry run is valid when the control parameter "G00 DRY RUN" is ON and is valid for the positioning command. If the feed hold button is pressed during G84 execution, and the sequence is at (3) to (6), the movement will not stop immediately, and instead will stop after (6). During the rapid traverse in sequence (1), (2) and (9), the movement will stop immediately. The operation stops at after the (1), (2) and (9) commands during single block operation. During the G84 modal, the "Tapping" NC output signal will be output. During the G84 synchronous tapping modal, the M3, M4, M5 and S code will not be output.

# 13. Program Support Functions

## 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)

- (f) G84.2 (Synchronous tapping cycle)  
 Program  
 G84.2  $Xx_1 Yy_1 Zz_1 Rr_1 Ff_1 Pp_1 ,Ss_1 ,Ss_2 ,li_1 ,Jj_1 ;$   
 P : Dwell designation



Operation pattern	i1	j1
Operation (1)	Valid	-
Operation (2)	-	Invalid
Operation (3)	-	Invalid
Operation (4)	-	-
Operation (5)	-	-
Operation (6)	-	Invalid
Operation (7)	-	-
Operation (8)	-	-
Operation (9)	-	Valid

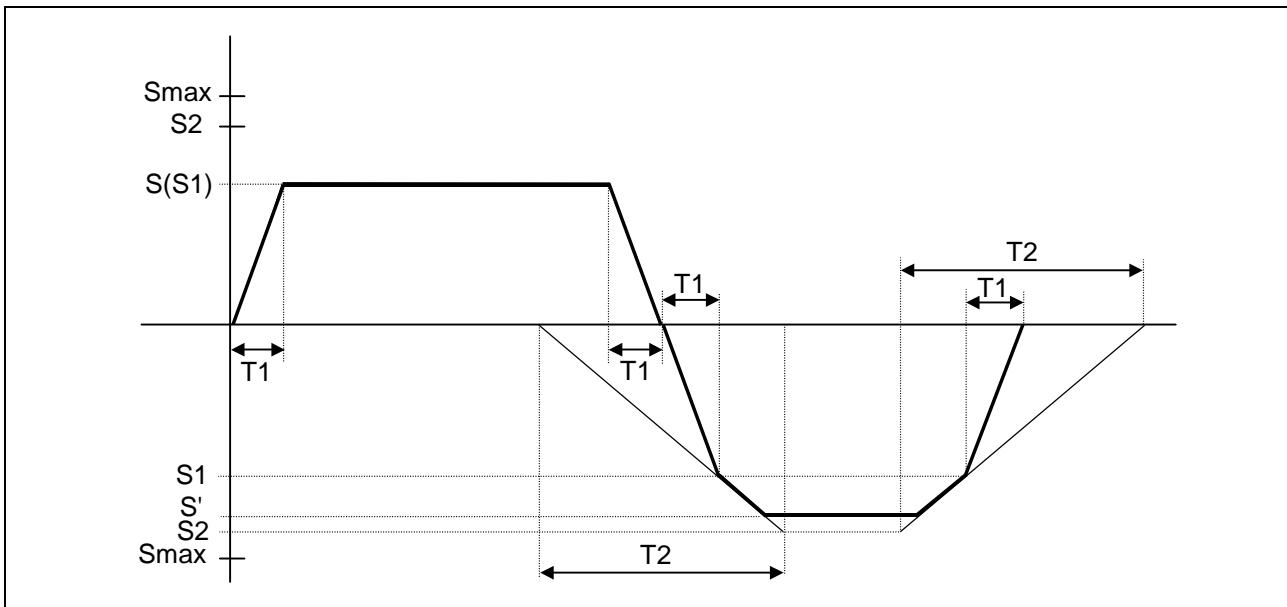
When G84.2 is executed, the override will be canceled and the override will automatically be set to 100%. Dry run is valid when the control parameter "G00 DRY RUN" is ON and is valid for the positioning command. If the feed hold button is pressed during G84.2 execution, and the sequence is at (3) to (6), the movement will not stop immediately, and instead will stop after (6). During the rapid traverse in sequence (1), (2) and (9), the movement will stop immediately. The operation stops at after the (1), (2) and (9) commands during single block operation. During the G84.2 modal, the "Tapping" NC output signal will be output. During the G84.2 synchronous tapping modal, the M3, M4, M5 and S code will not be output.

## 13. Program Support Functions

### 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)

This function allows spindle acceleration/deceleration pattern to be approached to the speed loop acceleration/deceleration pattern by dividing the spindle and hole drilling axis acceleration/deceleration pattern into up to three stages during synchronous tapping. The acceleration/deceleration pattern can be set up to three stages for each gear. When returning from the hole bottom, rapid return is possible depending on the spindle rotation speed during return. The spindle rotation speed during return is held as modal information.

When tap rotation speed < spindle rotation speed during return  $\leq$  synchronous tap changeover spindle rotation speed 2

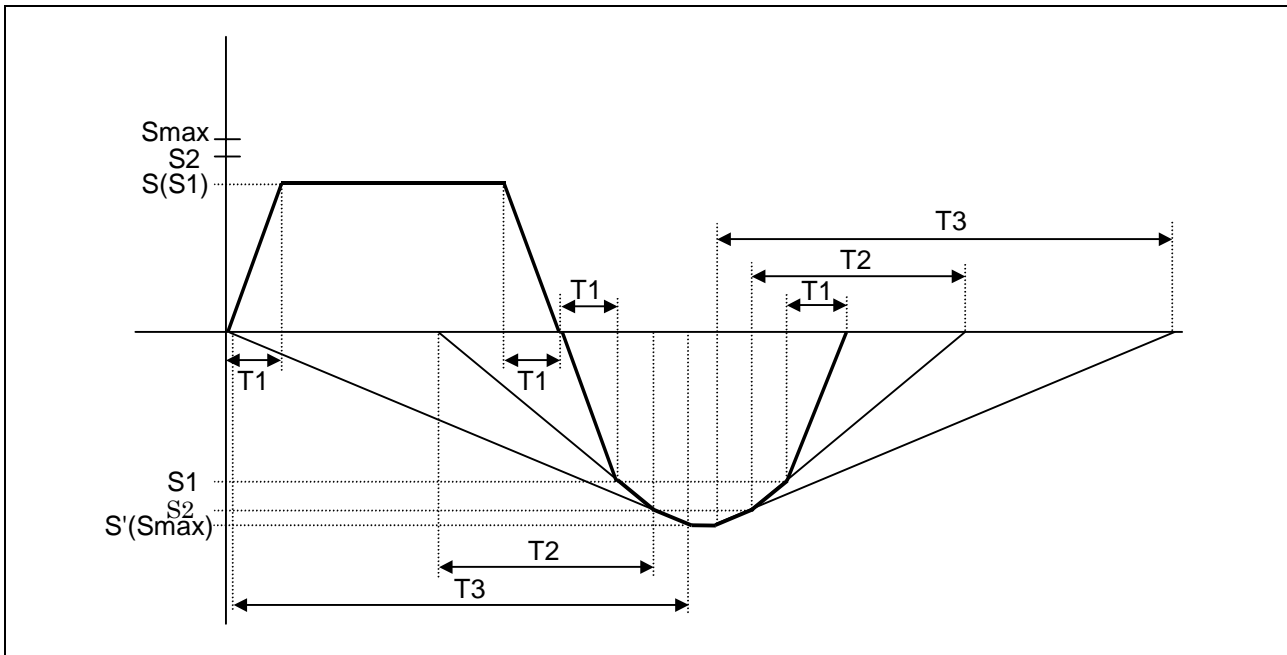


- S : Command spindle rotation speed
- S' : Spindle rotation speed during return
- S1 : Tap rotation speed (spindle base specification parameters #3013 to #3016)
- S2 : Synchronous tap changeover spindle rotation speed 2  
(spindle base specification parameters #3037 to #3040)
- Smax : Maximum rotation speed (spindle base specification parameters #3005 to #3008)
- T1 : Tap time constant (spindle base specification parameters #3017 to #3020)
- T2 : Synchronous tap changeover time constant 2  
(spindle base specification parameters #3041 to #3044)

# 13. Program Support Functions

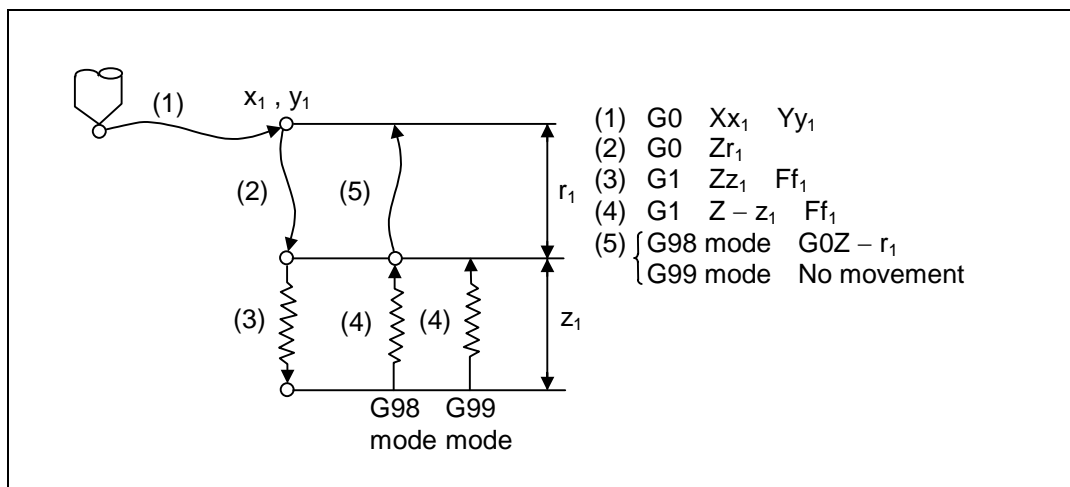
## 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)

When synchronous tap changeover spindle rotation speed 2 < spindle rotation speed during return



- S : Command spindle rotation speed
- S' : Spindle rotation speed during return
- S1 : Tap rotation speed (spindle base specification parameters #3013 to #3016)
- S2 : Synchronous tap changeover spindle rotation speed 2 (spindle base specification parameters #3037 to #3040)
- Smax : Maximum rotation speed (spindle base specification parameters #3005 to #3008)
- T1 : Tap time constant (spindle base specification parameters #3017 to #3020)
- T2 : Synchronous tap changeover time constant 2 (spindle base specification parameters #3041 to #3044)
- T3 : Synchronous tap changeover time constant 3 (spindle base specification parameters #3045 to #3048)

(g) G85 (Boring)  
 Program  
 G85 Xx<sub>1</sub> Yy<sub>1</sub> Zz<sub>1</sub> Rr<sub>1</sub> Ff<sub>1</sub> ,li<sub>1</sub> ,Jj<sub>1</sub> ;



# 13. Program Support Functions

## 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)

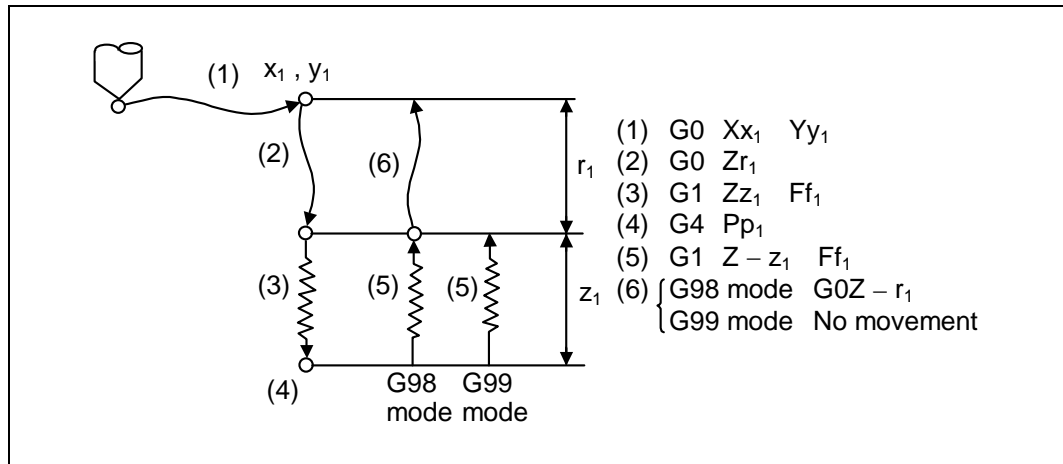
Operation pattern	i1	j1
Operation (1)	Valid	-
Operation (2)	-	Invalid
Operation (3)	-	Invalid
Operation (4)	-	Invalid
Operation (5)	-	Valid

The operation stops at after the (1), (2), (4) or (5) commands during single block operation.

(h) G89 (Boring)

Program

G89 Xx<sub>1</sub> Yy<sub>1</sub> Zz<sub>1</sub> Rr<sub>1</sub> Ff<sub>1</sub> Pp<sub>1</sub>, li<sub>1</sub>, Jj<sub>1</sub> ;



Operation pattern	i1	j1
Operation (1)	Valid	-
Operation (2)	-	Invalid
Operation (3)	-	Invalid
Operation (4)	-	-
Operation (5)	-	Invalid
Operation (6)	-	Valid

The operation stops at after the (1), (2), (5) or (6) commands during single block operation.

## 13. Program Support Functions

### 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)



#### Precautions for using fixed cycle

- (1) Before the fixed cycle is commanded, the spindle must be rotating in a specific direction with an M command (M3 or M4).
- (2) If there is a basic axis, additional axis or R data in the block during the fixed cycle mode, the hole drilling operation will be executed. If there is no data, the hole will not be drilled.  
Note that in the X axis data, if the data is a dwell (G04) time command, the hole will not be drilled.
- (3) Command the hole machining data (Q, P, I, J, K) in a block where hole drilling is executed. (Block containing a basic axis, additional axis or R data.)
- (4) The fixed cycle can be canceled by the G00 to G03 or G33 command besides the G80 command. If these are designated in the same block as the fixed cycle, the following will occur. (Where, 00 to 03 and 33 are m, and the fixed cycle code is n)

G <sub>m</sub>	G <sub>n</sub>	X	Y	Z	R	Q	P	L	F
Execute	Ignore		Execute		Ignore				Record
G <sub>m</sub>	G <sub>n</sub>	X	Y	Z	R	Q	P	L	F
Ignore		Execute			Ignore				Record

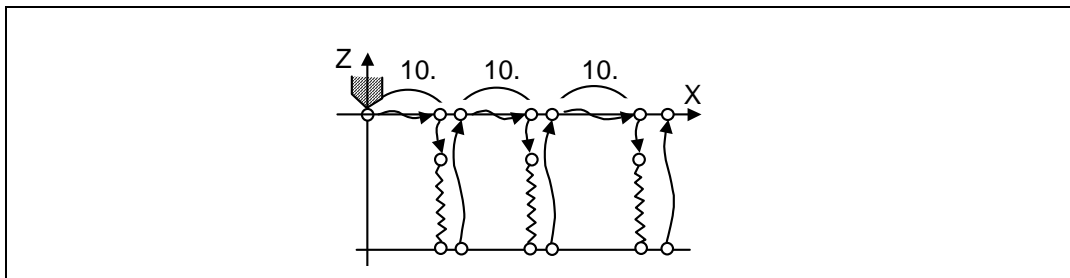
Note that for the G02 and G03 commands, R will be handled as the arc radius.

- (5) If an M function is commanded in the same block as the fixed cycle command, the M code and MF will be output during the initial positioning. The next operation will be moved to with FIN (finish signal).  
If there is a No. of times designated, the above control will be executed only on the first time.
- (6) If another control axis (ex., rotation axis, additional axis) is commanded in the same block as the fixed cycle control axis, the fixed cycle will be executed after the other control axis is moved first.
- (7) If the No. of repetitions L is not designated, L1 will be set. If L0 is designated in the same block as the fixed cycle G code command, the hole machining data will be recorded, but the hole machining will not be executed.

**(Example)** G83.1 X\_\_ Y\_\_ Z\_\_ R\_\_ Q\_\_ P\_\_ F\_\_ L0\_\_ ;  
Execute      Record only code having an address

- (8) When the fixed cycle is executed, only the modal command commanded in the fixed cycle program will be valid in the fixed cycle subprogram. The modal of the program that called out the fixed cycle will not be affected.
- (9) Other subprograms cannot be called from the fixed cycle subprogram.
- (10) Decimal points in the movement command will be ignored during the fixed cycle subprogram.
- (11) If the No. of repetitions L is 2 or more during the incremental value mode, the positioning will also be incremented each time.

**(Example)** G91 G81 X10. Z-50. R-20. F100. L3 ;



## 13. Program Support Functions

### 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)

- (12) If the spindle rotation speed value during return is smaller than the spindle rotation speed, the spindle rotation speed value is valid even during return.
- (13) If the 2nd and 3rd acceleration/deceleration stage inclinations following the spindle rotation speed and time constants set in the parameters are each steeper than the previous stage's inclination, the previous stage's inclination will be valid.
- (14) If the values set in the spindle base specification parameter "stap1-4" (tap rotation speed) and "taps21-24" (synchronous tap changeover spindle rotation speed 2) exceed the maximum rotation speed, the spindle rotation speed will be clamped at the maximum rotation speed.
- (15) If the spindle rotation speed during return is not 0, the tap return override value will be invalid.
- (16) In a block where the movement direction of any axis reverses as shown below, the servo system load will greatly increase so do not command the in-position width in the machining program.  
 G0 X100., I10.0 ;  
 X-200. ;
- (17) If the in-position width commanded with the programmable in-position width command is increased, the positioning time and linear interpolation time can be reduced. However, the position error amount of the previous block will also increase before the next block starts, so the actual machining could be obstructed.
- (18) The in-position width and the position error amount are compared at a set time, so the position error amount at the point to be judged as in-position will be smaller than the commanded in-position width.
- (19) If the in-position width commanded with the programmable in-position command is small, the commanded deceleration check or in-position check following the parameters may be carried out first.
- (20) Synchronous and asynchronous tap can be selected with the M function.

[Base specifications parameters]

#	Items	Details	Setting range
1513	stapM	M code for synchronous tap selection	0 to 99999999

The synchronous tap mode is selected with the miscellaneous function code set with this parameter.

The M function can be commanded in the same block before the tap command.

The synchronous and asynchronous tap will follow the combination shown below.

	Combination											
									No command			
Program command (G84/G84.2)	0	0	0	0	1	1	1	1				
#1229 (bit4)	0	0	1	1	0	0	1	1	0	0	1	1
M function code (M**)	x	○	x	○	x	○	x	○	x	○	x	○
Synchronous/asynchronous selection	A	A	A	A	S	S	S	S	A	S	S	S

0: G84    x : Does not command    A : Asynchronous tap  
 1: G84.2    ○: Commands    S : Synchronous tap

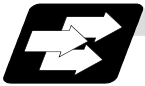
**(Note1)** Do not use M00, 01, 02, 30, 96, 97, 98 or 99.

**(Note2)** Depending on the model, selection with the M function may not be possible.

# 13. Program Support Functions

## 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)

### 13.6.1 Initial Point And R Point Level Return; G98, G99



#### Function and purpose

Whether to use R point or initial level for the return level in the final sequence of the fixed cycle can be selected.



#### Command format

G98 ; Initial level return  
G99 ; R point level return



#### Detailed description

The relation of the G98/G99 mode and No. of repetition designation is as shown below.

No. of hole drilling times	Program example	G98 (At power ON, at cancel with M02, M30, and reset button)	G99
Only one execution	G81 X100. Y100. Z-50. R25. F1000 ;	<p>Initial level return is executed.</p>	<p>R point level return is executed.</p>
Second and following executions	G81 X100. Y100. Z-50. R25. L5 F1000 ;	<p>Initial level return is executed for all times.</p>	



#### Example of program

##### (Example 1)

G82 Zz<sub>1</sub> Rr<sub>1</sub> Pp<sub>1</sub> Ff<sub>1</sub> L0 ;..... Record only the hole machining data (Do not execute)  
Xx<sub>1</sub> Yy<sub>1</sub> ;..... Execute hole drilling operation with G82 mode

The No. of fixed cycle repetitions is designated with L. If L1 is designated or L not designated, the fixed cycle will be executed once. The setting range is 1 to 9999.

If L0 is commanded, only the hole machining data will be recorded.

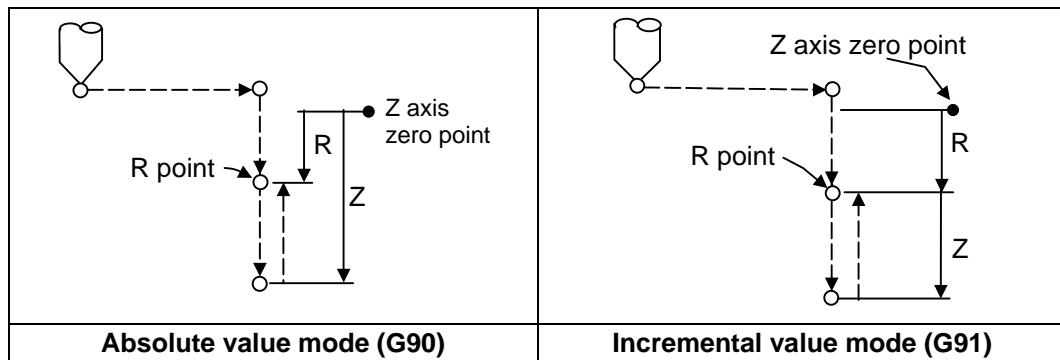
G8Δ Xx<sub>1</sub> Yy<sub>1</sub> Zz<sub>1</sub> Rr<sub>1</sub> Pp<sub>1</sub> Qq<sub>1</sub> Ff<sub>1</sub> Ll<sub>1</sub> ;



# 13. Program Support Functions

## 13.6 Hole Drilling Fixed Cycle (MELDAS Special Format)

The ideology of the data differs between the absolute value mode (G90) and incremental value mode (G91) as shown below.



Designate a command value with a symbol for X, Y and Z. R indicates the coordinate value from the zero point, so a symbol must always be added.

The hole machining data is held as shown below in the fixed cycle. The hole machining data is canceled when the G80 command or G commands (G00, G01, G02, G03, G2.1, G3.1, G33) in the 01 group are reached.

### (Example 2)

- N001 G81 Xx<sub>1</sub> Yy<sub>1</sub> Zz<sub>1</sub> Rr<sub>1</sub> Ff<sub>1</sub> ;
- N002 G81 ; ..... Only selection of fixed cycle sequence
- N003 Xx<sub>2</sub> Yy<sub>2</sub> ; ..... Change of positioning point, and execution of fixed cycle
- N004 M22 ; ..... Execution of only M22
- N005 G04 Xx<sub>3</sub> ; ..... Execution of only dwell
- N006 G92 Xx<sub>4</sub> Yy<sub>4</sub> ; ..... Execution of only coordinate system setting
- N007 G28 (G30) Z0 ; ..... Execution of only reference point (zero point) return
- N008 ; ..... No work
- N009 G99 Zz<sub>2</sub> Rr<sub>2</sub> Ff<sub>2</sub> L0 ; ..... Execution of only hole machining data recording
- N010 Xx<sub>5</sub> Yy<sub>5</sub> Ll<sub>5</sub> ; ..... Change of positioning point, and execution of R point return fixed cycle for 15 times
- N011 G98 Xx<sub>6</sub> Yy<sub>6</sub> Zz<sub>6</sub> Rr<sub>6</sub> ; .. Change of positioning point, and execution of fixed cycle
- N012 Ww<sub>1</sub> ; ..... Execute W axis according to 01 group modal before N001, and then execute fixed cycle

### 13.6.2 Setting of Workpiece Coordinates in Fixed Cycle Mode



#### Function and purpose

The designated axis moves with the workpiece coordinate system set for the axis. The Z axis is valid after the R point positioning after positioning or from Z axis movement.

**(Note)** When the workpiece coordinates are changed over for address Z and R, re-program even if the values are the same.

#### (Example)

- G54 Xx<sub>1</sub> Yy<sub>1</sub> Zz<sub>1</sub> ;
- G81 Xx<sub>2</sub> Yy<sub>2</sub> Zz<sub>2</sub> Rr<sub>2</sub> ;
- G55 Xx<sub>3</sub> Yy<sub>3</sub> Zz<sub>2</sub> Rr<sub>2</sub> ; ..... Re-command even if Z and R are the same as the previous value.
- Xx<sub>4</sub> Yy<sub>4</sub> ;
- Xx<sub>5</sub> Yy<sub>5</sub> ;

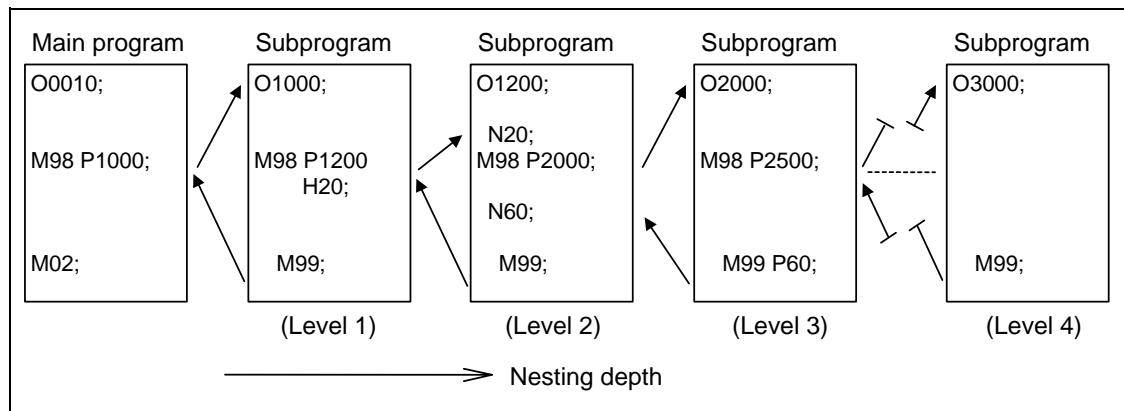
### 13.7 Subprogram control; M98, M99

#### 13.7.1 Calling subprogram with M98 and M99 commands



#### Function and purpose

Fixed sequences or repeatedly used parameters can be stored in the memory as subprograms which can then be called from the main program when required. M98 serves to call subprograms and M99 serves to return operation from the subprogram to the main program. Furthermore, it is possible to call other subprograms from particular subprograms and the nesting depth can include as many as 8 levels.



The table below shows the functions which can be executed by adding and combining the tape storing and editing functions, subprogram control functions and fixed cycle functions.

	Case 1	Case 2	Case 3	Case 4
1. Tape storing and editing	Yes	Yes	Yes	Yes
2. Subprogram control	No	Yes	Yes	No
3. Fixed cycles	No	No	Yes	Yes
<b>Function</b>				
1. Memory operation	○	○	○	○
2. Tape editing (main memory)	○	○	○	○
3. Subprogram call	×	○	○	×
4. Subprogram variable designation ( <b>Note 2</b> )	×	○	○	×
5. Subprogram nesting level call ( <b>Note 3</b> )	×	○	○	×
6. Fixed cycles	×	×	○	○
7. Subprogram editing for fixed cycle	×	×	○	○

**(Note 1)** "○" denotes a function which can be used and "×" a function which cannot be used.

**(Note 2)** Variables cannot be transferred with the M98 command but variable commands in subprograms can be used provided that the variable command option is available.

**(Note 3)** A maximum of 8 nesting levels form the nesting depth.



### Command format

#### Subprogram call

**M98 P\_ H\_ L\_ ;**  
M98 Subprogram call command  
M99 Subprogram return command  
P\_ Program number of subprogram to be called (own program if omitted)  
P can be omitted only during memory operation and MDI operation.  
(Max. 8 digits)  
H\_ Sequence number in subprogram to be called (head block if omitted)  
(Max. 5 digits)  
L\_ Number of subprogram repetitions  
(When omitted, this is interpreted at L1, and is not executed when L0.)  
(Between 1 and 9999 times according to 4-digit value.)  
For instance, M98 P1 L3 ; is equivalent to the following:  
M98 P1 ;  
M98 P1 ;  
M98 P1 ;

#### Return to main program from subprogram

**M98 P\_ L\_ ;**  
P Sequence number of return destination (returned to block following block to be called if omitted)  
\_ called if omitted  
L Number of times after repetition number has been changed



### Creating and entering subprograms

Subprograms have the same format as machining programs for normal memory operation except that the subprogram completion instruction M99 (P\_ L\_ ) ; is entered as an independent block at the last block.

<p>○△△△△△△△△ ;  ..... ;  ..... ;  :  :  :  ..... ;  M99 ;  % (EOR)</p>	<p>) Program number as subprogram number</p> <p>) Main body of subprogram</p> <p>) Subprogram return command Entry completion code</p>
--	--

- (1) The above program is entered by editing operations at the setting and display unit. For further details, refer to the section on program editing in the Operating Manual.
- (2) Only those subprograms numbers ranging from 1 to 99999999 designated by the optional specifications can be used. When there are no program numbers on the tape, they are entered as the setting number for "program input."
- (3) Up to 8 nesting levels can be used for calling subprograms from programs, and program error (P230) results if this number is exceeded.

- (4) No distinction between main programs and subprograms is made since they are entered in the sequence in which they were read. This means that main programs and subprograms should not be given the same numbers. (If they are, error "E11" appears during entry.)

```

;
O ○○○○○ ;
..... ;
:
:
M99 ;
%
;
    ) Subprogram A
    
```

```

O ΔΔΔΔ ;
..... ;
:
:
M99 ;
%
;
    ) Subprogram B
    
```

```

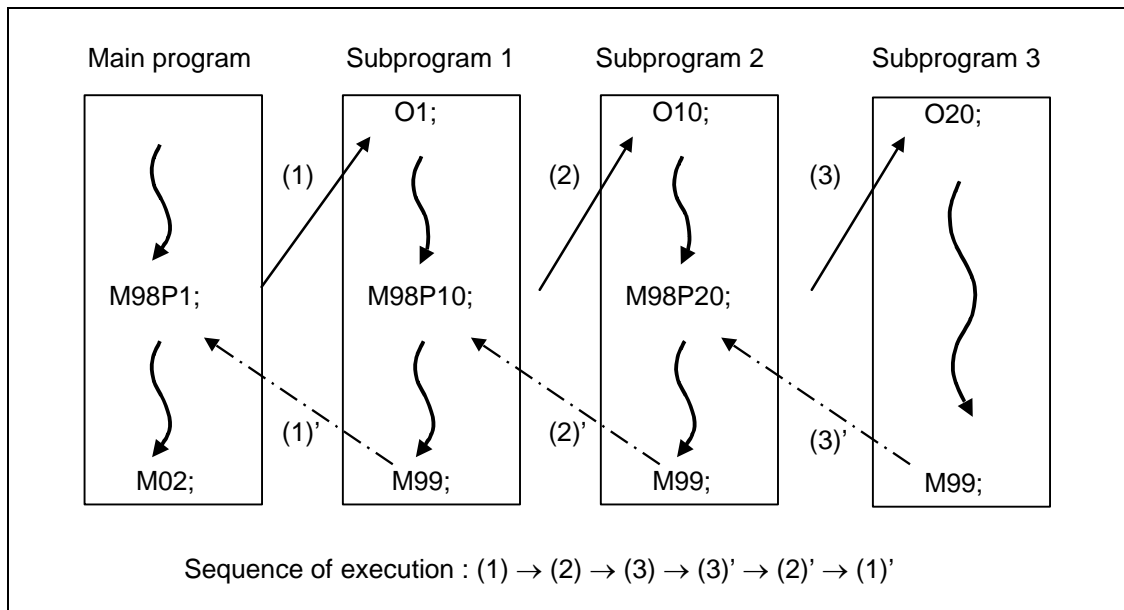
O **** ;
..... ;
:
:
M99 ;
%
;
    ) Subprogram C
    
```

- (5) Main programs can be entered during memory, tape or MDI operation but subprograms must be entered in the memory.
- (6) Besides the M98 command, subprogram nesting is subject to the following commands:
- G65 : Macro call
  - G66 : Modal call
  - G66.1 : Modal call
  - G code call
  - Miscellaneous function call
  - MDI interrupt
  - Automatic tool length measurement
  - Macro interrupt
  - Multi-step skip function
- (7) Subprogram nesting is not subject to the following commands which can be called even beyond the 8th nesting level.
- Fixed cycles
  - Pattern cycles
- (8) To repeatedly use the subprogram, it can be repeated l1 times by programming M98 Pp1 L1;.



### Example of program 1

When there are 3 subprogram calls (known as 3 nesting levels)

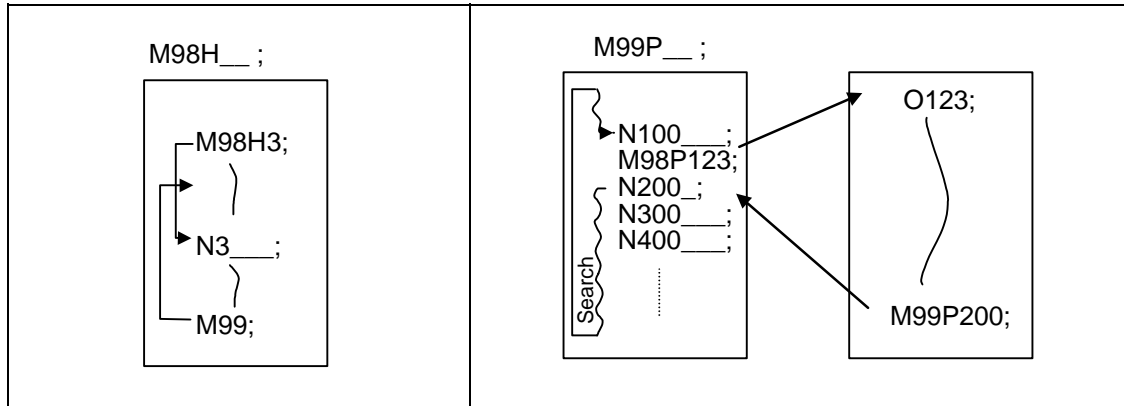


- (1) For nesting, the M98 and M99 commands should always be paired off on a 1:1 basis (1)' for (1), (2)' for (2), etc.
- (2) Modal information can be rewritten according to the execution sequence without distinction between main programs and subprograms. This means that after calling a subprogram, attention must be paid to the modal data status when programming.



### Example of program 2

The M98 H\_ ; M99 P\_ ; commands designate the sequence numbers in a program with a call instruction.



### Precautions

- (1) Program error (P232) results when the designated P (program number) is not located.
- (2) Single block stop does not occur with the M98 P\_ ; M99 ; block. If any address except O, N, P, L or H is used, single block stop can be executed. (With X100. M98 P100 ;, operation branches to O100 after X100. is executed.)
- (3) When M99 is commanded by the main program, operation returns to the head. (This is same for MDI.)
- (4) Branching from tape and BTR operation to the subprogram with M98 P\_ is possible, but the return destination sequence No. cannot be designated with M99 P\_ ;. (P\_ is ignored).
- (5) Bear in mind that the search operation will take time when the sequence number is designated by M99 P\_ ;.

### 13.8 Variable commands



#### Function and purpose

Programming can be endowed with flexibility and general-purpose capabilities by designating variables, instead of giving direct numerical values to particular addresses in a program, and by assigning the values or those variables as required when executing a program.



#### Command format

# $\Delta\Delta\Delta$  =  $\circ\circ\circ\circ\circ\circ\circ\circ$  or # $\Delta\Delta\Delta$  = [formula]



#### Detailed description

##### (1) Variable expressions

	<b>Example</b>
(a) #m ..... m = value consisting of 0-9	#100
(b) # [f] ..... f = one of the following in the formula	# [- #120]
Numerical value m	123
Variable	#543
Formula Operator Formula	#110 + #119
- (minus) formula	- # 120
[Formula]	[#119]
Function [formula]	SIN [#110]

**(Note 1)** The 4 standard operators are +, -, \* and / .

**(Note 2)** Functions cannot be used unless the user macro specifications are available.

**(Note 3)** Error (P241) results when a variable number is negative.

**(Note 4)** Examples of incorrect variable expressions are given below.

Incorrect	Correct
#6/2	→ # [6/2] (Note that expression such as "#6/2" is regarded as "[#6]/2".)
# - - 5	→ # [- [-5 ]]
# - [# 1]	→ # [- #1 ]

### (2) Types of variables

The following table gives the types of variables.

Type of variable	Number		Function
Common variables 200 sets	Common variables 1	Common variables 2	• Can be used in common throughout main, sub and macro programs.
	500 to 599	100 to 199	
Local variables	1 to 33		Can be used for local variables in macro programs.
System variables	1000 to		Application is fixed by system.
Fixed cycle variables	1 to 32		Local variables in fixed cycle programs.

**(Note 1)** All common variables are retained even when the power is turned OFF.

**(Note 2)** When the power is turned OFF or reset, the common variables can be set to <null> by setting the parameter (#1128 RstVC1, #1129 PwrVC1).

### (3) Variable quotations

Variables can be used for all addresses except O, N and / (slash).

- When the variable value is used directly:  
X#1 ..... Value of #1 is used as the X value.
- When the complement of the variable value is used:  
X-#2 ..... Value with the #2 sign changed is used as the X value.
- When defining variables:  
#3 = #5 ..... Variable #3 uses the equivalent value of variable #5.  
#1 = 1000 ..... Variable #1 uses the equivalent value 1000 (which is treated as 1000.).
- When defining the variable arithmetic formula:  
#1 = #3 + #2 - 100 ..... The value of the operation result of #3 + #2 - 100. is used as the #1 value.  
X [#1 + #3 + 1000] ..... The value of the operation result of #1 + #3 + 1000 is used as the X value.

**(Note 1)** A variable cannot be defined in the same block as an address. It must be defined in a separate block.

Incorrect	Correct
X #1 = #3 + 100 ;	→ #1 = #3 + 100 ; X#1 ;

**(Note 2)** Up to five sets of square parentheses [ ] may be used.

#543 = - [ [ [ [ [ #120]/2 + 15.] \*3 - #100]/#520 + #125 + #128]\*#130 + #132] ;

**(Note 3)** There are no restrictions on the number of characters and number of variables for variable definition.

**(Note 4)** The variable values should be within a range from 0 to ±99999999. If this range is exceeded, the operations may not be conducted properly.

**(Note 5)** The variable definitions are valid from the moment that the variables are actually defined.

#1 = 100 ; ..... #1 = 100  
#1 = 200 #2 = #1 + 200 ; ..... #1 = 200, #2 = 400  
#3 = #1 + 300 ; ..... #3 = 500

**(Note 6)** Variable quotations are always regarded as having a decimal point at the end.

When #100 = 10,  
then X#100 ; is treated as X10.



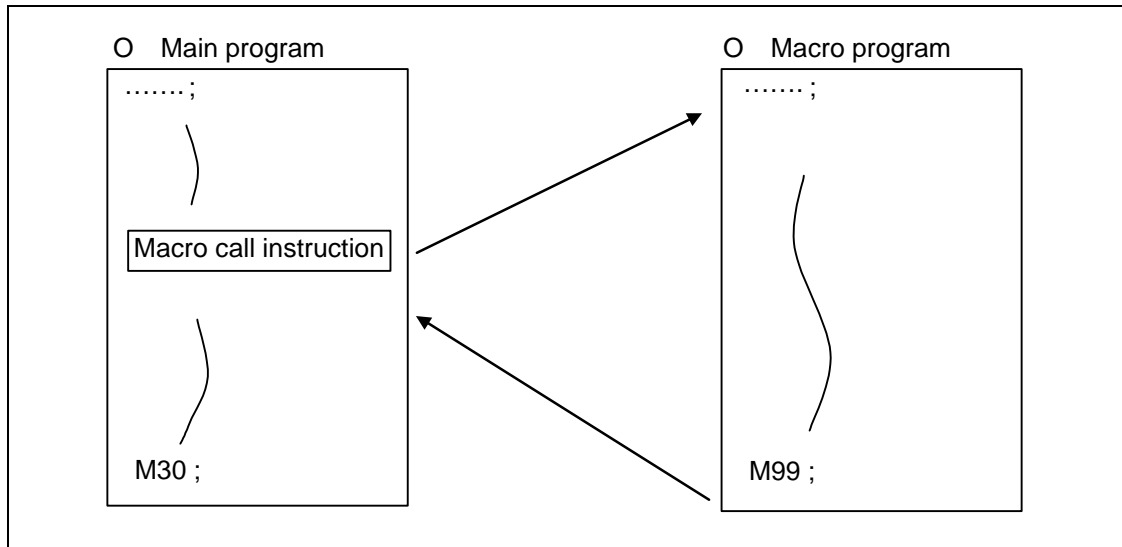
### 13.9 User macro

#### 13.9.1 User macro commands; G65, G66, G66.1, G67



#### Function and purpose

By combining the user macros with variable commands, it is possible to use macro program call, operation, data input/output with PLC, control, decision, branch and many other instructions for measurement and other such applications.



Macro programs use variables, operation instructions and control instructions to create subprograms which function to provide dedicated control. These dedicated control functions (macro programs) are called by the macro call instructions exactly when required from the main program.

G code	Function
G65	User macro Simple call
G66	User macro Modal call A (Movement command call)
G66.1	User macro Modal call B (Per-block call)
G67	User macro Modal call (G66, G66.1) cancel



#### Detailed description

- (1) When the G66 or G66.1 command is entered, the specified user macro program will be called after each block has been executed or after the movement command in the block with the movement commands has been executed until the G67 (cancel) command is entered.
- (2) The G66 (G66.1) and G67 commands must be paired in the same program.

### 13.9.2 Macro call instruction

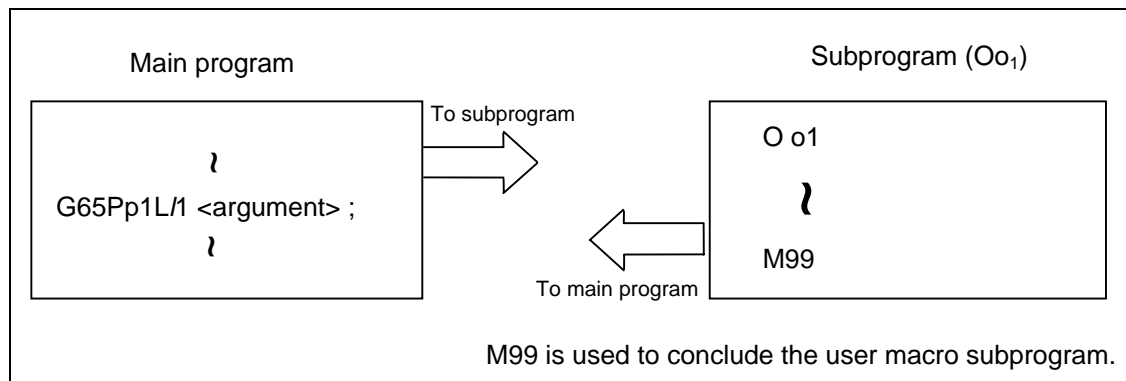


#### Function and purpose

Included among the macro call commands are the simple calls which apply only to the instructed block and also modal calls (types A and B) which apply to each block in the call modal.



#### Simple macro calls



#### Format

**M98 P\_\_ L\_\_ <argument> ;**

P\_                      Program No.  
L\_                      No. of repetitions

When the <argument> must be transferred as a local variable to a user macro subprogram, the actual value should be designated after the address.

Regardless of the address, a sign and decimal point can be used in the argument. There are 2 ways in which arguments are designated.

#### (1) Argument designation I

Format : A\_ B\_ C\_ ... X\_ Y\_ Z\_

##### Detailed description

- (a) Arguments can be designated using any address except G, L, N, O and P.
- (b) Except for I, J and K, there is no need for designation in alphabetical order.
- (c) I, J and K must be designated in alphabetical order.
  - I\_ J\_ K\_ ..... Correct
  - J\_ I\_ K\_ ..... Incorrect
- (d) Addresses which do not need to be designated can be omitted.
- (e) The following table shows the correspondence between the addresses which can be designated by argument designation I and the variable numbers in the user macro main body.

Address and variable number correspondence		Call instructions and usable address	
Argument designation I address	Variable in macro	G65, G66	G66.1
A	#1	○	○
B	#2	○	○
C	#3	○	○
D	#7	○	○
E	#8	○	○
F	#9	○	○
G	#10	×	×*
H	#11	○	○
I	#4	○	○
J	#5	○	○
K	#6	○	○
L	#12	×	×*
M	#13	○	○
N	#14	×	×*
O	#15	×	×
P	#16	×	×*
Q	#17	○	○
R	#18	○	○
S	#19	○	○
T	#20	○	○
U	#21	○	○
V	#22	○	○
W	#23	○	○
X	#24	○	○
Y	#25	○	○
Z	#26	○	○

○ : Can be used.

× : Cannot be used.

\* : Can be used while G66.1 command is modal.

### (2) Argument designation II

Format : A\_ B\_ C\_ I\_ J\_ K\_ I\_ J\_ K\_ ...

#### Detailed description

- (a) In addition to addresses A, B and C, up to 10 groups of arguments with I, J, K serving as 1 group can be designated.
- (b) When the same address is duplicated, designate the addresses in the specified order.
- (c) Addresses which do not need to be designated can be omitted.
- (d) The following table shows the correspondence between the addresses which can be designated by argument designation II and the variable numbers in the user macro main body.

Argument designation II address	Variable within macro
A	#1
B	#2
C	#3
I1	#4
J1	#5
K1	#6
I2	#7
J2	#8
K2	#9
I3	#10
J3	#11
K3	#12
I4	#13
J4	#14
K4	#15
I5	#16

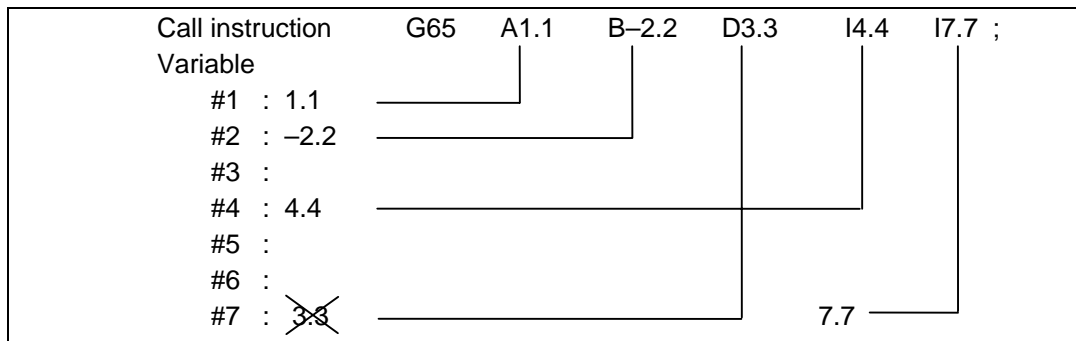
Argument designation II address	Variable within macro
J5	#17
K5	#18
I6	#19
J6	#20
K6	#21
I7	#22
J7	#23
K7	#24
I8	#25
J8	#26
K8	#27
I9	#28
J9	#29
K9	#30
I10	#31
J10	#32
K10	#33

**(Note 1)** The numbers 1 to 10 accompanying I, J and K denote the sequence of the commanded groups and they are not required for the actual instructions.

### (3) Using arguments designations I and II together

If addresses corresponding to the same variable are commanded when both types I and II are used to designate arguments, the latter address is valid.

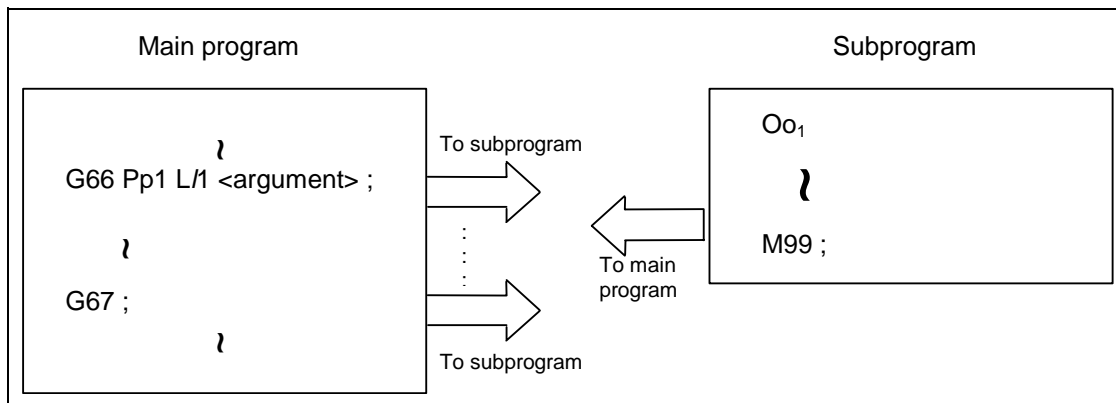
#### (Example 1)



In the above example the last I7.7 argument is valid when both arguments D3.3 and I7.7 are commanded for the #7 variable.



### Modal call A (movement command call)



When the block with a movement command is commanded between G66 and G67, the movement command is first executed and then the designated user macro subprogram is executed. The number of times the subprogram is executed is *l1* times with each call. The <argument> is the same as for a simple call.

#### Format

**G66 P\_ L\_ <argument> ;**

P\_                    Program No.

L\_                    No. of repetitions

#### Detailed description

- (1) When the G66 command is entered, the designated user macro subprogram will be called after the movement command in the block with the movement commands has been executed until the G67 (cancel) command is entered.
- (2) The G66 and G67 commands must be paired in the same program. A program error will result when G67 is issued without the G66 command.

**Modal call B (for each block)**

The designated user macro subprogram is unconditionally called at each command block commanded between G66.1 and G67, and is executed the designated number of times. The <argument> is the same as for a simple call.

**Format**

**G66.1 P\_\_ L\_\_ <argument> ;**

P\_                      Program No.

L\_                      No. of repetitions

**Detailed description**

- (1) In the G66.1 mode, everything except the O, N and G codes in the various command blocks which are read are handled as the argument without being executed. Note that the G code designated last, or the N code commanded after other than O and N is the argument.
- (2) The same applies as when G65 P\_ is assigned at the head of a block for all significant blocks in the G66.1 mode.

**(Example 1)**

```
N100 G01 G90 X100. Z200. F400 R1000 ;
in the G66.1 P1000 ; mode is the same as:
N100 G65 P1000 G01 G90 X100. Z200. F400 R1000 ;
```

**(Note 1)** The call is performed even in the G66.1 command block in the G66.1 mode and the correspondence between the argument address and the variable number is the same as for G65 (simple call).

- (3) The range of the G and N command values which can be used new as variables in the G66.1 mode is subject to the restrictions applying to value as normal NC command values.
- (4) Program No. O, sequence numbers N and modal G code are updated as modal information.



### G code macro call

User macro subprograms with prescribed program numbers can be called merely by issuing the G code command.

#### Format

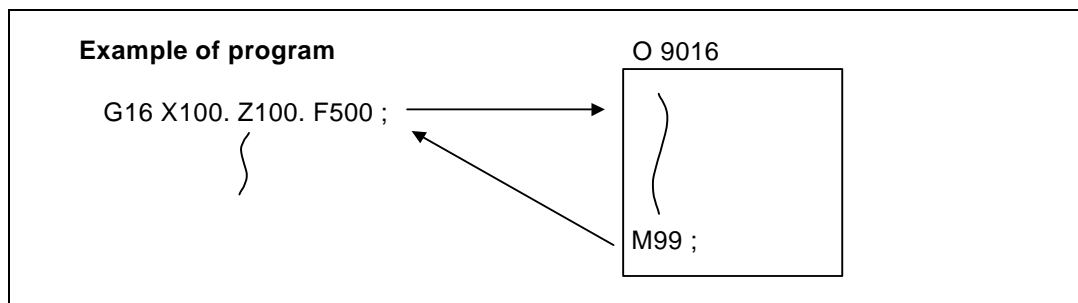
**G\*\*** <argument> ;  
 G\*\* G code for macro call

#### Detailed description

- (1) The above instruction functions in the same way as the instructions below, and parameters are set for each G code to determine the correspondence with the instructions.
  - a: M98 PΔΔΔΔ ;
  - b: G65 PΔΔΔΔΔ <argument> ;
  - c: G66 PΔΔΔΔΔ <argument> ;
  - d: G66.1 PΔΔΔΔΔ <argument> ;

When the parameters corresponding to c and d above are set, issue the cancel command (G67) either in the user macro or after the call code has been commanded so as to cancel the modal call.

- (2) The correspondence between the "\*\*\*" which conducts the macro call and the macro program No. PΔΔΔΔ to be called is set by parameter.
- (3) Up to 10 G codes from G00 to G255 can be used with this command. (G01 to G99 can also be used with parameter #1081Gmac\_P).
- (4) The commands cannot be issued during a user macro subprogram which has been called by a G code.





### Miscellaneous command macro call (for M, S, T, B code macro call)

The user macro subprograms of the specified program number can be called merely by issuing an M (or S, T, B) code. (Only entered codes apply for M but all S, T and B codes apply.)

#### Format

**M\*\* ; (or S\*\* ;, T\*\* ;, B\*\* ;)**

M\*\*                                    M code for macro call (or S, T, B code)

#### Detailed description

- (1) The above instruction functions in the same way as the instructions below, and parameters are set for each M code to determine the correspondence with the instructions. (Same for S, T and B codes)

- a: M98 PΔΔΔΔ ;
  - b: G65 PΔΔΔΔ M\*\* ;
  - c: G66 PΔΔΔΔ M\*\* ;
  - d: G66.1 PΔΔΔΔ M\*\* ;
- } M98, M\*\* are not output

When the parameters corresponding to c and d above are set, issue the cancel command (G67) either in the user macro or after the call code has been commanded so as to cancel the modal call.

- (2) The correspondence between the "M\*\*" which conducts the macro call and the macro program No. PΔΔΔΔ to be called is set by parameter. Up to 10 M codes from M00 to M95 can be entered.  
Any code except the codes basically required by the machine and the M0, M1, M2, M30 and M96 ~ M99 codes can be entered.
- (3) As with M98, the display appears on the CRT screen of the setting and display unit but the M codes and MF are not output.
- (4) Even if the miscellaneous command entered above is issued during a user macro subprogram called by the M code, macro call will not result and it will be handled as an ordinary miscellaneous command.
- (5) All S, T and B codes call the subprograms in the prescribed program numbers of the corresponding S, T and B functions.
- (6) Up to 10 M codes can be set, but set the parameter in the following manner when they will not all be used.

[MACRO FILE (1)]			
	<CODE>	<TYPE>	<PROGRAM-No.>
M[01]	20	0	8000
M[02]	21	0	8001
M[03]	9999	0	199999999
M[04]	9999	0	199999999
M[05]	9999	0	199999999
	:	:	:
	:	:	:
M[10]	9999	0	199999999

- For M20 commands, set to call O8000 in type 0 (M98 type)
- For M21 commands, set to call O8001 in type 0 (M98 type)
- Set parameters that will not be used as shown at the left.





### Differences between M98 and G65 commands

- (1) The argument can be designated for G65 but not for M98.
- (2) The sequence number can be designated for M98 but not for G65, G66 and G66.1.
- (3) M98 executes a subprogram after all the commands except M, P, H and L in the M98 block have been executed, but G65 branches to the subprogram without any further operation.
- (4) When any address except O, N, P, H or L is included in the M98 block, single block stop results. This is not the case with G65.
- (5) The level of the M98 local variables is fixed but it can be varied in accordance with the nesting depth for G65. (#1, for instance, has the same significance either before or after M98 but a different significance in each case with G65.)
- (6) The M98 nesting depth extends up to 8 levels in combination with G65, G66 and G66.1. The G65 nesting depth extends up to only 4 levels in combination with G66 and G66.1.



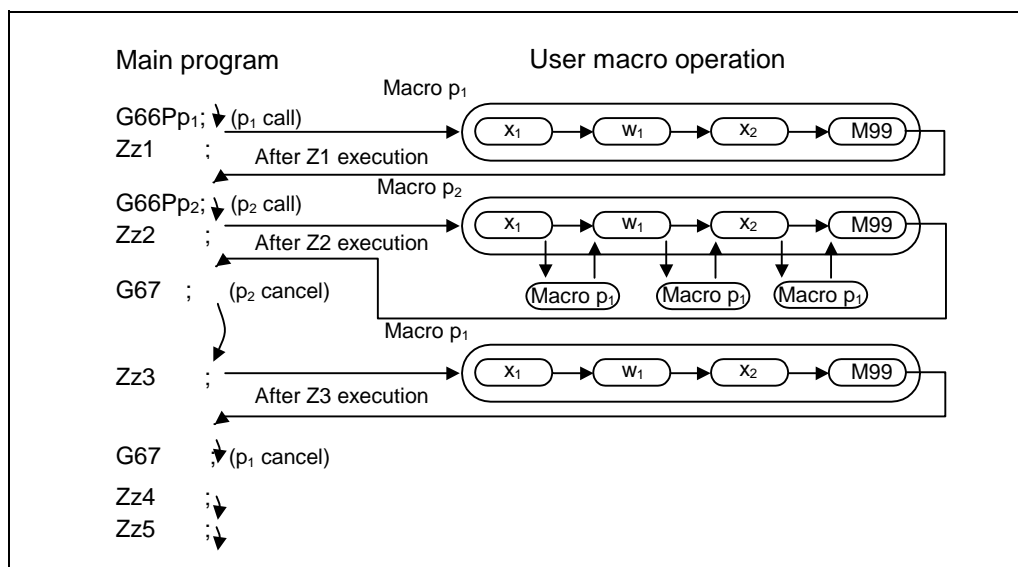
### Macro call command nesting depth

Up to 4 nesting levels are available for macro subprogram calls based on simple call or modal call. The argument with a macro call instruction is valid only on the called macro level. Since the nesting depth for macro calls extends up to 4 levels, the argument can be used as a local variable for the program with each respective macro call.

**(Note 1)** When a G65, G66, G66.1 G code macro call or miscellaneous command macro call is conducted, this is regarded as nesting level 1 and the level of the local variables is also incremented by one.

**(Note 2)** The designated user macro subprogram is called every time the movement command is executed with modal call A. However, when the G66 command has been duplicated, the next user macro subprogram is called every time an axis is moved even with movement commands in the macro. User macro subprograms are called in sequence from the subprogram commanded last.

### (Example 1)



### 13.9.3 ASCII code macro

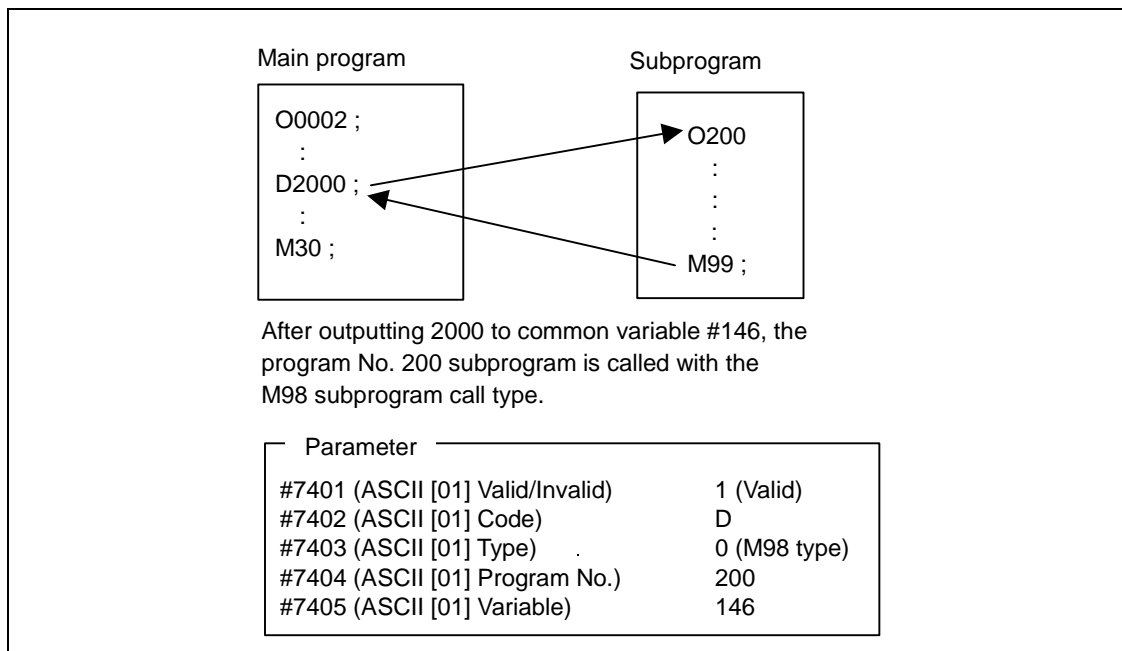


#### Function and purpose

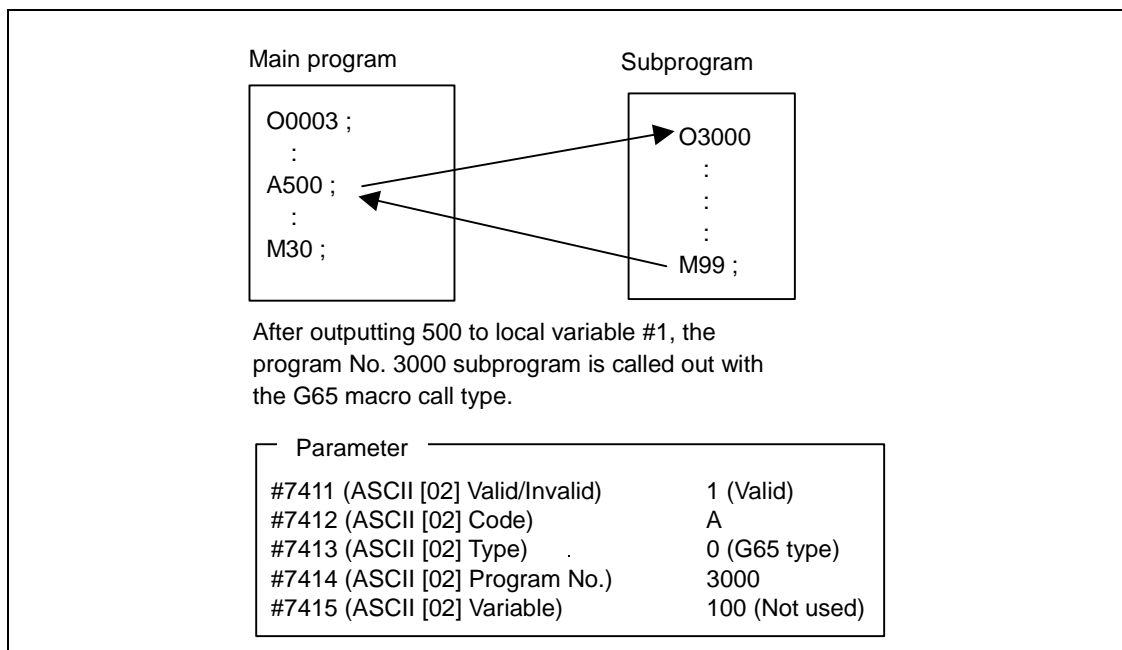
A macro program can be called out by setting the correspondence of a subprogram (macro program) preregistered with the parameters to codes, and then commanding the ASCII code in the machining program.

This function can be used in addition to the G, M, S, T and B miscellaneous command macro call function.

#### Execution example 1) M98 type



#### Execution example 2) G65 type





### Command format

□\*\*\*\*;

□

\*\*\*\*

**Designates the address and code**

ASCII code for calling out macro (one character)

Value or expression output to variable

(Setting range: ±999999.9999)



### Detailed description

- (1) The command above functions in the same way as that below. The correspondence of commands is set for each ASCII code with the parameters.

0: M98 PΔΔΔΔ ;

1: G65 PΔΔΔΔ <Argument> ;

2: G66 PΔΔΔΔ <Argument> ;

3: G66.1 PΔΔΔΔ <Argument> ;

When parameters are set for 2 and 3 above, the modal call will be canceled. Thus, command the cancel command (G67) after commanding the call code or the during the user macro.

- (2) The ASCII code for calling the macro with the ASCII codes and the program No. PΔΔΔΔ to be called are set with the parameters.

Up to two ASCII codes can be registered.

- (3) The code section is output to the variables, but the output destination differs according to the call type and address.

- (a) For M98 type

The code section is output to the common variable, and the variable No. is set with the parameters.

When corresponding to the first address (parameter #7401), the section is output to the common variable that indicates the first variable No. (parameter #7404).

- (b) For G65/G66/G66.1 type

The code section is output to the local variable. The variable No. differs according to the address, and corresponds to the following table.

Address	#	Address	#	Address	#	Address	#	Address	#	Address	#
A	1	F	9	K	6	P	16	U	21	Z	26
B	2	G	10	L	12	Q	17	V	22		
C	3	H	11	M	13	R	18	W	23		
D	7	I	4	N	14	S	19	X	24		
E	8	J	5	O	15	T	20	Y	25		

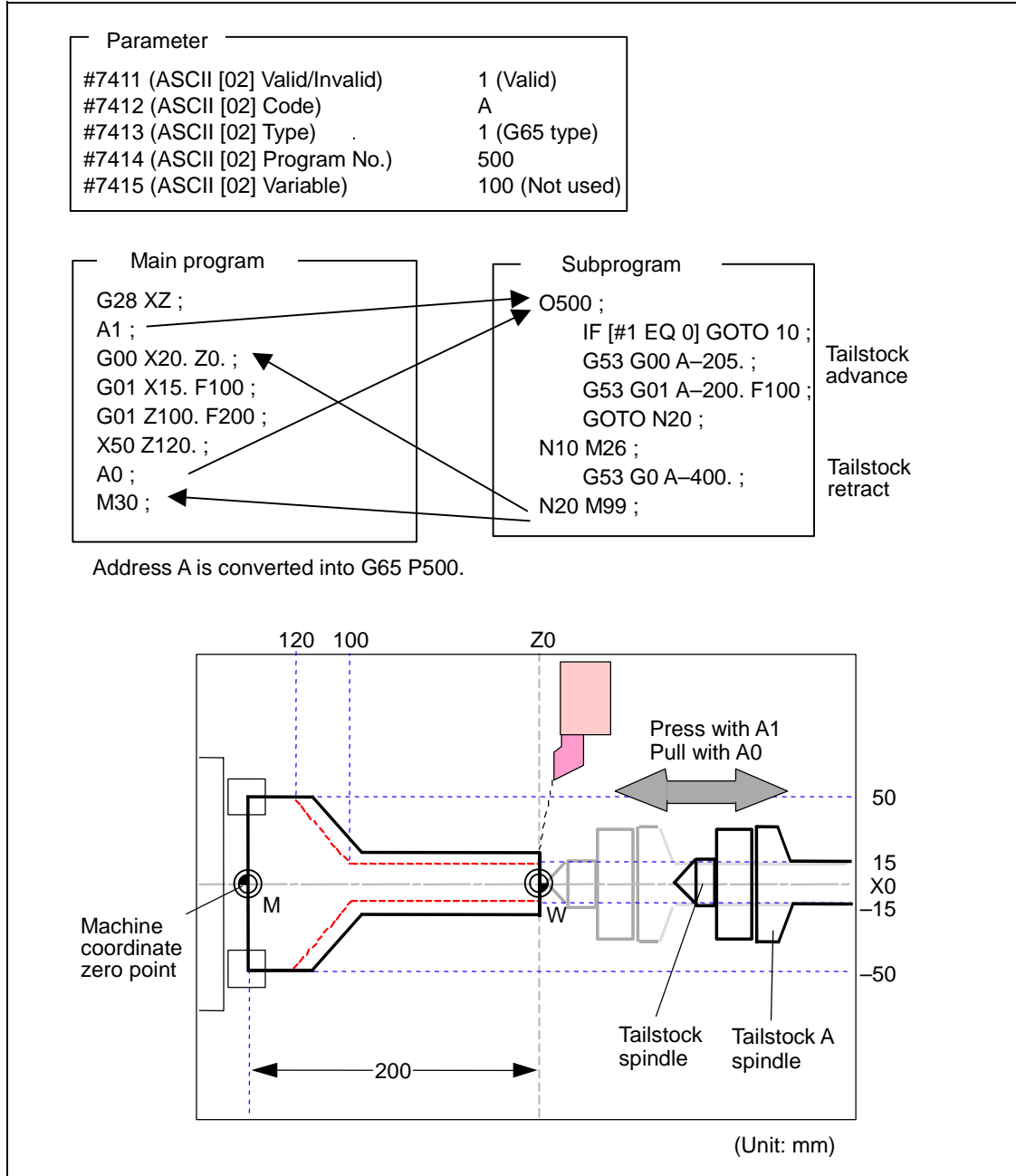
**(Note)** The following addresses can be used.

A, B, D, F, H, I, J, K, M, Q, R, S, T



### Example of program

A program that controls the tailstock with address A and machines a long thin workpiece is shown below.





### Restrictions

- (1) Calling a macro with an ASCII code from a program macro-called with an ASCII code

A macro cannot be called with an ASCII code from a program macro-called with an ASCII code.

The other patterns are shown below.

If it is determined that the macro cannot be called, the command will be handled as a normal command.

		Called side			
		ASCII	GMSTB macro	G65/66/66.1	M98
Calling side	ASCII	x	x	○	○
	GMSTB macro	x	x	○	○
	G65/66/66.1	○	○	○	○
	M98	○	○	○	○

- (2) Nest level of macro call command

The macro subprogram can be called in up to four levels using simple call (G65) and modal call (G66/G66.1).

The macro call command's argument is valid only in the called macro level.

Since the macro call nest level is four levels, the argument for each macro call can be used in the program as a local variable.

- (3) Nest level of subprogram call

Counting the main program as 0, up to eight levels of subprograms can be called (M98) from a subprogram.

The following commands are used for subprogram nesting.

- (a) M98
- (b) G65 G66 G66.1
- (c) G code call Miscellaneous function call (M/S/T/B)
- (d) MDI interrupt
- (e) Automatic tool length measurement
- (f) Multi-step skip function

The following commands can be commanded regardless of nesting.

- (g) Fixed cycle
- (h) Macro interrupt

## (4) Order of command priority

If "M" is designated for the ASCII code address, the codes basically necessary for that machine will be overlapped. In this case, commands will be identified with the following priority using code values.

- (a) M00 (program stop command), M01 (optional stop command)  
M02, M30 (end command)
- (b) When corresponding to miscellaneous code (M) call macro command
- (c) When corresponding to ASCII code macro command
- (d) Used as normal command

If "S", "T" and "B" are designated for the ASCII code address, commands will be identified with the following priority using code values.

- (a) When corresponding to miscellaneous code (S, T, B) call macro command
- (b) When corresponding to ASCII code macro command
- (c) Used as normal command

If the other addresses do not correspond to the ASCII code macro command, they will be identified as normal commands. If the command to be used overlaps with an ASCII code macro command, it must be commanded in the program macro-called with the ASCII code. Note that there are cases where the command will be unconditionally handled as a normal command, as explained in (5) below.

Note that there are cases when the command will be unconditionally handled as a normal command, as explained in (5) below.

## (5) Conditions for handling address set in ASCII code macro command as normal command

- (a) When there is a data setting command (G10) in the same block.
- (b) When ASCII code macro call is executed after the G code macro call command in the same block (also applies for M, S, T, B and ASCII)

**Example)** When address "D" (G65 type) is set in the ASCII code macro, and M50 is set in the macro call (G65 type).

M50 D200 ;    Execute M code macro with argument (200 set in #7)

- (c) When inputting parameters
- (d) When there is a comma (,) before the address. **Example)** ,D ,R, etc.
- (e) When commanded in fixed cycle
- (f) When commanded in macro subprogram called with G code macro call (Also applies when macro is called with M, S, T, B or ASCII)

### 13.9.4 Variables



#### Function and purpose

Both the variable specifications and user macro specifications are required for the variables which are used with the user macros.

The offset amounts of the local, common and system variables among the variables for this NC unit are retained even when the unit's power is turned OFF. (Common variables can also be set to < Blank > in parameter #1129PwrVC1)



#### Use of multiple variable

When the user macro specifications apply, variable numbers can be turned into variables (multiple use of variables) or replaced by <formula>. Only one of the arithmetical (+, -, x, ÷) operations can be conducted with <formula>.

#### (Example 1) Multiple use of variables

#1 = 10 #10 = 20 #20 = 30 ; #5 = # [# [#1]] ;	} # [# [#1]] = # [#10] from #1 = 10. # [#10] = #20 from #10 = 20. Therefore, #5 = #20 or #5 = 30.
--	---

#1 = 10 #10 = 20 #20 = 30 #5 = 1000 ; # [# [#1]] = #5 ;	} # [# [#1]] = # [#10] from #1 = 10. # [#10] = #20 from #10 = 20. Therefore, #20 = #5 or #20 = 1000.
--	--

#### (Example 2) Replacing variable numbers with <formula>

#10 = 5 ;	
# [#10 + 1] = 1000 ;	In which case, #6 = 1000.
# [#10 - 1] = -1000 ;	In which case, #4 = -1000.
# [#10*3] = 100 ;	In which case, #15 = 100.
# [#10/2] = -100 ;	In which case, #3 = -100. (fraction rounded up)



### Undefined variables

Variables applying with the user macro specifications such as variables which have not been used even once after the power was turned ON or local variables not quoted by the G65, G66 or G66.1 commands can be used as <Blank>. Also, variables can forcibly be set to <Blank>. Variable #0 is always used as the <Blank> variable and nothing can be defined in the left-side member.

#### (1) Arithmetic expressions

```
#1 = #0 ; ..... #1 = <Blank>
#2 = #0 + 1 ; ..... #2 = 1
#3 = 1 + #0 ; ..... #3 = 1
#4 = #0*10 ; ..... #4 = 0
#5 = #0 + #0 ; ..... #5 = 0
```

It should be borne in mind that <Blank> in an arithmetic expression is handled in the same way as 0.

```
<Blank> + <Blank> = 0
<Blank> + <Constant> = Constant
<Constant> + <Blank> = Constant
```

#### (2) Variable quotations

When undefined variable only are quoted, they are ignored up to the address.

When #1 = <Blank>

G0 X#1 Z1000 ; ..... Equivalent to G0 Z1000 ;

G0 X#1 + 10 Z1000 ; ..... Equivalent to G0 X10 Z1000 ;

#### (3) Conditional expressions

<Blank> differs from "0" only for EQ and NE. (#0 is <Blank>.)

When #101 = <Blank>	When #101 = 0
#101 EQ #0 <Blank> = <Blank> established	#101 EQ #0 0 = <Blank> not established
#101 NE 0 <Blank> ≠ 0 established	#101 NE 0 0 ≠ 0 not established
#101 GE #0 <Blank> ≥ 0 established	#101 GE #0 0 ≥ <Blank> established
#101 GT 0 <Blank> > 0 not established	#101 GT 0 0 > 0 not established
#101 LE #0 <Blank> ≤ <Blank> established	#101 LE #0 0 ≤ <Blank> established
#101 GT0 <Blank> < 0 not established	#101 LT0 0 < 0 not established

**(Note 1)** Only compare EQ and NE by integers. Use GE, GT, LE and LT for comparison when there are numbers below the decimal point.



### 13.9.5 Types of variables



#### Common variables

These are the variables used commonly from a random position. The number of the common variables sets depends on the specifications. Refer to the explanation about Variables for details.



#### Local variables (#1 to #33)

These can be defined as an <argument> when a macro subprogram is called or used locally within main programs and subprograms. They can be duplicated regardless of the relationship existing between macros (up to 4 levels).

**G65 Pp<sub>1</sub> L<sub>1</sub> <argument> ;**

p<sub>1</sub>                      Program number  
l<sub>1</sub>                        Number of repetitions

The <argument> is assumed to be Aa1 Bb1 Cc1..... Zz1.

The following table shows the correspondences between the addresses designated by <argument> and the local variable numbers used in the user macro main body.

#### [Argument designation I]

Call command		Argument address	Local variable number
G65, G66	G66.1		
○	○	A	#1
○	○	B	#2
○	○	C	#3
○	○	D	#7
○	○	E	#8
○	○	F	#9
x	x*	G	#10
○	○	H	#11
○	○	I	#4
○	○	J	#5
○	○	K	#6
x	x*	L	#12
○	○	M	#13
x	x*	N	#14
x	x*	O	#15
x	x*	P	#16

Call command		Argument address	Local variable number
G65, G66	G66.1		
○	○	Q	#17
○	○	R	#18
○	○	S	#19
○	○	T	#20
○	○	U	#21
○	○	V	#22
○	○	W	#23
○	○	X	#24
○	○	Y	#25
○	○	Z	#26
		-	#27
		-	#28
		-	#29
		-	#30
		-	#31
		-	#32
		-	#33

"x" in the above table denotes an argument address which cannot be used. However, provided that the G66.1 mode has been established, an argument address denoted by the asterisk can be added for use.

The hyphen (-) mark indicates that there is no corresponding address.

[Argument designation II]

Argument designation II address	Variable in macro
A	# 1
B	# 2
C	# 3
I1	# 4
J1	# 5
K1	# 6
I2	# 7
J2	# 8
K2	# 9
I3	#10
J3	#11
K3	#12
I4	#13
J4	#14
K4	#15
I5	#16

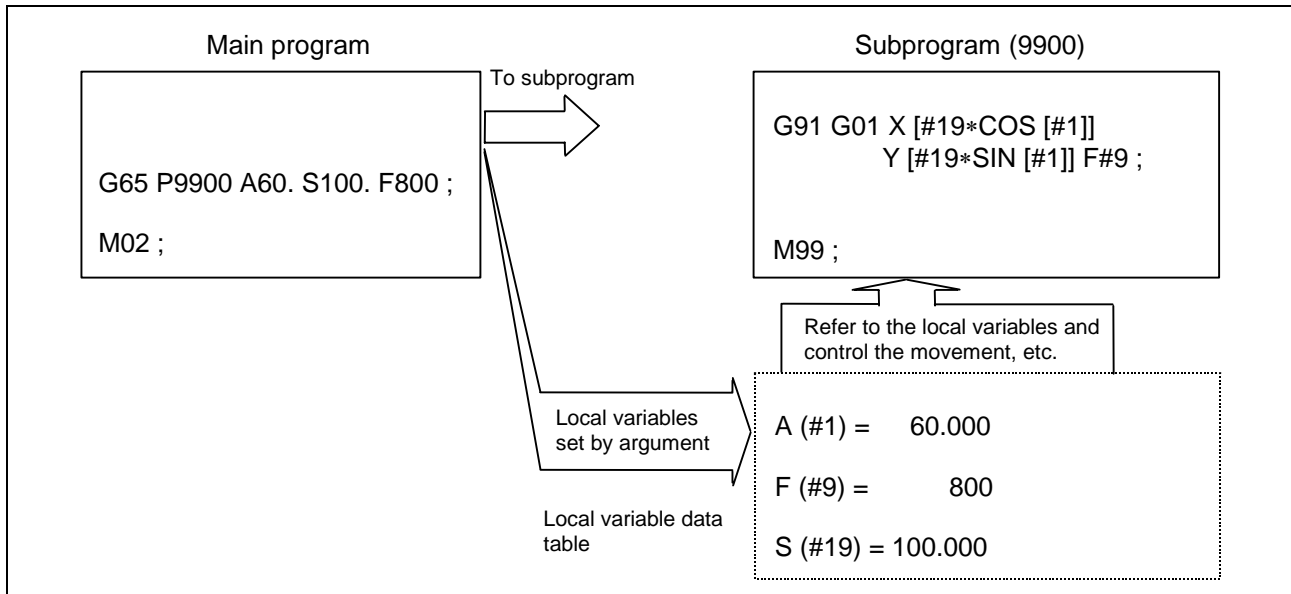
Argument designation II address	Variable in macro
J5	#17
K5	#18
I6	#19
J6	#20
K6	#21
I7	#22
J7	#23
K7	#24
I8	#25
J8	#26
K8	#27
I9	#28
J9	#29
K9	#30
I10	#31
J10	#32
K10	#33

**(Note 1)** The numbers 1 to 10 added to I, J and K indicate the sequence of the commanded sets, and are not required in the actual command.

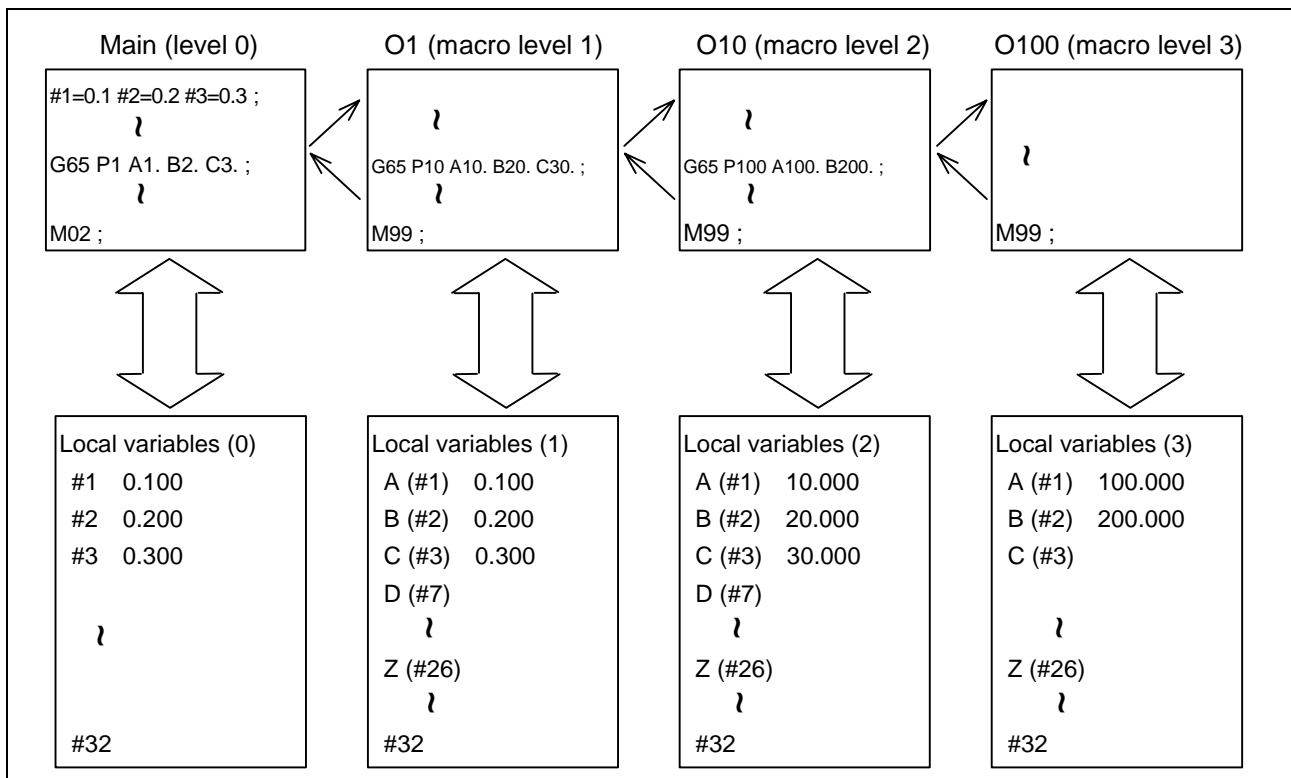
# 13. Program Support Functions

## 13.9 User macro

- (1) Local variables in subprograms can be defined by means of the <argument> designation during macro call. (Local variables can be used freely in those subprograms.)



- (2) Local variables can be used independently on each of the macro calls levels (4 levels). Local variables are also provided independently for the main program (macro level 0). Arguments cannot be used for the level 0 local variables.



The statuses of the local variables appear on the setting and display unit. For details, refer to the Operating Manual.



**Macro interface inputs (#1000 ~ #1035, #1200 ~ #1295) : PLC -> NC**

The status of the interface input signals can be ascertained by reading out the values of variable numbers #1000 to #1035, #1200 to #1295. Available value which has been read out can be only one of 2 values: 1 or 0 (1: contact closed, 0: contact open). All the input signals from #1000 to #1031 can be read at once by reading out the value of variable number 1032. In the same manner, by reading the values for variable numbers #1033 to #1035, the input signals for #1200 to #1231, #1232 to #1263 and #1264 to #1295 can be read.

Variable numbers #1000 to #1035, #1200 to #1295 are for readout only, and nothing can be placed in the left side member of their operation formula.

System variable	No. of points	Interface input signal	System variable	No. of points	Interface input signal
#1000	1	Register R72 bit 0	#1016	1	Register R73 bit 0
#1001	1	Register R72 bit 1	#1017	1	Register R73 bit 1
#1002	1	Register R72 bit 2	#1018	1	Register R73 bit 2
#1003	1	Register R72 bit 3	#1019	1	Register R73 bit 3
#1004	1	Register R72 bit 4	#1020	1	Register R73 bit 4
#1005	1	Register R72 bit 5	#1021	1	Register R73 bit 5
#1006	1	Register R72 bit 6	#1022	1	Register R73 bit 6
#1007	1	Register R72 bit 7	#1023	1	Register R73 bit 7
#1008	1	Register R72 bit 8	#1024	1	Register R73 bit 8
#1009	1	Register R72 bit 9	#1025	1	Register R73 bit 9
#1010	1	Register R72 bit 10	#1026	1	Register R73 bit 10
#1011	1	Register R72 bit 11	#1027	1	Register R73 bit 11
#1012	1	Register R72 bit 12	#1028	1	Register R73 bit 12
#1013	1	Register R72 bit 13	#1029	1	Register R73 bit 13
#1014	1	Register R72 bit 14	#1030	1	Register R73 bit 14
#1015	1	Register R72 bit 15	#1031	1	Register R73 bit 15

System variable	No. of points	Interface input signal
#1032	32	Register R72, R73
#1033	32	Register R74, R75
#1034	32	Register R76, R77
#1035	32	Register R78, R79

System variable	No. of points	Interface input signal	System variable	No. of points	Interface input signal
#1200	1	Register R74 bit 0	#1216	1	Register R75 bit 0
#1201	1	Register R74 bit 1	#1217	1	Register R75 bit 1
#1202	1	Register R74 bit 2	#1218	1	Register R75 bit 2
#1203	1	Register R74 bit 3	#1219	1	Register R75 bit 3
#1204	1	Register R74 bit 4	#1220	1	Register R75 bit 4
#1205	1	Register R74 bit 5	#1221	1	Register R75 bit 5
#1206	1	Register R74 bit 6	#1222	1	Register R75 bit 6
#1207	1	Register R74 bit 7	#1223	1	Register R75 bit 7
#1208	1	Register R74 bit 8	#1224	1	Register R75 bit 8
#1209	1	Register R74 bit 9	#1225	1	Register R75 bit 9
#1210	1	Register R74 bit 10	#1226	1	Register R75 bit 10
#1211	1	Register R74 bit 11	#1227	1	Register R75 bit 11
#1212	1	Register R74 bit 12	#1228	1	Register R75 bit 12
#1213	1	Register R74 bit 13	#1229	1	Register R75 bit 13
#1214	1	Register R74 bit 14	#1230	1	Register R75 bit 14
#1215	1	Register R74 bit 15	#1231	1	Register R75 bit 15

System variable	No. of points	Interface input signal	System variable	No. of points	Interface input signal
#1232	1	Register R76 bit 0	#1248	1	Register R77 bit 0
#1233	1	Register R76 bit 1	#1249	1	Register R77 bit 1
#1234	1	Register R76 bit 2	#1250	1	Register R77 bit 2
#1235	1	Register R76 bit 3	#1251	1	Register R77 bit 3
#1236	1	Register R76 bit 4	#1252	1	Register R77 bit 4
#1237	1	Register R76 bit 5	#1253	1	Register R77 bit 5
#1238	1	Register R76 bit 6	#1254	1	Register R77 bit 6
#1239	1	Register R76 bit 7	#1255	1	Register R77 bit 7
#1240	1	Register R76 bit 8	#1256	1	Register R77 bit 8
#1241	1	Register R76 bit 9	#1257	1	Register R77 bit 9
#1242	1	Register R76 bit 10	#1258	1	Register R77 bit 10
#1243	1	Register R76 bit 11	#1259	1	Register R77 bit 11
#1244	1	Register R76 bit 12	#1260	1	Register R77 bit 12
#1245	1	Register R76 bit 13	#1261	1	Register R77 bit 13
#1246	1	Register R76 bit 14	#1262	1	Register R77 bit 14
#1247	1	Register R76 bit 15	#1263	1	Register R77 bit 15

System variable	No. of points	Interface input signal	System variable	No. of points	Interface input signal
#1264	1	Register R78 bit 0	#1280	1	Register R79 bit 0
#1265	1	Register R78 bit 1	#1281	1	Register R79 bit 1
#1266	1	Register R78 bit 2	#1282	1	Register R79 bit 2
#1267	1	Register R78 bit 3	#1283	1	Register R79 bit 3
#1268	1	Register R78 bit 4	#1284	1	Register R79 bit 4
#1269	1	Register R78 bit 5	#1285	1	Register R79 bit 5
#1270	1	Register R78 bit 6	#1286	1	Register R79 bit 6
#1271	1	Register R78 bit 7	#1287	1	Register R79 bit 7
#1272	1	Register R78 bit 8	#1288	1	Register R79 bit 8
#1273	1	Register R78 bit 9	#1289	1	Register R79 bit 9
#1274	1	Register R78 bit 10	#1290	1	Register R79 bit 10
#1275	1	Register R78 bit 11	#1291	1	Register R79 bit 11
#1276	1	Register R78 bit 12	#1292	1	Register R79 bit 12
#1277	1	Register R78 bit 13	#1293	1	Register R79 bit 13
#1278	1	Register R78 bit 14	#1294	1	Register R79 bit 14
#1279	1	Register R78 bit 15	#1295	1	Register R79 bit 15



**Macro interface outputs (#1100 ~ #1135, #1300 ~ #1395) : NC -> PLC**

The interface output signals can be sent by substituting values in variable numbers #1100 to #1135, #1300 to #1395. An output signal can be only 0 or 1.

All the output signals from #1100 to #1131 can be sent at once by substituting a value in variable number 1132. In the same manner, by reading the values for variable numbers 1133 to 1135, the output signals for #1300 to #1331, #1332 to #1363 and #1364 to #1395 can be read.

The status of the writing and output signals can be read in order to offset the #1100 to #1135, #1300 to #1395 output signals. Output here refers to the output from the NC.

System variable	No. of points	Interface input signal	System variable	No. of points	Interface input signal
#1100	1	Register R172 bit 0	#1116	1	Register R173 bit 0
#1101	1	Register R172 bit 1	#1117	1	Register R173 bit 1
#1102	1	Register R172 bit 2	#1118	1	Register R173 bit 2
#1103	1	Register R172 bit 3	#1119	1	Register R173 bit 3
#1104	1	Register R172 bit 4	#1120	1	Register R173 bit 4
#1105	1	Register R172 bit 5	#1121	1	Register R173 bit 5
#1106	1	Register R172 bit 6	#1122	1	Register R173 bit 6
#1107	1	Register R172 bit 7	#1123	1	Register R173 bit 7
#1108	1	Register R172 bit 8	#1124	1	Register R173 bit 8
#1109	1	Register R172 bit 9	#1125	1	Register R173 bit 9
#1110	1	Register R172 bit 10	#1126	1	Register R173 bit 10
#1111	1	Register R172 bit 11	#1127	1	Register R173 bit 11
#1112	1	Register R172 bit 12	#1128	1	Register R173 bit 12
#1113	1	Register R172 bit 13	#1129	1	Register R173 bit 13
#1114	1	Register R172 bit 14	#1130	1	Register R173 bit 14
#1115	1	Register R172 bit 15	#1131	1	Register R173 bit 15

System variable	No. of points	Interface input signal
#1132	32	Register R172, R173
#1133	32	Register R174, R175
#1134	32	Register R176, R177
#1135	32	Register R178, R179

System variable	No. of points	Interface input signal	System variable	No. of points	Interface input signal
#1300	1	Register R174 bit 0	#1316	1	Register R175 bit 0
#1301	1	Register R174 bit 1	#1317	1	Register R175 bit 1
#1302	1	Register R174 bit 2	#1318	1	Register R175 bit 2
#1303	1	Register R174 bit 3	#1319	1	Register R175 bit 3
#1304	1	Register R174 bit 4	#1320	1	Register R175 bit 4
#1305	1	Register R174 bit 5	#1321	1	Register R175 bit 5
#1306	1	Register R174 bit 6	#1322	1	Register R175 bit 6
#1307	1	Register R174 bit 7	#1323	1	Register R175 bit 7
#1308	1	Register R174 bit 8	#1324	1	Register R175 bit 8
#1309	1	Register R174 bit 9	#1325	1	Register R175 bit 9
#1310	1	Register R174 bit 10	#1326	1	Register R175 bit 10
#1311	1	Register R174 bit 11	#1327	1	Register R175 bit 11
#1312	1	Register R174 bit 12	#1328	1	Register R175 bit 12
#1313	1	Register R174 bit 13	#1329	1	Register R175 bit 13
#1314	1	Register R174 bit 14	#1330	1	Register R175 bit 14
#1315	1	Register R174 bit 15	#1331	1	Register R175 bit 15

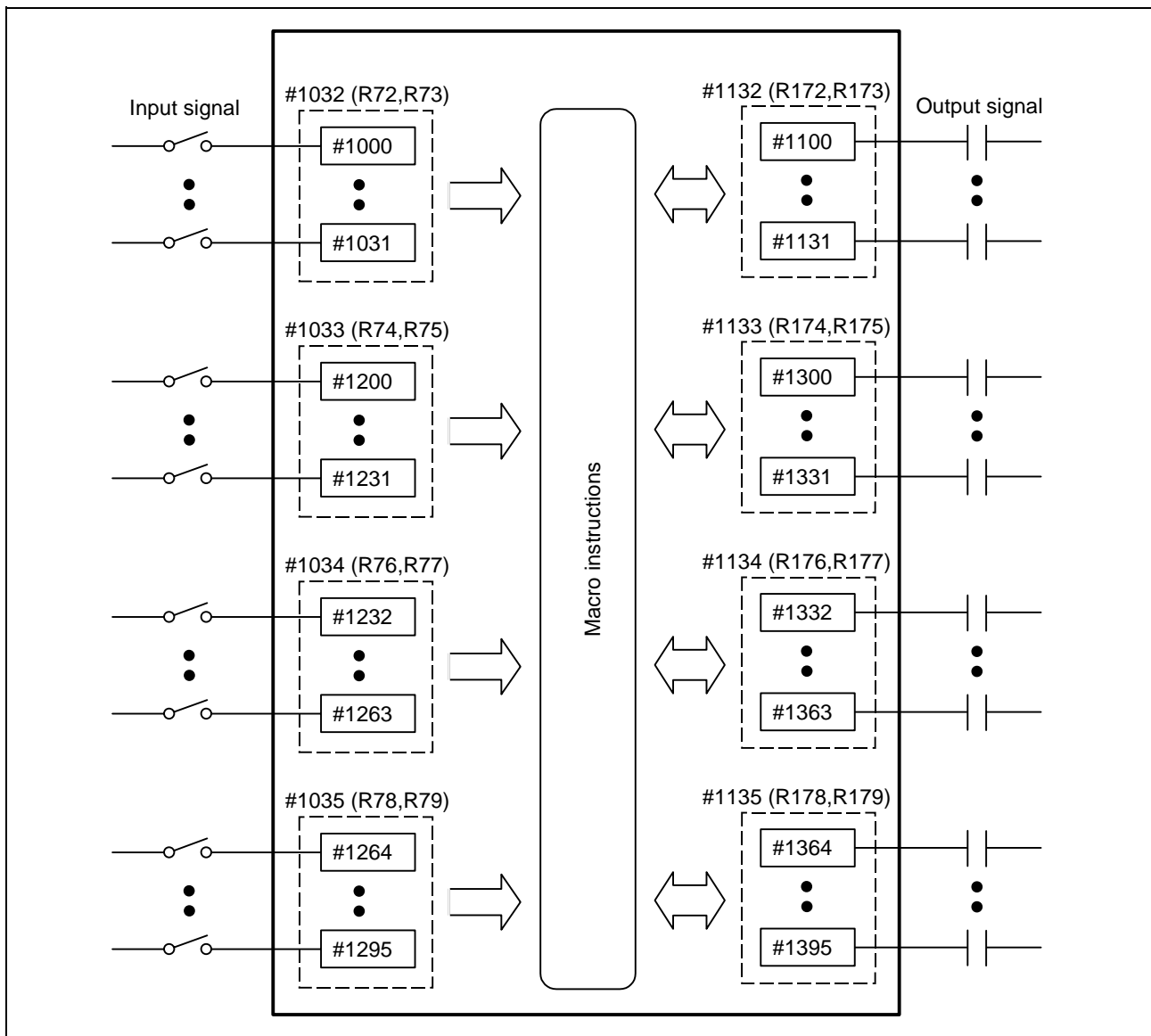
System variable	No. of points	Interface input signal	System variable	No. of points	Interface input signal
#1332	1	Register R176 bit 0	#1348	1	Register R177 bit 0
#1333	1	Register R176 bit 1	#1349	1	Register R177 bit 1
#1334	1	Register R176 bit 2	#1350	1	Register R177 bit 2
#1335	1	Register R176 bit 3	#1351	1	Register R177 bit 3
#1336	1	Register R176 bit 4	#1352	1	Register R177 bit 4
#1337	1	Register R176 bit 5	#1353	1	Register R177 bit 5
#1338	1	Register R176 bit 6	#1354	1	Register R177 bit 6
#1339	1	Register R176 bit 7	#1355	1	Register R177 bit 7
#1340	1	Register R176 bit 8	#1356	1	Register R177 bit 8
#1341	1	Register R176 bit 9	#1357	1	Register R177 bit 9
#1342	1	Register R176 bit 10	#1358	1	Register R177 bit 10
#1343	1	Register R176 bit 11	#1359	1	Register R177 bit 11
#1344	1	Register R176 bit 12	#1360	1	Register R177 bit 12
#1345	1	Register R176 bit 13	#1361	1	Register R177 bit 13
#1346	1	Register R176 bit 14	#1362	1	Register R177 bit 14
#1347	1	Register R176 bit 15	#1363	1	Register R177 bit 15



System variable	No. of points	Interface input signal	System variable	No. of points	Interface input signal
#1364	1	Register R178 bit 0	#1380	1	Register R179 bit 0
#1365	1	Register R178 bit 1	#1381	1	Register R179 bit 1
#1366	1	Register R178 bit 2	#1382	1	Register R179 bit 2
#1367	1	Register R178 bit 3	#1383	1	Register R179 bit 3
#1368	1	Register R178 bit 4	#1384	1	Register R179 bit 4
#1369	1	Register R178 bit 5	#1385	1	Register R179 bit 5
#1370	1	Register R178 bit 6	#1386	1	Register R179 bit 6
#1371	1	Register R178 bit 7	#1387	1	Register R179 bit 7
#1372	1	Register R178 bit 8	#1388	1	Register R179 bit 8
#1373	1	Register R178 bit 9	#1389	1	Register R179 bit 9
#1374	1	Register R178 bit 10	#1390	1	Register R179 bit 10
#1375	1	Register R178 bit 11	#1391	1	Register R179 bit 11
#1376	1	Register R178 bit 12	#1392	1	Register R179 bit 12
#1377	1	Register R178 bit 13	#1393	1	Register R179 bit 13
#1378	1	Register R178 bit 14	#1394	1	Register R179 bit 14
#1379	1	Register R178 bit 15	#1395	1	Register R179 bit 15

**(Note 1)** The last values of the system variables #1100 to #1135, #1300 to #1395 sent are retained as 1 or 0.  
(They are not cleared even with resetting.)

**(Note 2)** The following applies when any number except 1 or 0 is substituted into #1100 to #1131, #1300 to #1395.  
<Blank> is treated as 0.  
Any number except 0 and <Blank> is treated as 1.  
Any value less than 0.00000001 is indefinite.





### Tool offset

Tool data can be read and set using the variable numbers.

Variable number range			Details
	#1120 TofVal=0	#1120 TofVal=1	
#10001 to #10000+n	#2001 to #2000+n	#2701 to #2700+n	X shape compensation amount
#11001 to #11000+n	#2701 to #2700+n	#2001 to #2000+n	X wear compensation amount
#12001 to #12000+n	-		Tool compensation selection additional axis shape compensation amount
#13001 to #13000+n	-		Tool compensation selection additional axis wear compensation amount
#14001 to #14000+n	#2101 to #2100+n	#2801 to #2800+n	Z shape compensation amount
#15001 to #15000+n	#2801 to #2800+n	#2101 to #2100+n	Z wear compensation amount
#16001 to #16000+n	#2201 to #2200+n	#2901 to #2900+n	R shape compensation amount
#17001 to #17000+n	#2901 to #2900+n	#2201 to #2200+n	R wear compensation amount
#18001 to #18000+n	#2301 to #2300+n		Nose compensation amount

n in table corresponds to the tool No. Maximum n value is number of tool offset sets.

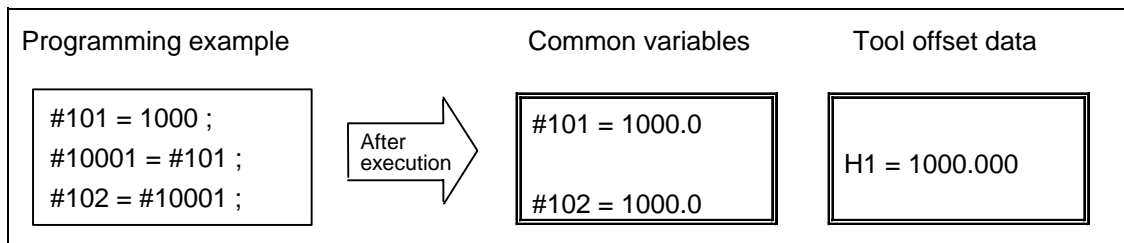
The numbers in the #10000 order and #2000 order are similar function.

The tool offset data is configured as data with a decimal point in the same way as for other variables.

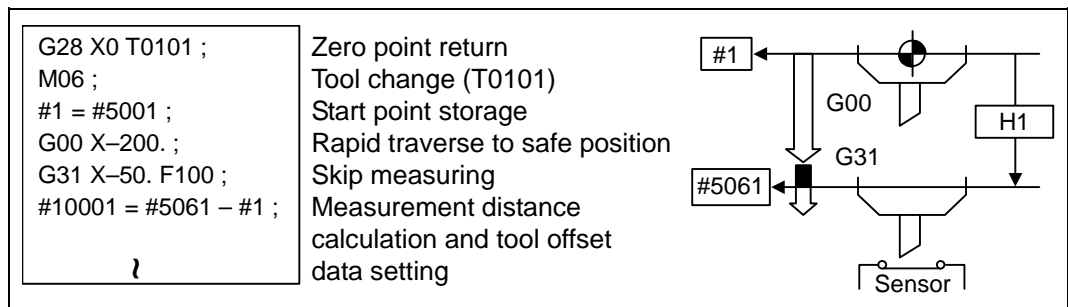
When "# 10001=1000;" is programmed, 1000.000 is set in tool offset data.

The load axis' tool compensation can be used for only either the 3rd axis or 4th axis. Selected by the parameter "#1520 TchG34".

The variable number corresponding to #2000 order's shape/wear compensation amount can be switched with the parameter "#1122 TofVal".



#### (Example 1) Calculation and tool offset data setting



**(Note 1)** In (Example 1) no consideration is given to the delay in the skip sensor signal. #5001 is the X-axis start point position and #5061 is the X-axis skip coordinates, and indicated is the position at which the skip signal is input while G31 is being executed.



### Workpiece coordinate system offset

By using variable numbers #5201 to #532n, it is possible to read out the workpiece coordinate system offset data or to substitute values.

**(Note)** The number of axes which can be controlled differs according to the specifications. The last digit in the variable number corresponds to the control axis number.

Coordinate name \ Axis No.	1st axis	2nd axis	3rd axis	Remarks
External workpiece offset	#5201	#5202	#5203	
G54	#5221	#5222	#5223	
G55	#5241	#5242	#5243	
G56	#5261	#5262	#5263	
G57	#5281	#5282	#5283	
G58	#5301	#5302	#5303	
G59	#5321	#5322	#5323	

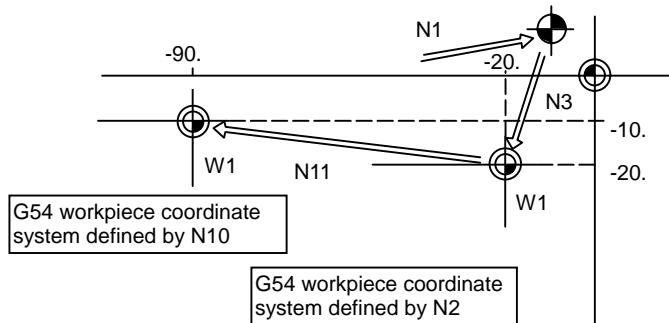
#### (Example 1)

```

N1 G28 X0 Z0 ;
N2 #5221 = -20. #5223 = -20. ;
N3 G00 G54 X0 Z0 ;
}

N10 #5221 = -10. #5223 = -90. ;
N11 G00 G54 X0 Z0 ;
}

M02 ;
    
```

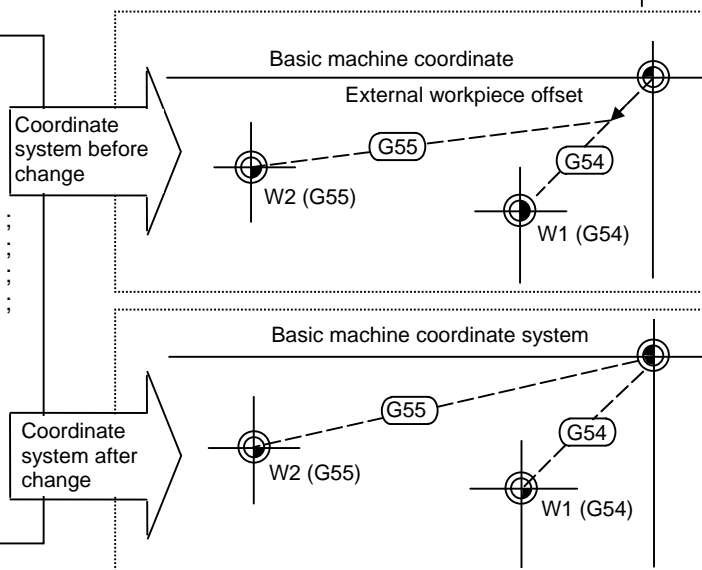


#### (Example 2)

```

}

N100 #5221 = #5221 + #5201 ;
      #5222 = #5222 + #5202 ;
      #5241 = #5241 + #5201 ;
      #5242 = #5242 + #5202 ;
      #5201 = 0 #5202 = 0 ;
}
    
```



This is an example where the external workpiece offset values are added to the workpiece coordinate (G54, G55) system offset values without changing the position of the workpiece coordinate systems.



### NC alarm (#3000)

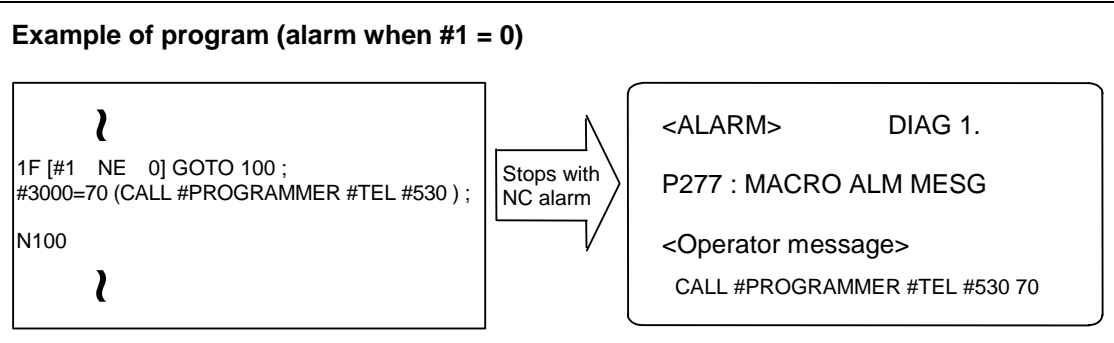
The NC unit can be forcibly set to the alarm state by using variable number #3000.

#### Format

<b>#3000 = 70 (CALL #PROGRAMMER #TEL #530) ;</b>	
70	Alarm number
CALL #PROGRAMMER #TEL #530	Alarm message

Any alarm number from 1 to 9999 can be specified.  
 The alarm message must be less than 31 characters long.  
 NC alarm 3 signal (program error) is output.

The "P277: MACRO ALM MESSG" appears in the <ALARM> column on "DIAG 1." screen while the alarm number and alarm message "70: (CALL #PROGRAMMER #TEL #530)" is indicated in the <Operator message>.



**(Note 1)** Alarm number 0 is not displayed and any number exceeding 9999 cannot be indicated.

**(Note 2)** The characters following the first alphabet letter in the right member is treated as the alarm message. Therefore, a number cannot be designated as the first character of an alarm message. It is recommended that the alarm messages be enclosed in round parentheses.

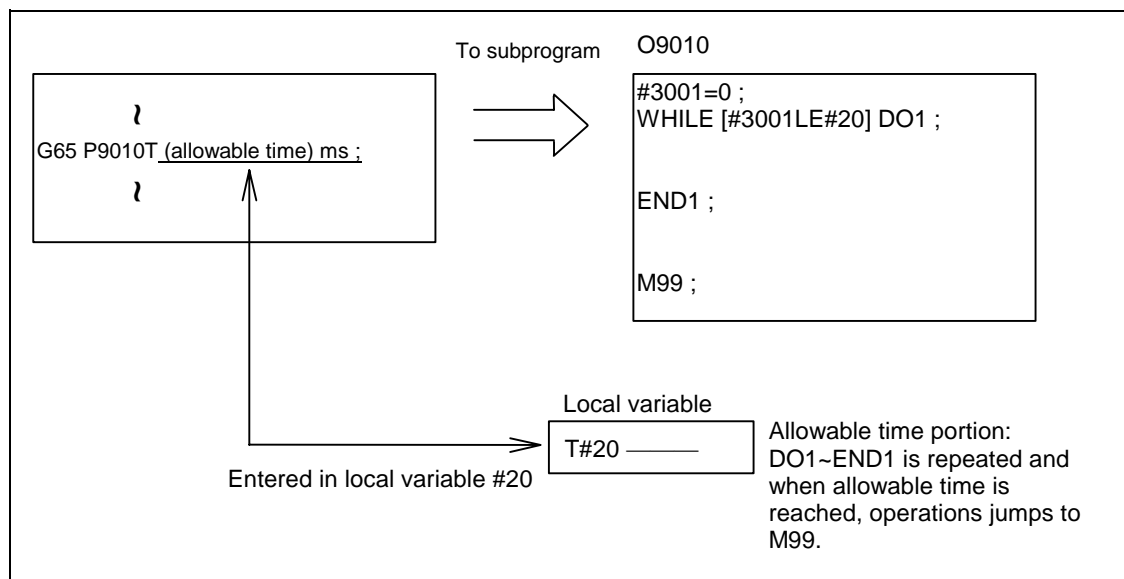


### Integrating time (#3001, #3002)

The integrating time can be read when the power is turned ON or automatic start or values can be substituted by using variable numbers #3001 and #3002.

Type	Variable number	Unit	Contents when power is switched on	Initialization of contents	Count conditions
Power ON	#3001	ms	Same as when power is turned OFF	Value substituted for variable	At all times while power is ON
Automatic start	#3002				In-automatic start

The integrating time returns to zero in about  $2.44 \times 10^{11}$  ms (approximately 7.7 years).



### Suppression of single block stop and miscellaneous function finish signal waiting

By substituting the values below in variable number #3003, it is possible to suppress single block stop in the subsequent blocks or to advance to the next block without waiting for the miscellaneous function (M, S, T, B) finish (FIN) signal.

#3003	Single block stop	Miscellaneous function finish signal
0	Not suppressed	Wait
1	Suppressed	Wait
2	Not suppressed	No wait
3	Suppressed	No wait

**(Note 1)** #3003 is cleared to zero by resetting.



### Feed hold, feedrate override, G09 valid/invalid

By substituting the values below in variable number #3004, it is possible to make the feed hold, feedrate override and G09 functions either valid or invalid in the subsequent blocks.

#3004 contents (value)	Bit 0	Bit 1	Bit 2
	Feed hold	Feedrate override	G09 check
0	Valid	Valid	Valid
1	Invalid	Valid	Valid
2	Valid	Invalid	Valid
3	Invalid	Invalid	Valid
4	Valid	Valid	Invalid
5	Invalid	Valid	Invalid
6	Valid	Invalid	Invalid
7	Invalid	Invalid	Invalid

**(Note 1)** #3004 is cleared to zero by resetting.

**(Note 2)** The functions are valid when the above bits are 0 and invalid when they are 1.

**(Note 3)** When the feed hold is set to invalid with #3004, the following will occur when the feed hold switch is pressed.

- During thread cutting, block stop will be carried out at the end of the next block of the block where thread cutting is completed.
- During tapping with tap cycle, block stop will be carried out after the operation of R point return.
- In the case other than above, block stop will be carried out after termination of the currently executing block.



### Message display and stop

By using variable number #3006, the execution is stopped after the previous block has been executed and, if message display data have been commanded, then the corresponding message will be indicated on the operator message area.

#### Format

```
#3006=1 (TAKE FIVE) ;
TAKE FIVE           Message
```

The message should not be longer than 31 characters and it should be enclosed within round parentheses.



### Mirror image

By reading variable number #3007, it is possible to ascertain the status of mirror image at a particular point in time for each axis.

The axes correspond to the bits of #3007. When the bits are 0, it means that the mirror image function is invalid; when they are 1, it means that it is valid.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nth axis													4	3	2	1



## G command modals

Using variable numbers #4001 to #4021, it is possible to read the G modal commands which have been issued up to the block immediately before.

Similarly, it is possible to read the modals in the block being executed with variable numbers #4201 to #4221.

Variable number		Function
Pre-read block	Execution block	
#4001	#4201	Interpolation mode: G00:0, G01:1, G02:2, G03:3, G33:33
#4002	#4202	Plane selection: G17:17, G18:18, G19:19
#4003	#4203	Absolute/incremental: G90:90, G91:91
#4004	#4204	Barrier check: G22:22, G23:23
#4005	#4205	Feed designation: G94:94, G95:95
#4006	#4206	Inch/metric: G20:20, G21:21
#4007	#4207	Nose R compensation: G40:40, G41:41, G42:42, G46:46
#4008	#4208	No variable No.
#4009	#4209	Fixed cycle: G80:80, G70~79:70~79, G83~G85:83~85, G83.2:83.2, G87~G89:87~89
#4010	#4210	Return level: G98:98, G99:99
#4011	#4211	
#4012	#4212	Workpiece coordinate system: G54~G59:54~59
#4013	#4213	Acceleration/deceleration:G61~G64:61~64
#4014	#4214	Macro modal call: G66:66, G66.1:66.1, G67:67
#4015	#4215	
#4016	#4216	No variable No.
#4017	#4217	Constant surface speed control: G96:96, G97:97
#4018	#4218	
#4019	#4219	
#4020	#4220	
#4021	#4221	

**(Example)**

```
G29 X0 Z0 ;
G00 X150. Z200 ;
G65 P300 G02 W-30. K-15. F1000 ;
M02 ;
O300
#1 = #4001 ; → Group 01 G modal (pre-read) #1 = 2.0
#2 = #4201 ; → Group 01 G modal (now being executed) #2 = 0.0
G#1 W#24 ;
M99 ;
%
```





### Other modals

Using variable numbers #4101 to #4120, it is possible to read the modal commands assigned up to the block immediately before.

Similarly, it is possible to read the medals in the block being executed with variable numbers #4301 to #4320.

Variable number		Modal information
Pre-read	Execution	
#4101	#4301	
#4102	#4302	
#4103	#4303	
#4104	#4304	
#4105	#4305	
#4106	#4306	
#4107	#4307	
#4108	#4308	
#4109	#4309	Feedrate F
#4110	#4310	

Variable number		Modal information
Pre-read	Execution	
#4111	#4311	
#4112	#4312	
#4113	#4313	Miscellaneous function M
#4114	#4314	Sequence number N
#4115	#4315	Program number O
#4116	#4316	
#4117	#4317	
#4118	#4318	
#4119	#4319	Spindle function S
#4120	#4320	Tool function T

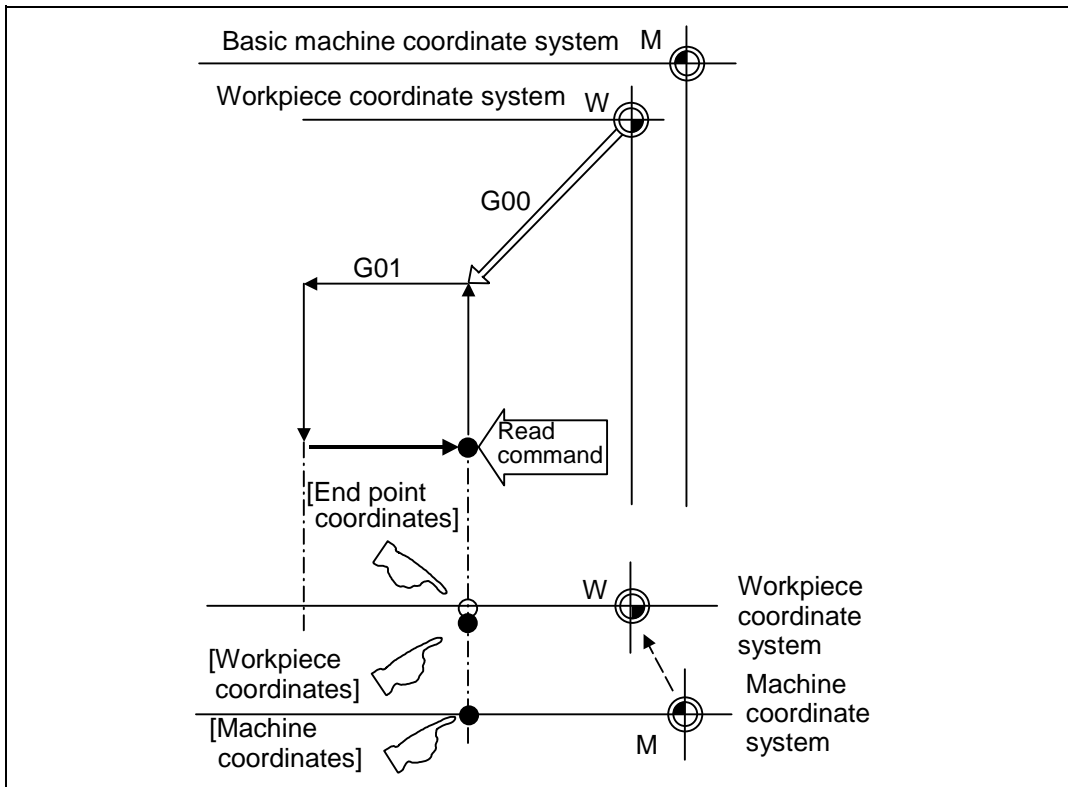


### Position information

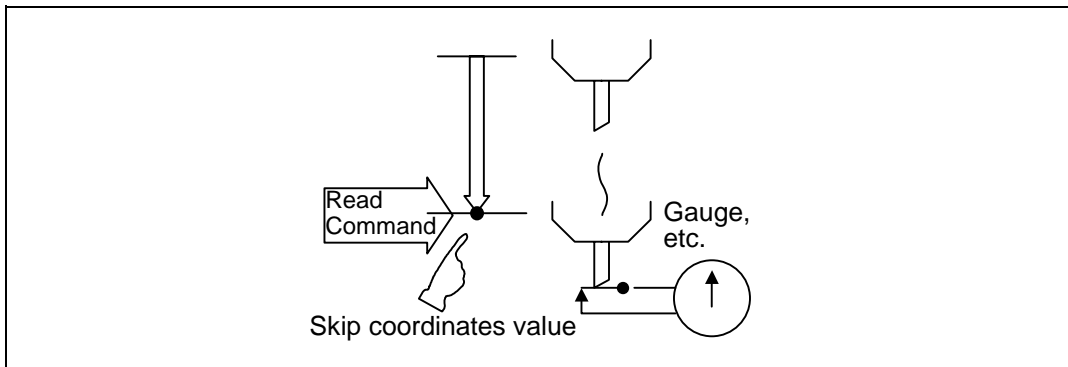
Using variable numbers #5001 to #5104, it is possible to read the servo deviation amounts, skip coordinates, work coordinates, machine coordinates and end point coordinates in the block immediately before.

Position information / Axis No.	End point coordinate of block immediately before	Machine coordinate	Workpiece coordinate	Skip coordinate	Servo deviation amount
1	#5001	#5021	#5041	#5061	#5101
2	#5002	#5022	#5042	#5062	#5102
3	#5003	#5023	#5043	#5063	#5203
Remarks (reading during movement)	Yes	No	No	Yes	Yes

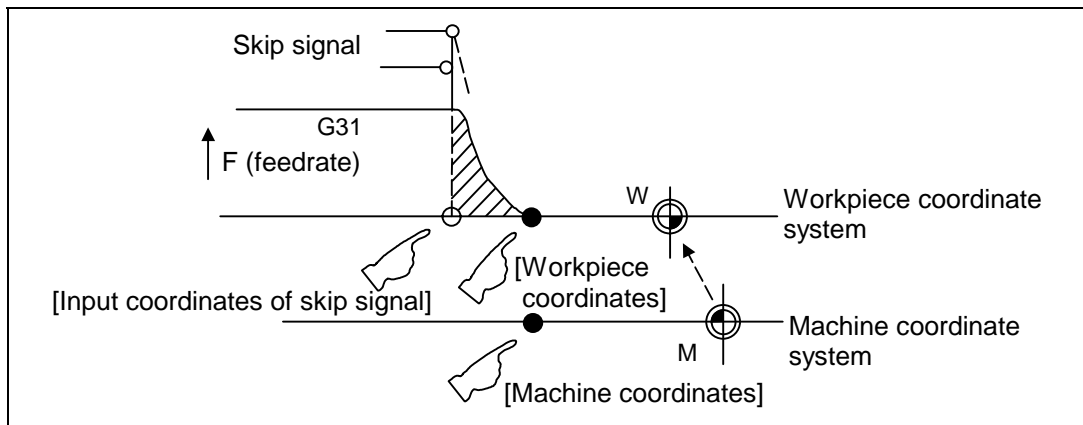
**(Note)** The number of axes which can be controlled differs according to the NC specifications. The last digit of the variable number corresponds to the control axis number.



- (1) The positions of the end point coordinates and skip coordinates are positions in the workpiece coordinate system.
- (2) The end point coordinates, skip coordinates and servo deviation amounts can be read even during movement. However, it must first be checked that movement has stopped before reading the machine coordinates and workpiece coordinates.
- (3) The position where the skip signal is turned ON in the G31 block is indicated for the skip coordinates. The end point position is indicated when the skip signal has not been turned ON. (For further details, refer to the section on tool length measurement.)



- (4) The tool nose position where the tool offset and other such factors are not considered is indicated as the end point position. The tool reference point position with consideration given to tool offset is indicated for the machine coordinates, workpiece coordinates and skip coordinates.



For "●", check stop and then proceed to read.  
 For "○", reading is possible during movement.

The position of the skip signal input coordinate value is the position in the workpiece coordinate system. The coordinate value in variable numbers #5061 to #5064 memorize the moments when the skip input signal during movement was input and so they can be read at any subsequent time.

For further details, reference should be made to the section on the skip function.



## Variable name setting and quotation

Any name (variable name) can be given to common variables #500 to #519. It must be composed of not more than 7 alphanumeric characters and it must begin with a letter. Do not use "#" in the variable name. An alarm will result at execution when this sign is used.

## Format

```
SETVn [NAME1, NAME2,.....];
```

n	Head number of variable which assigns name
NAME1	#n name (variable name)
NAME2	#n + 1 name (variable name)

Variable names are separated by a comma (,).

## Detailed description

- (1) Once variable names have been set, they will not be cleared even when the power is turned OFF.
- (2) The variables in the program are quoted as variable names. In cases like this, the variables should be enclosed in square parentheses.

**(Example 1)** G01X [#POINT1];

- (3) The variable numbers, data and variable names appear on the CRT screen of the setting and display unit.

**(Example 2)**

Program... SETVN500 [A234567, DIST, TOOL25];

[Common variables]		
#500	-12345.678	A234567
#501	5670.000	DIST
#502	-156.500	TOOL25
~~~~~		
#518	10.000	NUMBER
Common variable #(502) DATA (-156.5) NAME (TOOL25)		

**(Note)** Do not use characters (SIN, COS, etc.) predetermined by the NC for use as operation commands at the head of the variable name.



### External workpiece coordinate system offset

The external workpiece coordinate system offset can be read using variables #2501 to #2601. By substituting a value in these variables, the external workpiece coordinate system offset can be changed.

Axis No.	External workpiece coordinate system offset
1	#2501
2	#2601



### Number of workpiece machining times

The number of workpiece machining times can be read using variables #3901 and #3902. By substituting a value in these variables, the number of workpiece machining times can be changed.

Type	Variable No.	Data setting range
Number of workpiece machining times	#3901	0 to 999999
Maximum workpiece value	#3902	

**(Note)** Always input the number of workpiece machining times as a positive value.



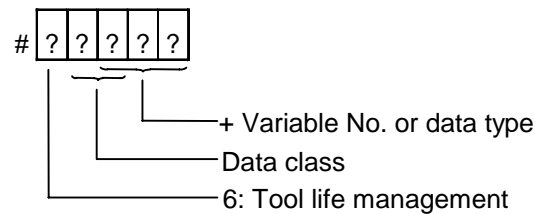
### Tool life management

#### (1) Definition of variable numbers

- (a) Designation of group No.  
#60000

The tool life management data group No. to be read with #60001 to #64700 is designated by substituting a value in this variable. If a group No. is not designated, the data of the group registered first is read. This is valid until reset.

- (b) Tool life management system variable No. (Read)  
#60001 to #64700



- (c) Details of data classification

Data class	M System	L System	Remarks
00	For control	For control	Refer to following types
05	Group No.	Group No.	Refer to registration No.
10	Tool No.	Tool No.	Refer to registration No.
15	Tool data flag	Method	Refer to registration No.
20	Tool status	Status	Refer to registration No.
25	Life data	Life time/No. of times	Refer to registration No.
30	Usage data	Usage time/No. of times	Refer to registration No.
35	Tool length compensation data	-	Refer to registration No.
40	Tool radius compensation data	-	Refer to registration No.
45	Auxiliary data	-	Refer to registration No.

The group No., L System method, and life data are common for the group.

- (d) Registration No.

<b>M System</b>	1 to 200
<b>L System</b>	1 to 16

- (e) Data type

Type	M System	L System	Remarks
1	Number of registered tools	Number of registered tools	
2	Life current value	Life current value	
3	Tool selected No.	Tool selected No.	
4	Number of remaining registered tools	Number of remaining registered tools	
5	Signal being executed	Signal being executed	
6	Cutting time cumulative value (minute)	Cutting time cumulative value (minute)	
7	Life end signal	Life end signal	
8	Life prediction signal	Life prediction signal	

Variable No.	Item	Type	Details	Data range
60001	Number of registered tools	System common	Total number of tools registered in each group.	0 to 80
60002	Life current value	For each group (Designate group No. #60000)	Usage time/No. of uses of tool being used. Usage data of tool being used (If tool uses several compensation numbers, the total of the usage data for each compensation No.).	0 to 999999 min 0 to 999999 times
60003	Tool selected No.		Registration No. of tool being used. Designated group's selection tool registration No. (If tool is not selected, the first tool of ST:1, or if ST:1 is not used, the first tool of ST:0. When all tools have reached their lives, the last tool).	0 to 16
60004	Remaining number of registered tools		Total number of "usable" tools in the group. Number of tools registered in designated group for which ST is 0: Not used.	0 to 16
60005	Signal being executed		"1" when this group is used in program being executed. "1" when tool in designated group is selected.	0/1
60006	Cutting time cumulative value (minute)		Indicates the time that this group is used in the program being executed.	
60007	Life end signal		"1" when lives of all tools in this group have been reached. "1" when all tools registered in designated group reach lives.	0/1
60008	Life prediction signal		"1" when new tool is selected with next command in this group. "1" when there is a tool for which ST is 0: Not used in the tools registered the designated group, and there are no tools for which ST is 1: Tools in use.	0/1

Variable No.	Item	Type	Details	Data range
60500 +***	Group No.	Each group/registration No.	This group's No.	1 to 9999
61000 +***	Tool No.	(Group No. #60000/registration No. *** is designated.)	Tool's No. and compensation No. Tool No. + compensation No. (When tool No. = 22 and compensation No. = 01, 2201=899H)	0 to 9999
61500 +***	Method		Whether to management this group's life as time or No. of uses. 0: Time, 1: No. of uses	0/1
62000 +***	Status	Note the group No., method and life are common for the groups.	Tool usage state 0: Not used tool 1: Tool being used 2: Normal life tool 3: Tool skip tool	0 to 3
62500 +***	Life time/No. of uses		This group's tool life value	0 to 999999 min 0 to 999999 times
63000 +***	Usage time/No. of uses			0 to 999999 min 0 to 999999 times
63500 +***	-			
64000 +***	-			
64500 +***	-			





### Example of program

**(1) Normal commands**

```
#101 = #60001 ; ... Reads the number of registered tools.
#102 = #60002 ; ... Reads the life current value.
#103 = #60003 ; ... Reads the tool selection No.
#60000 = 10 ; ..... Designates the group No. of the life data to be
                    read.
#104 = #60004 ; ... Reads the remaining number of registered
                    tools in group 10.
#105 = #60005 ; ... Reads the signal being executed in group 10.
#111 = #61001 ; ... Reads the group 10, #1 tool No.
#112 = #62001 ; ... Reads the group 10, #1 status.
#113 = #61002 ; ... Reads the group 10, #2 tool No.
%
```

} Designated program No. is valid until reset.

**(2) When group No. is not designated**

```
#104 = #60004 ; ... Reads the remaining number of registered tools in the group registered
                    first.
#111 = #61001 ; ... Reads the #1 tool No. in the group registered first.
%
```

**(3) When non-registered group No. is designated (Group 9999 does not exist.)**

```
#60000 = 9999 ; ... Designates the group No.
#104 = #60004 ; ... #104 = -1.
```

**(4) When registration No. not used is designated (Group 10 has 15 tools)**

```
#60000 = 10 ; ..... Designates the group No.
#111 = #61016 ; ... #101 = -1.
```

**(5) When registration No. out of the specifications is designated**

```
#6000 = 10 ;
#111 = #61017 ; ... "Program error (P241)"
```

**(6) When tool life management data is registered with G10 command after group No. is designated**

```
#60000 = 10 ; ..... Designates the group No.
G10 L3 ; ..... Starts the life management data registration.
P10 LLn NNn ; .... 10 is the group No., Ln is the life per tool,
                    Nn is the method.
TTn ; ..... Tn is the tool No.
:
G11 ; ..... Registers the group 10 data with the G10
                    command.
#111 = #61001 ; ... Reads the group 10, #1 tool No.
G10 L3 ; ..... Starts the life management data registration.
P1 LLn NNn ; .... 1 is the group No., Ln is the life per tool,
                    Nn is the method.
TTn ; ..... Tn is the tool No.
:
G11 ; ..... Registers the life data with the G10 command.
                    (The registered data is deleted.)
#111 = 61001 ; .... Group 10 does not exist. #111 = -1.
```

} The group 10 life data is registered.

} The life data other than group 10 is registered.



#### Precautions

- (1) If the tool life management system variable is commanded without designating a group No., the data of the group registered at the head of the registered data will be read.
- (2) If a non-registered group No. is designated and the tool life management system variable is commanded, "-1" will be read as the data.
- (3) If an unused registration No. tool life management system variable is commanded, "-1" will be read as the data.
- (4) Once commanded, the group No. is valid until NC reset.
- (5) When the tool life control I specifications are provided with the L System, the tool life management system variable specifications are not provided. "Program error (P241)" will occur if commanded.



### Time reading variables

System data can be read in with the system variables.

**(Note)** These can be used only with some models.

Variable No.	Application
#100000	Parameter # designation
#100001	System reserved
#100002	Axis No./spindle No. designation
#100010	Parameter value read

The parameter values are read in with the following four blocks using these four system variables.

#100000 = 1001 ; ····· Designates the parameter #.

#100001 = 0 ;

#100002 = 1 ; ····· Designates the axis No./spindle No.

#100 = #100010 ; ····· Reads the parameter value.

#### (1) Parameter # designation (#100000)

The parameter to be read in is designated by substituting the parameter # in this system variable.

If the parameters are read without designating this number, the parameters will be read in the same manner as if the minimum parameter # (#1) is designated. Once designated, the setting is held until the parameter # is designated again or until it is reset.

A program error (P39) will occur if a parameter # that does not exist is set.

#### (2) System reserved (#100001)

Always set "0".

When using the PLC axis, set "10".

#### (3) Axis/spindle No. designation (#100002)

(a) System variable for axis/spindle No. designation

The axis No./spindle No. of the parameter to be read in is designated by substituting an index value in this system variable. This designation will be ignored when reading in parameters that are not for a specific axis or spindle.

A program error (P39) will occur if an axis/spindle No. that does not exist is set.

(b) Index values

Index values	Axis parameter	Spindle parameter
1	1st axis	1st spindle
2	2nd axis	2nd spindle
3	3rd axis	-

#### (4) Parameter read (#100010)

The designated parameter data is read with this system variable.

The following data is read according to the parameter type.

Type	Read data
Value	The values displayed on the Parameter screen are output.
Text	ASCII codes are converted into decimal values.



## Examples of using time reading variable

**(1) To read the parameter [#1002 axisno Number of axes]**

#100000 = 1002 ; ..... Designates [#1002].  
 #100001 = 0 ;  
 #101 = #100010 ; ..... Reads the number of axes.  
 #100001 = 10 ; ..... Designates [PLC axis].  
 #110 = #100010 ; ..... Reads the number of PLC axes.

**(2) To read the axis parameter [#2037 G53ofs #1 reference point]**

[Conditions]

	<1st axis>	<2nd axis>
#2037 G53ofs	1000.000	200.000

#100002 = 1 ; ..... Designates [1st axis].  
 #100000 = 2037 ; ..... Designates [#2037].  
 #101 = #100010 ; ..... Reads the [#1 reference point] for the 1st axis.  
 (#101 = 100.000.)  
 #100002 = 2 ; ..... Designates [2nd axis].  
 #102 = #100010 ; ..... Reads the [#1 reference point] for the 2nd axis.  
 (#102 = 200.000.)

**(3) To read each parameter for axis and spindle**

#100002 = 1 ; ..... Designates [1st spindle].  
 #100000 = 3001 ; ..... Designates [#3001].  
 #101 = #100010 ; ..... Reads the [#3001 slimt1 Limit rotation speed gears 00]  
 for 1st spindle.  
 #100000 = 3002 ; ..... Designates [#3002].  
 #102 = #100010 ; ..... Reads the [#3002 slimt2 Limit rotation speed gears 01]  
 for 1st spindle.



### Example of parameter read macro program

#### <Macro specifications>

Q341 A\_ Q\_ . ;

A\_ . . . Storage common variable

Designates the common variable No. for storing the data read in.

Q\_ . . Parameter # designation

For an axis/spindle parameter, designates the axis/spindle No. with one digit after the decimal point.

#### <Macro contents>

```
#100000 = FIX [#17] ;
```

..... Designates parameter #.

```
#100002 = FIX [#17*10] MOD 10 ;
```

..... Designates axis/spindle No.

```
#[#1] = #100010 ;
```

..... Reads parameter data.

```
M99 ;
```



### Precautions for reading parameters

- (1) The number of axes and spindles is the maximum number in the specifications.
- (2) The inch/metric changeover function for the setting and display is valid even for the data read in.



### Reading PLC data

PLC data can be read in with the system variables.

**(Note1)** These can be used only with some models.

**(Note2)** The read devices are limited.

Variable No.	Application
#100100	Device type designation
#100101	Device No. designation
#100102	Number of read bytes designation
#100103	Read bit designation
#100110	PLC data read

The PLC data is read in with the following five blocks using these five system variables.

#100100 = 1 ; ..... Designates the device type.

#100101 = 0 ; ..... Designates the device No.

#100102 = 1 ; ..... Designates the number of bytes.

#100103 = 2 ; ..... Designates the bit. (Valid only when reading word device bits.)

#100 = #100110 ; ... Reads in the PLC data.

#### (1) Device designation (#100100)

##### (a) System variable for device designation

The type of device to be read in can be designated by substituting the device designation value in this system variable.

If the data is read without designating this number, the data will be read in the same manner as if the minimum device designation value (0: M device) is designated. Once designated, the setting is held until the device is designated again or until it is reset.

A program error (P39) will occur if a device that does not exist is set.

##### (b) Device designation values

Device designation value	Device			Device designation value	Device		
	Device	Unit	Device No.		Device	Unit	Device No.
0	M	Bit	M0 to M8191	6	R	Word	R0 to R8191 (Note)
1	D	Word	D0 to D1023	7	T	Bit	T0 to T255
2	C	Bit	C0 to C127	9	SM	Bit	SM0 to SM127
4	X *	Bit	X0 to X4BF	10	F	Bit	F0 to F127
5	Y *	Bit	Y0 to Y53F	13	L	Bit	L0 to L255

The unit is the amount of data per device No. "Word" has 16 bits, and "Bit" has one bit.

\* is a device that expresses the device No. as a hexadecimal.

**(Note)** The device which is used in 2-part system cannot be used because E60/E68 is 1-part system specifications.

### (2) Device No. designation (#100101)

The device to be read in is designated by substituting the device No. in this system variable. Convert a device expressed as a hexadecimal into a decimal when designating. If the data is read without designating this number, the data will be read in the same manner as if the minimum device No. (0) is designated. Once designated, the setting is held until the device No. is designated again or until it is reset. A program error (P39) will occur if a device No. that does not exist is set.

### (3) Number of bytes designation (#100102)

(a) System variable for number of bytes designation  
The reading size is designated by substituting the number of bytes designation value in this system variable. If the data is read without designating this number, the data will be read in the same manner as if the minimum number of bytes designation value (0: bit designation) is designated. Once designated, the setting is held until the number of bytes is designated again or until it is reset. A program error (P39) will occur if a number of bytes that does not exist in the specifications is set.

(b) Number of bytes designation value

Number of bytes designation value	Read in data			Operation	
	Size	Sign	Range	Word device	Bit device
0	1 bit	-	0 to 1	The number of bits designated is read in.	The bits for the designated device No. are read in.
1	1 byte	No	0 to 255	The low-order byte is read in.	8 bits are read in from the designated device No.
101		Yes	-128 to 127		
2	2 bytes	No	0 to 65535	Two bytes are read in.	16 bits are read in from the designated device No.
102		Yes	-32768 to 32767		
4	4 bytes	No	0 to 4294967295	The designated device (L) and next device (H) are read in.	32 bits are read in from the designated device No.
104		Yes	-2147483648 to 2147483647		

0 to 4 are designated without a sign, and 101 to 104 are designated with a sign.

### (4) Bit designation (#100103)

(a) System variable for bit designation  
The bit to be read in is designated by substituting the bit designation value in this system variable. This designation is valid only when reading the bits for a 16-bit device, and is invalid in all other cases. If the data is read without designating this number, the data will be read in the same manner as if the minimum bit designation value (0: bit 0) is designated. Once designated, the setting is held until the bit is designated again or until it is reset. A program error (P39) will occur if a bit that does not exist is set.

(b) Bit designation values

Bit designation value	Read in bit
0	Bit 0
1	Bit 1
:	:
15	Bit 15

#### (5) PLC data read (#100110)

The data for the designated device is read in with this system variable.

Refer to the table for number of bytes designation for details on the range of data read in.



#### Examples of programs for reading PLC data

##### (1) To read a bit device

#100100 = 0 ; ..... Designates [M device].  
 #100101 = 0 ; ..... Designates [Device No. 0].  
 #100102 = 0 ; ..... Designates [Bit].  
 #100 = #100110 ; ... Reads M0 (one bit).  
 #100102 = 1 ; ..... Designates [1 byte].  
 #101 = #100110 ; ... Reads M0 to M7 (eight bits).  
 (If M7 to M0 is 0001 0010, this will be #102 = 18 (0x12).)  
 #100102 = 102 ; ... Designates [Signed two bytes].  
 #102 = #100110 ; ... Reads M0 to M15 (16 bits).  
 (If M15 to M0 is 1111 1110 1101 1100, this will be  
 #102 = -292 (0xFEDC).)  
 #100102 = 4 ; ..... Designates [4 bytes].  
 #104 = #100110 ; ... Reads M0 to M31 (32 bits).  
 (If M31 to M0 is 0001 0010 0011 0100 0101 0110 0111 1000,  
 this will be #104 = 305419896 (0x12345678).)

##### (2) To read a word device

#100100 = 1 ; ..... Designates [D device].  
 #100101 = 0 ; ..... Designates [Device No. 0].  
 #100102 = 0 ; ..... Designates [Bit].  
 #100103 = 1 ; ..... Designates [Bit 1].  
 #100 = #100110 ; ... Reads the D0 bit 1.  
 (If D0 = 0x0102, this will be #101 =1.)  
 #100102 = 1 ; ..... Designates [1 byte].  
 #101 = #100110 ; ... Reads the low-order byte of D0.  
 (If D0 = 0x0102, this will be #101 =2.)  
 #100102 = 2 ; ..... Designates [2 bytes].  
 #102 = #100110 ; ... Reads D0. (If D0 = 0x0102, this will be #102 =258.)  
 #100102 = 104 ; ... Designates [Signed four bytes].  
 #104 = #100110 ; ... Reads D0 and D1.  
 (If D0 = 0xFFFFE and D1 = 0xFFFF, this will be #104 =-2.)





### Examples of using macro program for reading PLC data

#### <Macro specifications>

G340 F\_ A\_ Q\_ H\_ ;

F\_ ..... Number of bytes designation

A\_ ..... Device designation

Q\_ ..... Device No. designation

H\_ ..... Storage common variable

F0 ... Designates bit.

F1 ... Designates one byte.

F2 ... Designates two bytes.

A0 ... Designates M.

A1 ... Designates D.

A2 ... Designates C.

A4 ... Designates X.

A5 ... Designates Y.

A6 ... Designates R.

A7 ... Designates T.

For a bit, designates the bit with two digits after the decimal point.

For a byte, a decimal value is not designated.

Designates the common variable No. for storing the read data.

#### <Macro contents>

#100100 = #1 ;

#100101 = FIX [#17] ;

#100102 = #9 ;

#100103 = FIX [#17\*100] MOD 100 ;

#[#11] = #100110 ;

M99 ;

... Designates device.

... Designates device No.

... Designates number of bytes.

... Designates bit.

... Reads PLC data.



### Precautions for reading PLC data

- (1) As the PLC data is read asynchronously from the ladder execution, the data is not necessarily from the running program. Take care when reading devices that change.
- (2) If reading of a device that does not exist is attempted by designating the device No. and number of bytes, the 0 value will be read in only for the sections that do not exist.



### Time reading variables

The following operations can be carried out using the system variable extension for the user macro time.

- (1) By adding time information system variable #3011 and #3012, the current date (#3011) and current time (#3012) can be read and written.
- (2) By adding parameter #1273/bit1, the unit (millisecond unit/hour unit) of the cumulative time during system variable #3002 automatic start can be changed.

Variable No.	Details
#3001	The cumulative time during power ON can be read and the value can be substituted. The unit is a millisecond unit.
#3002	The cumulative time during automatic start can be read and the value can be substituted. The unit can be changed between millisecond and hour with parameter #1273/bit1.
#3011	The current date can be read and written. YYYY/MM/DD is read as a YYYYMMDD value. When a YYYYMMDD value is written in, it will be set as YY/MM/DD (the last two digits of the year are displayed). Command range for year/month/date setting Year (YYYY) : 2000 to 2099 Month (MM) : 1 to 12 Date (DD) : 1 to maximum number of days in month
#3012	The current time can be read and written. HH/MM/SS is read as a HHMMSS value. When an HHMMSS value is written in, it will be set as HH/MM/DD. Command range for hour/minute/second setting Hour (HH) : 0 to 23 (24-hour system) Minute (MM) : 0 to 59 Second (SS) : 0 to 59

- (3) The cumulative time returns to 0 at approx.  $2.44 \times 10^{11}$  milliseconds (approx. 7.7 years).
- (4) If a negative value or a value exceeding 244335917226 milliseconds (67871.08811851 hours for #3002 time designation) is set for the cumulative time, a program error (P35) will occur.
- (5) If a value exceeding the command range is set for the date or time, a program error (P35) will occur.
- (6) Always set the month/date/hour/minute/second as a two-digit value when setting the date and time.  
If the value only has one digit, always add 0. (February 14, 2001  $\Rightarrow$  #3001= 20010214 ;, etc.)



## Examples of using time reading variable

- (Example 1)** To read the current date (February 14, 2001) in common variable #100  
 #100 = #3011 ; (20010214 is inserted in #100)
- (Example 2)** To write current time (18 hours, 13 minutes, 6 seconds) into system variable #3012  
 #3012 = 181306 ; (The command value cumulative time #2: time is set to 18:13:06.)
- (Example 3)** By setting the following type of program, the machining start/end time  
 (year/month/date/hour/minute/second) can be viewed.

```
#100=#3011 ; ⇒ Machining start year/month/date
#101=#3012 ; ⇒ Machining start hour/minute/second
G28 X0 Y0 Z0 ;
G92 ;
G0 X50. ;
.
.
.
#102=#3011 ; ⇒ Machining end year/month/date
#103=#3012 ; ⇒ Machining end hour/minute/second
M30 ;
```



## Limits and precautions for using time reading variable

- (1) #3011 reads the date as an eight-digit value, so the difference of the two dates read in will not be the difference of days.
- (2) #3012 reads the time as a six-digit value, so the difference of the two times read in will not be the difference of hours.

### 13.9.6 Operation commands

A variety of operations can be performed between variables.



#### Command format

**#i = <formula> ;**

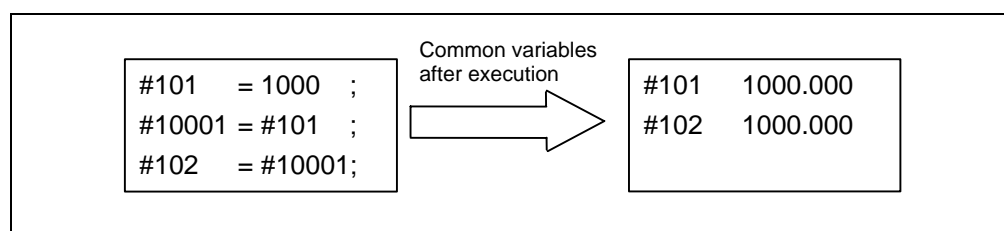
<Formula> is a combination of constants, variables, functions and operators. Constants can be used instead of #j and #k below.

(1) Definition and substitution of variables	#i = #j	Definition, substitution
(2) Addition operation	#i = #j + #k	Addition
	#i = #j - #k	Subtraction
	#i = #j OR #k	Logical sum (at every bit of 32 bits)
	#i = #j XOR #k	Exclusive OR (at every bit of 32 bits)
(3) Multiplication operation	#i = #j * #k	Multiplication
	#i = #j / #k	Division
	#i = #j MOD #k	Remainder
	#i = #j AND #k	Logical product (at every bit of 32 bits)
(4) Functions	#i = SIN [#k]	Sine
	#i = COS [#k]	Cosine
	#i = TAN [#k]	Tangent tan $\theta$ uses $\sin\theta/\cos\theta$ .
	#i = ASIN [#k]	Arcsine
	#i = ATAN [#k]	Arctangent (ATAN or ATN may be used)
	#i = ACOS [#k]	Arccosine
	#i = SQRT [#k]	Square root (SQRT or SQR may be used)
	#i = ABS [#k]	Absolute value
	#i = BIN [#k]	Conversion from BCD to BIN
	#i = BCD [#k]	Conversion from BIN to BCD
	#i = ROUND [#k]	Rounding off (ROUND or RND may be used)
	#i = FIX [#k]	Discard fractions less than 1
	#i = FUP [#k]	Add for fractions less than 1
	#i = LN [#k]	Natural logarithm
#i = EXP [#k]	Exponent with e (=2.718...) as bottom	

**(Note 1)** A value without a decimal point is basically treats as a value with a decimal point at the end (1 = 1.000).

**(Note 2)** Offset amounts from #10001 and workpiece coordinate system offset values from #5201 are handled as data with a decimal point. Consequently, data with a decimal point will be produced even when data without a decimal point have been defined in the variable numbers.

#### (Example)



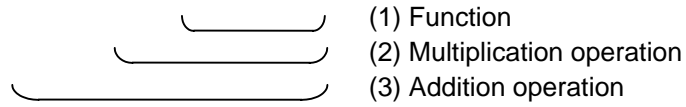
**(Note 3)** The <formula> after a function must be enclosed in the square parentheses.



### Sequence of operations

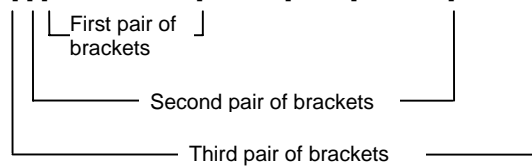
- (1) The sequence of the operations (1) to (3) is, respectively, the functions followed by the multiplication operation followed in turn by the addition operation.

$$\#101 = \#111 + \#112 * \text{SIN} [\#113]$$



- (2) The part to be given priority in the operation sequence should be enclosed in brackets. Up to 5 pairs of such brackets including those for the functions may be used.

$$\#101 = \text{SQRT} [ [ [\#111 - \#112] * \text{SIN} [\#113] + \#114] * \#115] ;$$





### Examples of operation commands

(1) Main program and argument designation	G65 P100 A10 B20. ; #101=100.000 #102=200.000 ;	#1 10.000 #2 20.000 #101 100.000 #102 200.000
(2) Definition and substitution (=)	#1=1000 #2=1000. #3=#101 #4=#102 #5=#10001 (#10001=-10.)	#1 1000.000 #2 1000.000 #3 100.000 #4 200.000 #5 -10.000 ...
		} From common variables ... From offset amount
(3) Addition and subtraction (+, -)	#11=#1+1000 #12=#2-50. #13=#101+#1 #14=#10001-3. (#10001 = -10.) #15=#10001+#102	#11 2000.000 #12 950.000 #13 1100.000 #14 -13.000 #15 190.000
(4) Multiplication and division (*, /)	#21=100*100 #22=100.*100 #23=100*100 #24=100.*100. #25=100/100 #26=100./100 #27=100/100. #28=100./100. #29=#10001*#101 (#10001 = -10.) #30=#10001/#102	#21 10000.000 #22 10000.000 #23 10000.000 #24 10000.000 #25 1.000 #26 1.000 #27 1.000 #28 1.000 #29 -1000.000 #30 -0.050
(5) Remainder (MOD)	#19=48 #20=9 #31=#19 MOD #20	#19/#20 = 48/9 = 5 with 3 over #31 = 3
(6) Logical sum (OR)	#3=100 #4=#3 OR 14	#3 = 01100100 (binary) 14 = 00001110 (binary) #4 = 01101110 = 110
(7) Exclusive OR (XOR)	#3=100 #4=#3 XOR 14	#3 = 01100100 (binary) 14 = 00001110 (binary) #4 = 01101010 = 106
(8) Logical product (AND)	#9=100 #10=#9 AND 15	#9 = 01100100 (binary) 15 = 00001111 (binary) #10 = 00000100 = 4

## 13. Program Support Functions

### 13.9 User macro

(9) Sin (SIN)	#501 = SIN [60] #502 = SIN [60.] #503 = 1000*SIN [60] #504 = 1000*SIN [60.] #505 = 1000.*SIN [60] #506 = 1000.*SIN [60.] <b>(Note)</b> SIN [60] is equivalent to SIN [60.]	#501 0.866 #502 0.866 #503 866.025 #504 866.025 #505 866.025 #506 866.025
(10) Cosine (COS)	#541 = COS [45] #542 = COS [45.] #543 = 1000*COS [45] #544 = 1000*COS [45.] #545 = 1000.*COS [45] #546 = 1000.*COS [45.] <b>(Note)</b> COS [45] is equivalent to COS [45.]	#541 0.707 #542 0.707 #543 707.107 #544 707.107 #545 707.107 #546 707.107
(11) Tangent (TAN)	#551 = TAN [60] #552 = TAN [60.] #553 = 1000*TAN [60] #554 = 1000*TAN [60.] #555 = 1000.*TAN [60] #556 = 1000.*TAN [60.] <b>(Note)</b> TAN [60] is equivalent to TAN [60.]	#551 1.732 #552 1.732 #553 1732.051 #554 1732.051 #555 1732.051 #556 1732.051
(12) Arcsine ASIN	#531 = ASIN [100.500/201.] #532 = ASIN [100.500/201] #533 = ASIN [0.500] #534 = ASIN [-0.500]	#531 30.000 #432 30.000 #533 30.000 #534 -30.000 <b>(Note)</b> When #1273/bit 0 is set to 1, #534 will be 330°.
(13) Arctangent (ATAN or ATN)	#561 = ATAN [173205/100000] #562 = ATAN [173205/100000.] #563 = ATAN [173.205/100] #564 = ATAN [173.205/100.] #565 = ATAN [1.73205]	#561 60.000 #562 60.000 #563 60.000 #564 60.000 #565 60.000
(14) Arccosine (ACOS)	#521 = ACOS [100./141.421] #522 = ACOS [100./141.421]	#521 45.000 #522 45.000
(15) Square root (SQR or SQRT)	#571 = SQRT [1000] #572 = SQRT [1000.] #573 = SQRT [10. *10. +20. *20.] <b>(Note)</b> In order to increase the accuracy, proceed with the operation inside parentheses.	#571 31.623 #572 31.623 #573 22.360
(16) Absolute value (ABS)	#576 = -1000 #577 = ABS [#576] #3 = 70. #4 = -50. #580 = ABS [#4 - #3]	#576 -1000.000 #577 1000.000 #580 120.000
(17) BIN, BCD	#1 = 100 #11 = BIN [#1] #12 = BCD [#1]	#11 64 #12 256

(18)	Rounding off (ROUND or RND)	#21 = ROUND [14/3] #22 = ROUND [14./3] #23 = ROUND [14/3.] #24 = ROUND [14./3.] #25 = ROUND [-14/3] #26 = ROUND [-14./3] #27 = ROUND [-14/3.] #28 = ROUND [-14./3.]	#21 #22 #23 #24 #25 #26 #27 #28	5 5 5 5 -5 -5 -5 -5
(19)	Discarding fractions below decimal point (FIX)	#21 = FIX [14/3] #22 = FIX [14./3] #23 = FIX [14/3.] #24 = FIX [14./3.] #25 = FIX [-14/3] #26 = FIX [-14./3] #27 = FIX [-14/3.] #28 = FIX [-14./3.]	#21 #22 #23 #24 #25 #26 #27 #28	4.000 4.000 4.000 4.000 -4.000 -4.000 -4.000 -4.000
(20)	Adding fractions less than 1 (FUP)	#21 = FUP [14/3] #22 = FUP [14./3] #23 = FUP [14/3.] #24 = FUP [14./3.] #25 = FUP [-14/3] #26 = FUP [-14./3] #27 = FUP [-14/3.] #28 = FUP [-14./3.]	#21 #22 #23 #24 #25 #26 #27 #28	5.000 5.000 5.000 5.000 -5.000 -5.000 -5.000 -5.000
(21)	Natural logarithms (LN)	#101 = LN [5] #102 = LN [0.5] #103 = LN [-5]	#101 #102 Error	1.609 -0.693 "P282"
(22)	Exponents (EXP)	#104 = EXP [2] #105 = EXP [1] #106 = EXP [-2]	#104 #105 #106	7.389 2.718 0.135





### Operation accuracy

As shown in the following table errors will be generated when performing operations once and these errors will be accumulated by repeating the operations.

Operation format	Average error	Maximum error	Type of error
a = b + c a = b - c	$2.33 \times 10^{-10}$	$5.32 \times 10^{-10}$	Min. $ \frac{\epsilon}{b} ,  \frac{\epsilon}{c} $
a = b*c	$1.55 \times 10^{-10}$	$4.66 \times 10^{-10}$	Relative error $ \frac{\epsilon}{a} $
a = b/c	$4.66 \times 10^{-10}$	$1.86 \times 10^{-9}$	
a = b	$1.24 \times 10^{-9}$	$3.73 \times 10^{-9}$	
a = SIN [b] a = COS [b]	$5.0 \times 10^{-9}$	$1.0 \times 10^{-8}$	Absolute error $ \epsilon ^\circ$
a = ATAN [b/c]	$1.8 \times 10^{-6}$	$3.6 \times 10^{-6}$	

(Note) SIN/COS is calculated for the function TAN.



### Notes on reduced accuracy

#### (1) Addition and subtraction

It should be noted that when absolute values are used subtractively in addition or subtraction, the relative error cannot be kept below 10.

For instance, it is assumed that the real value produced as the operation calculation result of #10 and #20 are as follows (these value cannot be substituted directly):

#10 = 2345678988888.888

#20 = 2345678901234.567

Performing #10 - #20 will not produce #10 - #20 = 87654.321. There are 8 decimal digits in the variables and so the values of #10 and #20 will be as follows (strictly speaking, the internal values will differ somewhat from the value below because they are binary numbers):

#10 = 2345679000000.000

#20 = 2345678900000.000

Consequently, #10 - #20 = 100000.000 will generate a large error.

#### (2) Logical relation

EQ, NE, GT, LT, GE and LE are basically the same as addition and subtraction and so care should be taken with errors. For instance, to determine whether or not #10 and #20 are equal in the above example:

IF [#10 EQ #20]

It is not always possible to provide proper evaluation because of the above-mentioned error. Therefore when the error is evaluated as in the following expression:

IF [ABS [#10 - #20] LT 200000]

and the difference between #10 and #20 falls within the designated range error, both values should be considered equal.

#### (3) Trigonometric functions

Absolute errors are guaranteed with trigonometric functions but since the relative error is not under 10, care should be taken when dividing or multiplying after having used a trigonometric function.

### 13.9.7 Control commands

The flow of programs can be controlled by IF ~ GOTO ~ and WHILE ~ DO ~.



#### Branching

##### Format

**IF [conditional expression] GOTO n ; (where n = sequence number in the program)**

When the condition is satisfied, control branches to "n" and when it is not satisfied, the next block is executed.

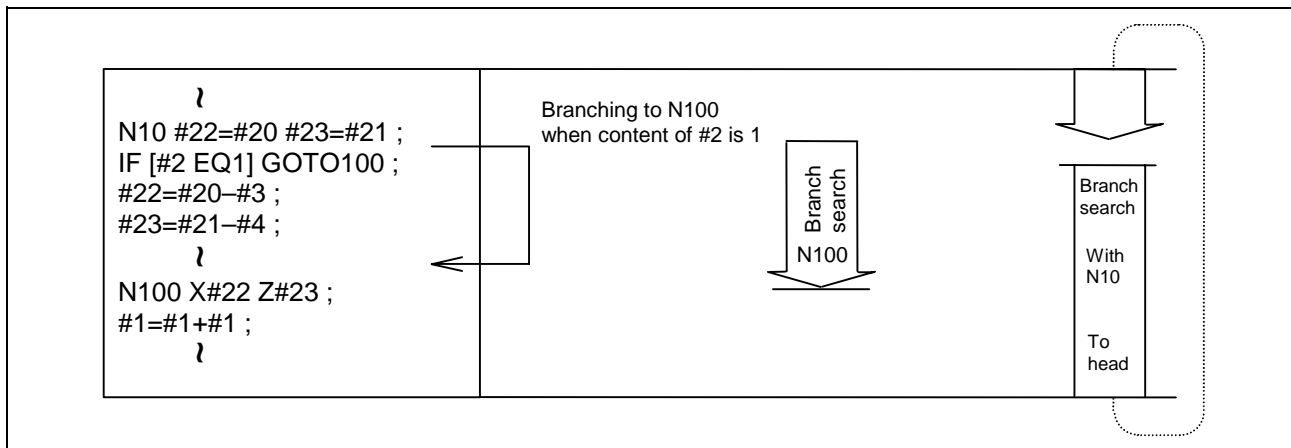
IF [conditional expression] can be omitted and, when it is, control passes to "n" unconditionally. The following types of [conditional expressions] are available.

#i EQ #j	= When #i and #j are equal
#i NE #j	≠ When #i and #j are not equal
#i GT #j	> When #i is greater than #j
#i LT #j	< When #i is less than #j
#i GE #j	≥ When #i is #j or more
#i LE #j	≤ When #i is #j or less

"n" of "GOTO n" must always be in the same program. Program error (P231) will result if it is not. A formula or variable can be used instead of #i, #j and "n".

In the block with sequence number "n" which will be executed after a "GOTO n" command, the sequence number Nn must always be at the head of the block. Otherwise, program error (P231) will result.

If "/" is at the head of the block and Nn follows, control can be branched to the sequence number.



**(Note 1)** When the sequence number of the branch destination is searched, the search is conducted up to the end of the program (% code) from the block following "IF.....;" and if it is not found, it is conducted up to the block before "IF .....;". Therefore, branch searches in the opposite direction to the program flow will take longer to execute compared with branch searches in the forward direction.

**(Note 2)** Only compare EQ and NE by integers. Use GE, GT, LE and LT for comparison when there are numbers below the decimal point.



### Repetition

#### Format

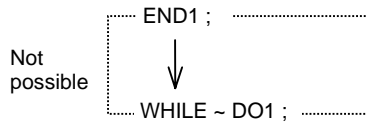
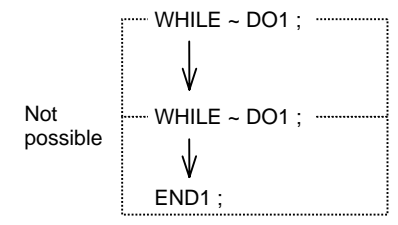
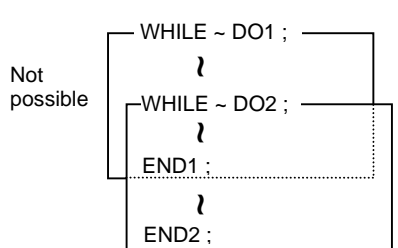
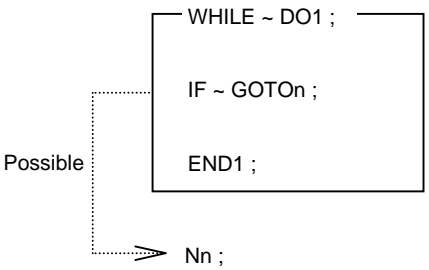
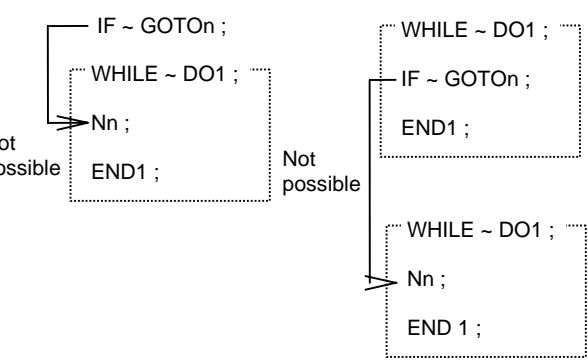
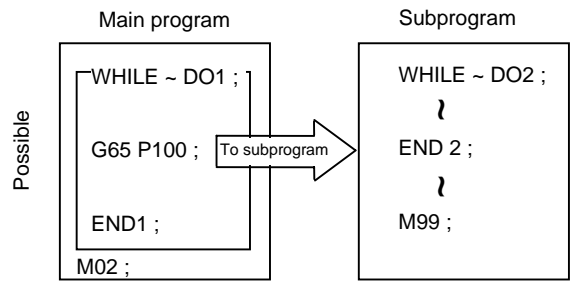
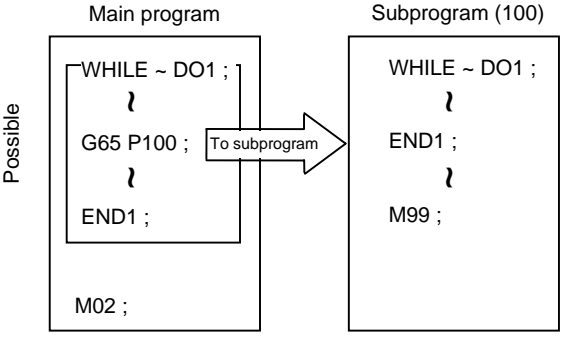
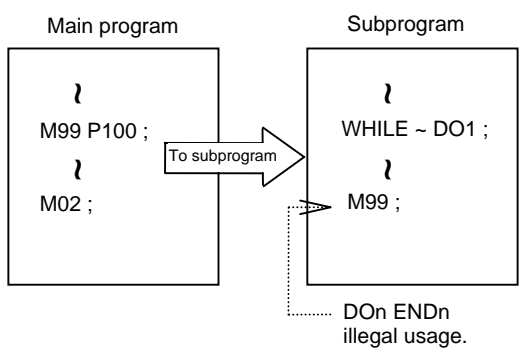
```
WHILE [conditional expression] DOm ; (m =1, 2, 3 ..... 127)
}
ENDm ;
```

While the conditional expression is established, the blocks from the following block to ENDm are repeatedly executed; when it is not established, execution moves to the block after ENDm. DOM may come before WHILE.

"WHILE [conditional expression] DOm" and "ENDm" must be used as a pair. If "WHILE [conditional expression]" is omitted, these blocks will be repeatedly ad infinitum. The repeating identification numbers range from 1 to 127 (DO1, DO2, DO3, ..... DO127). Up to 27 nesting levels can be used.

<p>(1) Same identification number can be used any number of times.</p> <div style="margin-left: 20px;"> <p>Possible</p> <pre>WHILE ~ DO1 ; } END1 ;</pre> <p>Possible</p> <pre>WHILE ~ DO1 ; } END1 ;</pre> </div>	<p>(2) Any number may be used for the WHILE~ DOM identification number.</p> <div style="margin-left: 20px;"> <pre>WHILE ~ DO1 ; } END1 ;</pre> <p>}</p> <pre>WHILE ~ DO3 ; } END3 ;</pre> <p>Possible</p> <p>}</p> <pre>WHILE ~ DO2 ; } END2 ;</pre> <p>}</p> <pre>WHILE ~ DO1 ; } END1 ;</pre> </div>
<p>(3) Up to 27 nesting levels for WHILE~DOM. "m" is any number form 1 to 127 for the nesting depth.</p> <div style="margin-left: 20px;"> <p>Possible</p> </div> <p><b>(Note)</b> With nesting, "m" which has been used once cannot be used.</p>	<p>(4) The number of WHILE~DOM nesting levels cannot exceed 27.</p> <div style="margin-left: 20px;"> </div>

**(Note)** Even if a fixed cycle containing WHILE is called, the nesting level will be counted up.

<p>(5) WHILE ~ DOn must be designated first and ENdn last.</p> <p>Not possible</p> 	<p>(6) WHILE ~ DOn and ENdn must correspond on a 1:1 (pairing) basis in the same program.</p> <p>Not possible</p> 
<p>(7) Two WHILE ~ DOn's must not overlap.</p> <p>Not possible</p> 	<p>(8) Branching externally is possible from the WHILE ~ DOn range.</p> <p>Possible</p> 
<p>(9) No branching is possible inside WHILE ~ DOn.</p> <p>Not possible</p> 	<p>(10) Subprograms can be called by M98, G65 or G66 between WHILE ~ DOn's.</p> <p>Possible</p> 
<p>(11) Calls can be initiated by G65 or G66 between WHILE ~ DOn's and commands can be issued again from 1. Up to 27 nesting levels are possible for the main program and subprograms.</p> <p>Possible</p> 	<p>(12) A program error results with M99 unless WHILE and END are paired within the subprograms (including macro programs).</p> <p>DOn ENdn illegal usage.</p> 

## 13.9.8 External output commands



## Function and purpose

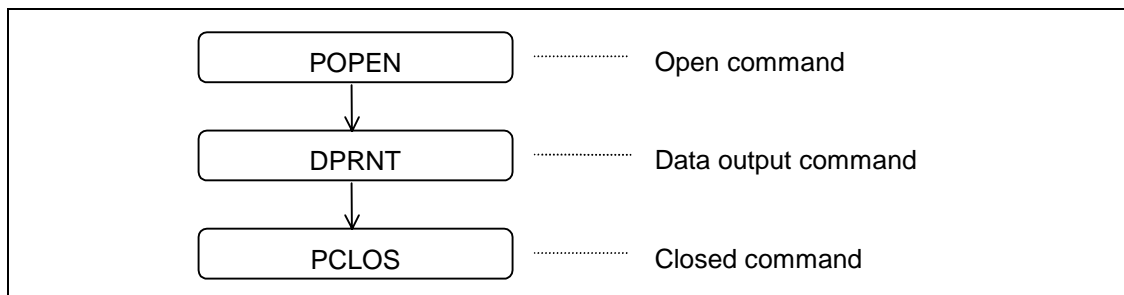
Besides the standard user macro commands, the following macro instructions are also available as external output commands. They are designed to output the variable values or characters via the RS-232C interface.



## Command format

**POPEN** For preparing the processing of data outputs  
**PCLOS** For terminating the processing of data outputs  
**DPRNT** For character output and digit-by-digit variable numerical output

## Command sequence



## Open command : POPEN

- (1) The command is issued before the series of data output commands.
- (2) The DC2 control code and % code are output from the NC system to the external output device.
- (3) Once POPEN ; has been issued, it will remain valid until PCLOS ; is issued.



## Close command : PCLOS

- (1) This command is issued when all the data outputs are completed.
- (2) The DC4 control code and % code are output from the NC unit to the external output device.
- (3) This command is used together with the open command and it should not be issued unless the open mode has been established.
- (4) Issue the close command at the end of the program even when operation has been suspended by resetting or some other operation during data output.



### Data output command : DPRNT

<b>DPRNT [ /1 # v1 [ d1 c1 ] /2 # v2 [ d2 c2 ] ..... ]</b>		
/1	Character string	
v1	Variable number	
d1	Significant digits above decimal point	} C + d ≤ 8
c1	Significant digits below decimal point	

- (1) The character output and decimal output of the variable values are done with ISO codes.
- (2) The commanded character string is output as is by the ISO code. Alphanumerics (A ~ Z, 0 ~ 9) and special characters (+, -, \*, /) can be used.
- (3) The required significant digits above and below the decimal point in the variable values are commanded within square parentheses. As a result, the variable values equivalent to the commanded number of digits including the decimal point are output in ISO code in decimal notation from the high-order digits. Trailing zeroes are not omitted.
- (4) Leading zeroes are omitted.  
The leading zeroes can also be replaced by blank if so specified with a parameter. This can justify printed data on the last column.

## 13.9.9 Precautions

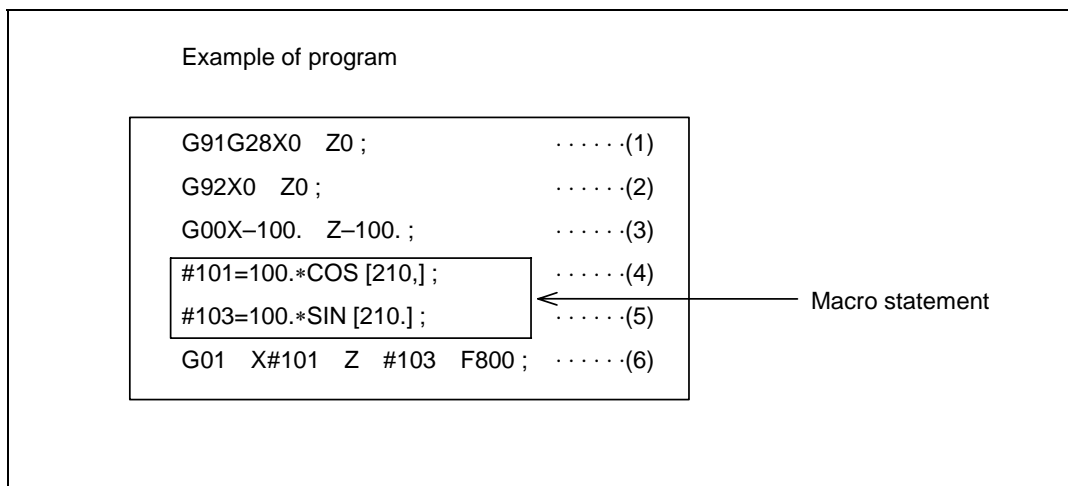


## Precautions

- (1) When the user macro commands are employed, it is possible to use the M, S, T and other control commands together with the operation, decision, branching and other macro commands for creating the machining programs. When the former commands are made into executable statements and the latter commands into macro statements, the macro statement processing should be accomplished as quickly as possible in order to minimize the machining time, because such processing is not directly related to machine control.

As a result, the parameter (#8101 MACRO SINGLE on the parameter screen) can be decided upon and the macro statements can be processed in parallel with the execution of the executable statement.

(The parameter can be set OFF during normal machining to process all the macro statements together or set ON during a program check to execute the macro statements block by block. This enables the setting to be made in accordance with the intended objective in mind.)

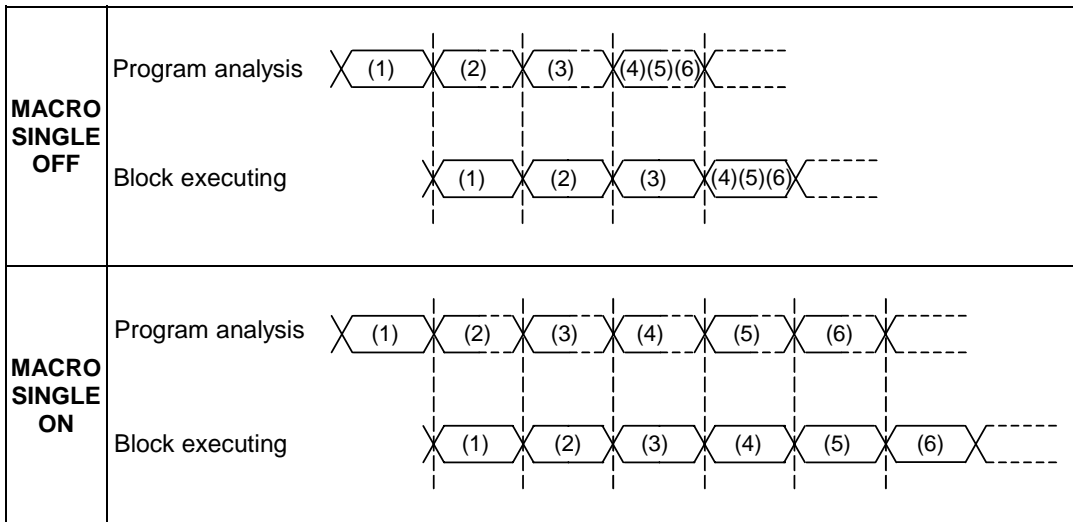


Macro statements are:

- (a) Operation commands (blocks including "=")
- (b) Control commands (blocks including GOTO, DO~END, etc.)
- (c) Macro call commands (including macro calls based on G codes and cancel command (G65. G66. G66.1, G67))

NC statements are all those statements which are not executable statements.

### Flow of processing



### Machining program display

<b>MACRO SINGLE OFF</b>	<pre>[Executing] N3 G00 X-100. Z-100. ; [Next command] N6 G01 X#101 Z#103 F800 ;</pre>	<p>N4, N5 and N6 are processed in parallel with the control of the executable statement of N3. N6 is an executable statement and so it is displayed as the next command. If the N4, N5 and N6 analysis is in time during N3 control, the machine movement will be continuously controlled.</p>
<b>MACRO SINGLE ON</b>	<pre>[Executing] N3 G00 X-100. Z-100. ; [Next command] N4 #101=100.*COS[210.] ;</pre>	<p>N4 is processed in parallel with the control of the executable statement of N3, and it is displayed as the next command. N5 and N6 are analyzed and N6 is executed after N3 has finished, and so the machine control is held on standby during the N5 and N6 analysis time.</p>



## 13. Program Support Functions

### 13.10 Corner chamfering/corner rounding I

#### 13.10 Corner chamfering/corner rounding I

Chamfering at any angle or corner rounding is performed automatically by adding ",C\_" or ",R\_" to the end of the block to be commanded first among those command blocks which shape the corner with lines only.

By setting the parameters, "I\_", "K\_", "C\_" can be set for the chamfer instead of ",C\_", and "R\_" can be set for rounding instead of ",R\_".

##### 13.10.1 Corner chamfering ",C" (or "I\_", "K\_", "C\_")



#### Function and purpose

The corner is chamfered in such a way that the positions produced by subtracting the lengths commanded by ",C\_" (or "I\_", "K\_", "C\_") from the hypothetical starting and final corners which would apply if no chamfering were to be performed, are connected.



#### Command format

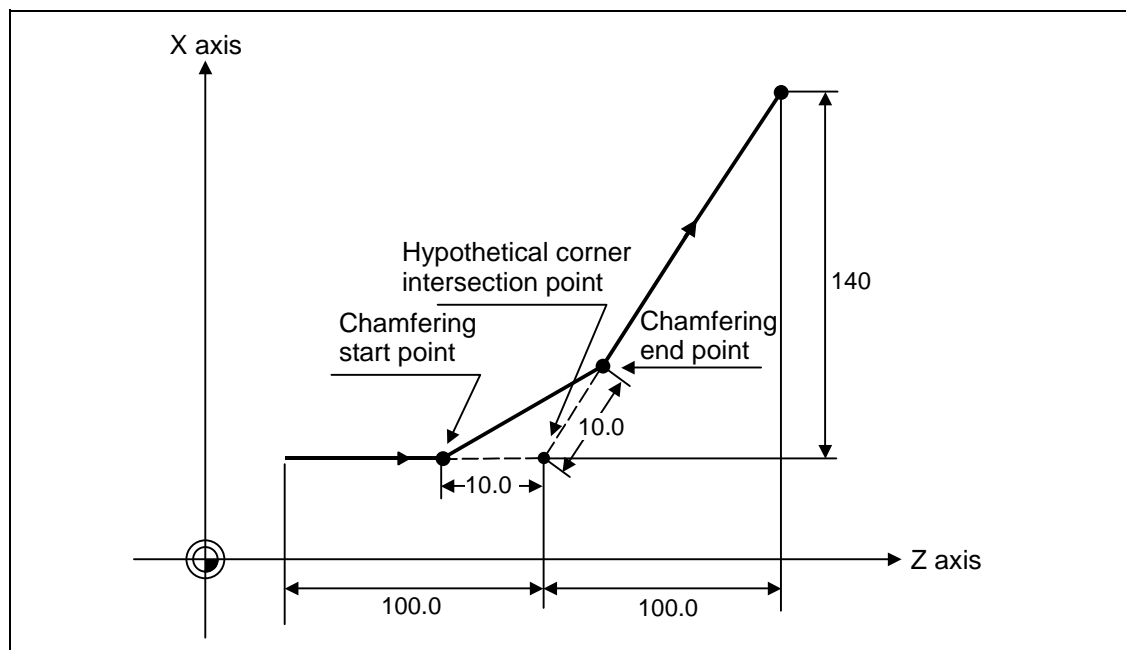
```
N100 G01 X__ Z__,C_ (or I_/K_/C_);  
N200 G01 X__ Z__;  
,C_/I_/K_/C_ Length up to chamfering starting point or end point from hypothetical corner
```

Chamfering is performed at the point where N100 and N200 intersect.



#### Example of program

```
G01 W100. ,C10. F100 ;  
U280. W100. ;
```



## 13. Program Support Functions

### 13.10 Corner chamfering/corner rounding I



#### Detailed description

- (1) The start point of the block following the corner chamfering serves as the hypothetical corner intersection point.
- (2) If the base specification parameter "#1272 ext08/bit6" is "0", the ",C" command will be interpreted as a C command if there is no "," (comma).
- (3) If there are multiple or duplicate corner chamfering commands in the same block, the command issued last will be valid.
- (4) When both corner chamfering and corner rounding are commanded in the same block, the latter command is valid.
- (5) Tool offset is calculated for the shape which has already been subjected to corner chamfering.
- (6) When the block following a command with corner chamfering does not contain a linear command, a corner chamfering/corner rounding II function results.
- (7) Program error (P383) results when the movement amount in the block commanding corner chamfering is less than the chamfering amount.
- (8) Program error (P384) results when the movement amount in the block following the block commanding corner chamfering is less than the chamfering amount.
- (9) A program error (P382) results when a movement command is not issued in the block following the corner chamfering I command.
- (10) If "C" is used as the axis name or 2nd miscellaneous function, corner chamfering cannot be commanded with "C".
- (11) Corner chamfering cannot be commanded with "I" or "K" in an arc command block. "I" and "K" are the arc center commands.



#### Precautions

- (1) Corner chamfering and corner rounding can be commanded with "I", "K", "R" only when the 1st block of the corner chamfering/corner rounding command is linear.
- (2) Corner chamfering with "I", "K", and corner rounding with "R" can be commanded when the 1st block of the corner chamfering/corner rounding command is linear and the 2nd block is an arc command. "I", "K" in the 2nd block are the arc center commands.  

```
N100 G01 Xx Zz li ;..... li corner chamfering length  
N200 G02 Xx Zz li Kk ; ..... li, Kk arc center command
```
- (3) If ",C\_", ",R\_" or "I\_", "K\_", "C\_", "R\_" are commanded in the same block, ",C\_", ",R\_" will have the priority.

## 13. Program Support Functions

### 13.10 Corner chamfering/corner rounding I

#### 13.10.2 Corner rounding ",R\_" (or "R\_")



##### Function and purpose

Corner rounding of the hypothetical starting and final corners when corner is assumed as not rounded is performed at the radius of the arc commanded in ",R\_" (or "R\_")



##### Command format

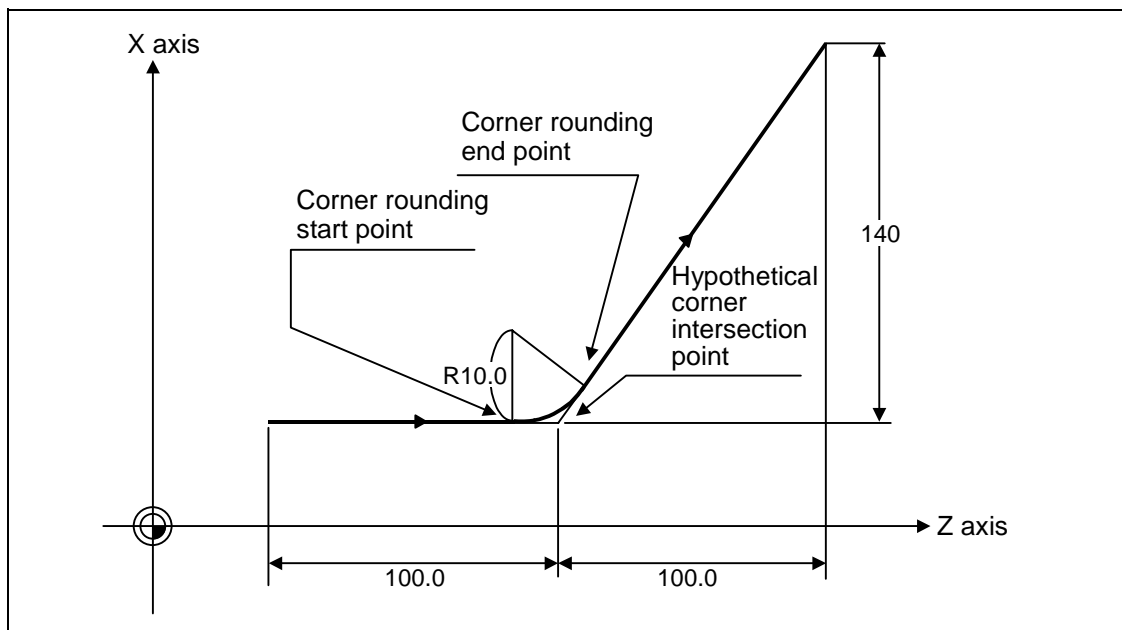
```
N100 G01 X__ Z__ ,R__ (or R_) ;  
N200 G01 X__ Z__ ;  
 ,R_ / R_      Arc radius of corner rounding
```

Corner rounding is performed at the point where N100 and N200 intersect.



##### Example of program

```
G01 W100. ,R10. F100 ;  
U280. W100. ;
```



## 13. Program Support Functions

### 13.10 Corner chamfering/corner rounding I



#### Detailed description

- (1) The start point of the block following the corner rounding serves as the hypothetical corner intersection point.
- (2) If the base specification parameter "#1272 ext08/bit6" is "0", the ",R" command will be interpreted as a R command if there is no "," (comma).
- (3) When both corner chamfering and corner rounding are commanded in the same block the latter command is valid.
- (4) Tool offset is calculated for the shape which has already been subjected to corner rounding.
- (5) When the block following a command with corner rounding does not contain a linear command, a corner chamfering/corner rounding II function results.
- (6) Program error (P383) results when the movement amount in the block commanding corner rounding is less than the R value.
- (7) Program error (P384) results when the movement amount in the block following corner rounding is less than the R value.
- (8) A program error (P382) will occur if a movement command is not issued in the block following corner rounding.
- (9) Corner rounding cannot be commanded with "R" in an arc command block. "R" is the arc radius command.



#### Precautions

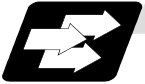
- (1) Corner chamfering and corner rounding can be commanded with "I", "K", "R" only when the 1st block of the corner chamfering/corner rounding command is linear.
- (2) Corner chamfering with "I", "K", and corner rounding with "R" can be commanded when the 1st block of the corner chamfering/corner rounding command is linear and the 2nd block is an arc command. "I", "K" in the 2nd block are the arc center commands.  

```
N100 G01 Xx Zz li ;..... li corner chamfering length  
N200 G02 Xx Zz li Kk ; ..... li, Kk arc center command
```
- (3) If ",C\_", ",R\_" or "I\_", "K\_", "C\_", "R\_" are commanded in the same block, ",C\_", ",R\_" will have the priority.

## 13. Program Support Functions

### 13.10 Corner chamfering/corner rounding I

#### 13.10.3 Corner chamfering/corner rounding expansion

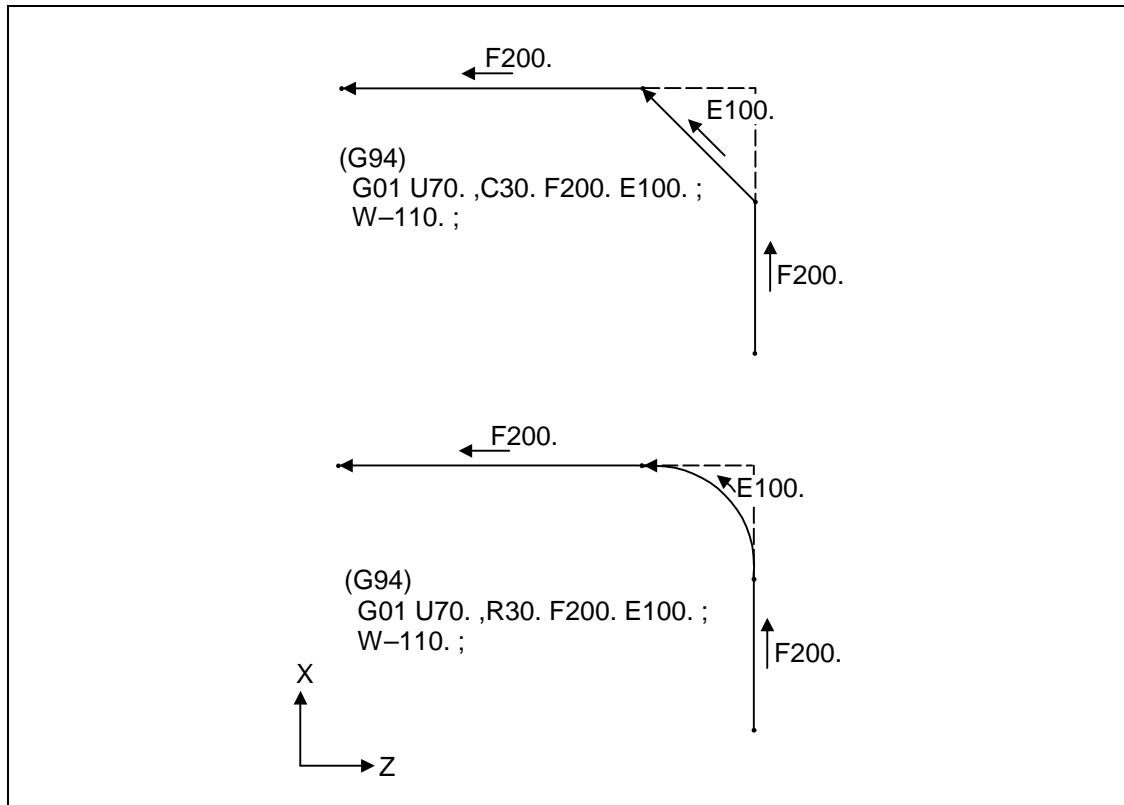


#### Function and purpose

Using an E command, the feedrate can be designated for the corner chamfering and corner rounding section.

In this way the corner section can be cut to the correct shape.

#### Example



# 13. Program Support Functions

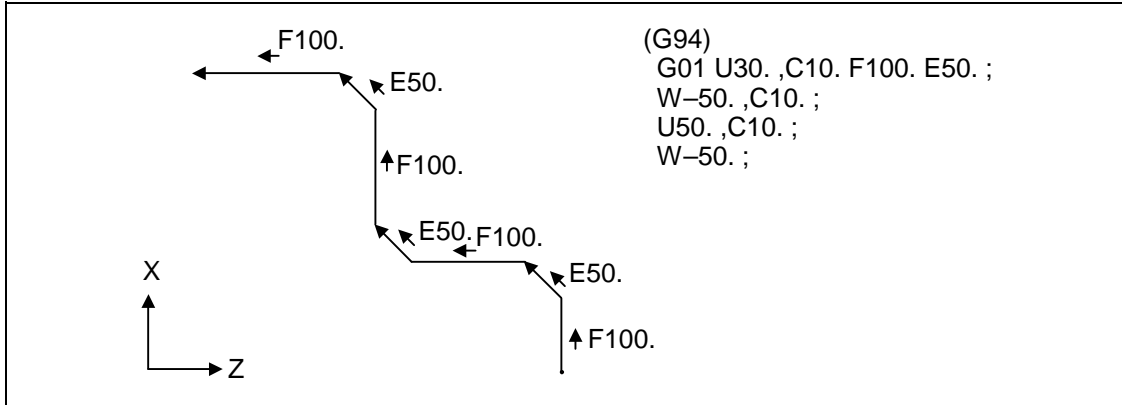
## 13.10 Corner chamfering/corner rounding I



### Detailed description

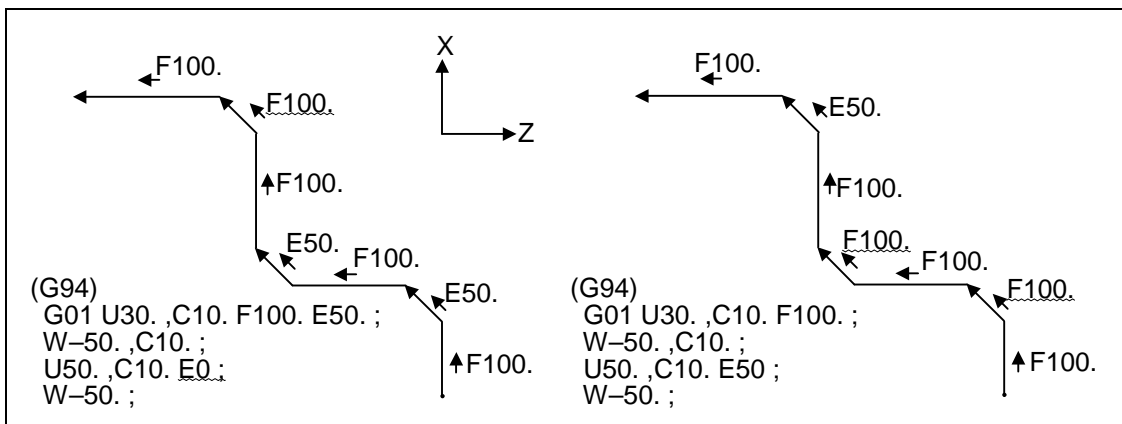
- (1) The E command is modal. It is also valid for the feed in the next corner chamfering/corner rounding section.

#### Example



- (2) E command modal has separate asynchronous feedrate modal and synchronous feedrate modal functions. Which one is validated depends on the asynchronous/synchronous mode (G94/G95).
- (3) When the E command is 0, or there has not been an E command up to now, the corner chamfering/corner rounding section feedrate is the same as the F command feedrate.

#### Example



- (4) E command modal is not cleared when the reset button is pressed. It is cleared when the power is turned OFF. (In the same manner as F commands.)
- (5) All E commands except those shown below are at the corner chamfering/corner rounding section feedrate.
  - E commands during thread cutting modal
  - E commands during thread cutting cycle modal

## 13. Program Support Functions

### 13.10 Corner chamfering/corner rounding I

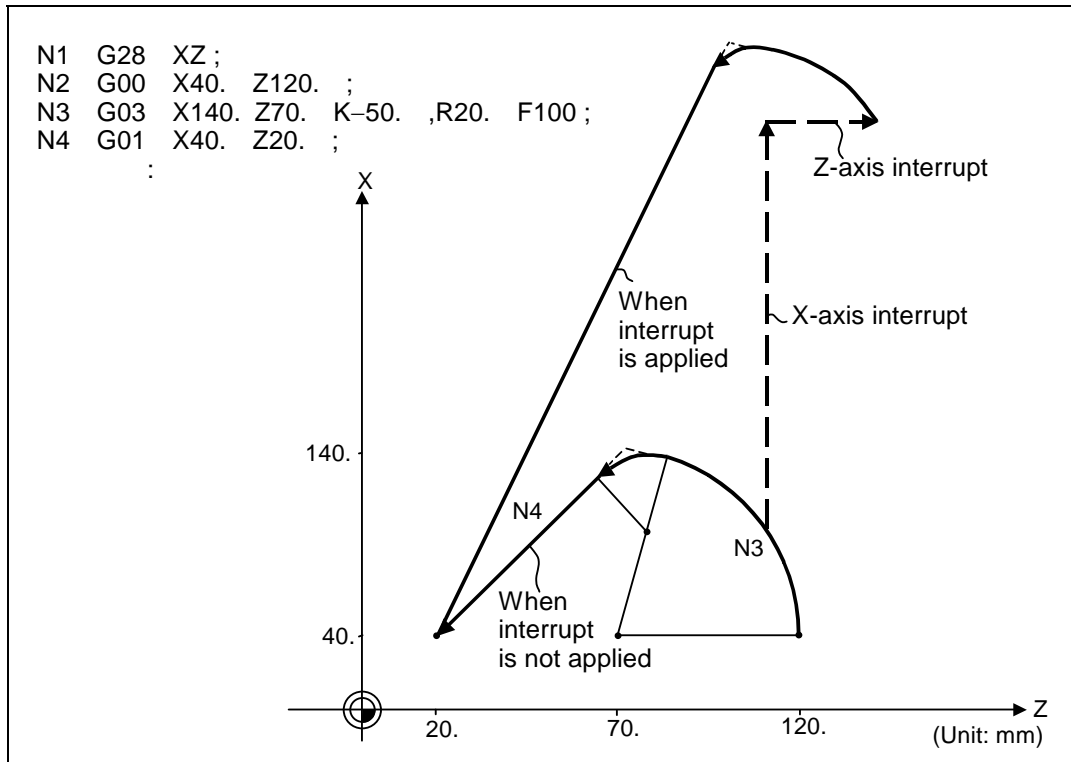
#### 13.10.4 Interrupt during corner chamfering/corner rounding



##### Detailed description

- (1) The operations are shown below for manual interrupt during corner chamfering or corner rounding.

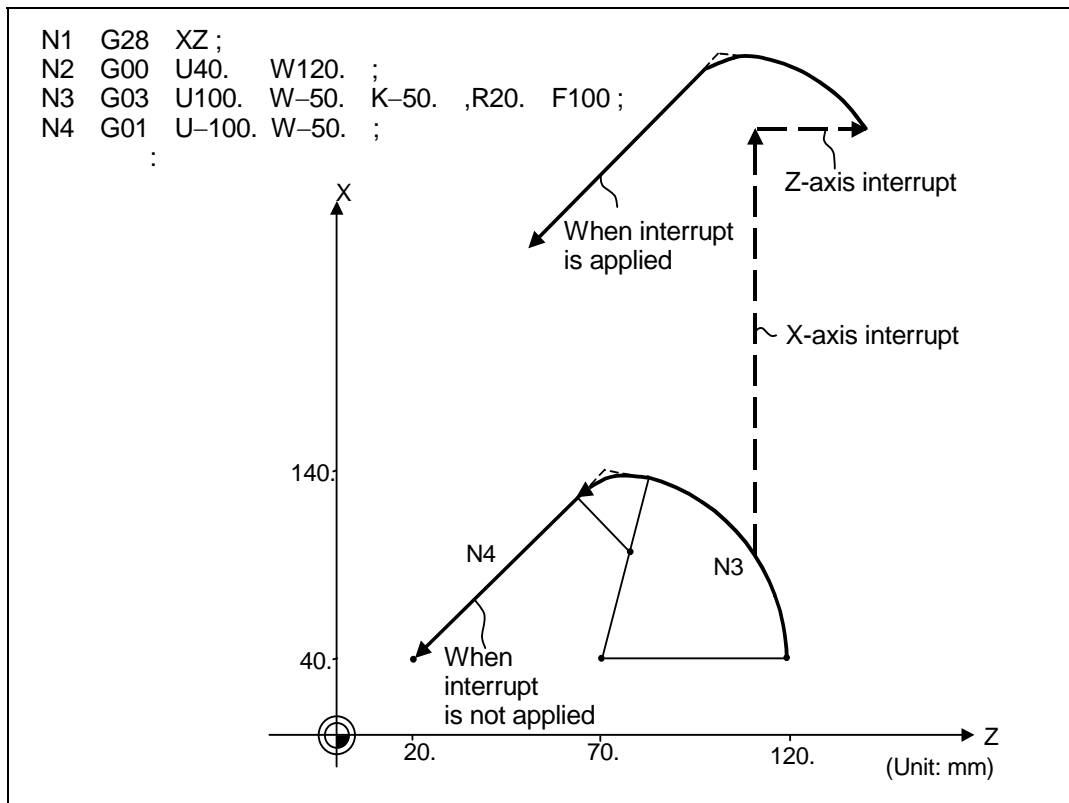
**With an absolute value command and manual absolute switch ON.**



## 13. Program Support Functions

### 13.10 Corner chamfering/corner rounding I

With an incremental value command and manual absolute switch OFF



- (2) With a single block during corner chamfering or corner rounding, the tool stops after corner chamfering or corner rounding is executed.



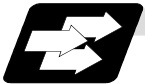
## 13. Program Support Functions

### 13.11 Corner chamfering/corner rounding II

#### 13.11 Corner chamfering/corner rounding II

In a block that forms a corner with continuous random angle lines or arcs, corner chamfering and corner rounding can be performed by adding ",C" or ",R" to the end of the block commanded first. By setting the parameters, "I\_", "K\_", "C\_" can be set for the chamfering instead of ",C\_", and "R\_" can be set for rounding instead of ",R\_". Corner chamfering and corner rounding can both be commanded as an absolute value or incremental value.

##### 13.11.1 Corner chamfering ",C\_" (or "I\_", "K\_", "C\_")



###### Function and purpose

The corner is chamfered by commanding ",C" (or "I\_", "K\_", "C\_") in the 1st block of two blocks having continuous arcs. For an arc, this will be the chord length.



###### Command format

<b>N100 G03 X__ Z__ K__ ,C__ ; (or C_);</b>
<b>N200 G01 X__ Z__ ;</b>
,C_ / C_                      Length up to chamfering starting point or end point from hypothetical corner

Corner chamfering is performed at the point where N100 and N200 intersect.



###### Example of program

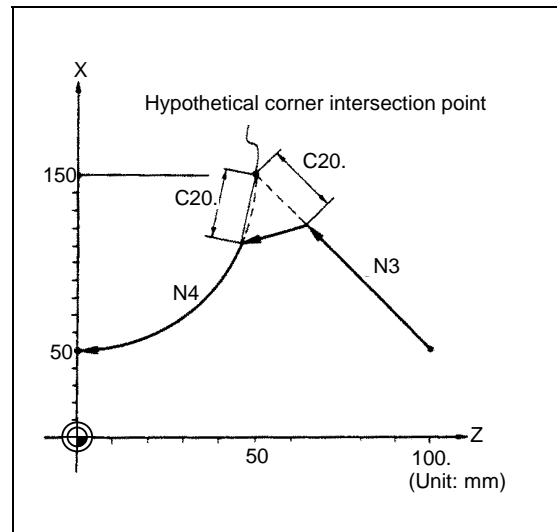
##### (1) Linear – arc

###### Absolute value command

N1 G28 X Z ;
N2 G00 X50. Z100. ;
N3 G01 X150. Z50. ,C20. F100 ;
N4 G02 X50. Z0 I0 K-50. ;
:

###### Relative value command

N1 G28 X Z ;
N2 G00 U25. W100. ;
N3 G01 U50. W-50. ,C20. F100 ;
N4 G02 U-50. W-50. I0 K-50. ;
:



## 13. Program Support Functions

### 13.11 Corner chamfering/corner rounding II

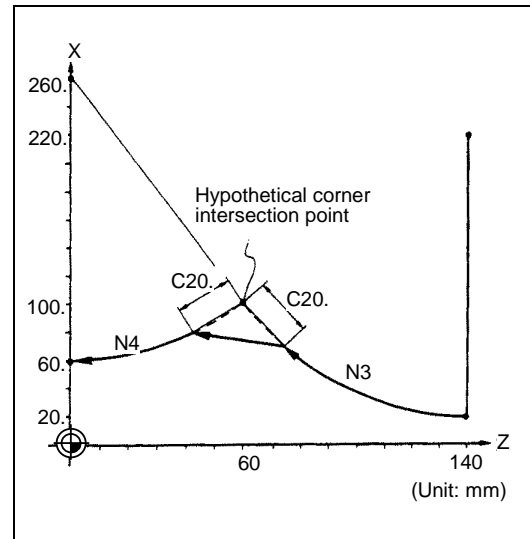
#### (2) Arc – arc

Absolute value command

N1 G28 X Z;
N2 G00 X20. Z140. ;
N3 G02 X100. Z60 I100 K0. ,C20. F100 ;
N4 X60. Z0 I80. K-60. ;

Relative value command

N1 G28 X Z ;
N2 G00 U10. W140. ;
N3 G02 U40. W-80. R100. ,C20. F100 ;
N4 U-20. W-60. I80. K-60. ;



#### Detailed description

- (1) The start point of the block following the corner chamfering serves as the hypothetical corner intersection point.
- (2) If the base specification parameters "#1272 ext08/bit6" is "0", the ",C" command will be interpreted as a C command if there is no "," (comma).
- (3) If there are multiple or duplicate corner chamfering commands in the same block, the command issued last will be valid.
- (4) When both corner chamfering and corner rounding are commanded in the same block, the latter command is valid.
- (5) Tool offset is calculated for the shape which has already been subjected to corner chamfering.
- (6) A program error (P385) results when positioning or thread cutting is commanded in the corner chamfering command block or next block.
- (7) A program error (P382) results when the block following corner chamfering contains a G command other than group 01 or another command.
- (8) Program error (P383) results when the movement amount in the block commanding corner chamfering is less than the chamfering amount.
- (9) Program error (P384) results when the movement amount in the block following the block commanding corner chamfering is less than the chamfering amount.
- (10) Even if a diameter is commanded, it will be handled as a radius value during corner chamfering.
- (11) A program error (P382) results when a movement command is not issued in the block following the corner chamfering II command.
- (12) If "C" is used as the axis name or 2nd miscellaneous function, corner chamfering cannot be commanded with "C".
- (13) Corner chamfering cannot be commanded with "I" or "K" in an arc command block. "I" and "K" are the arc center commands.

## 13. Program Support Functions

### 13.11 Corner chamfering/corner rounding II



#### Precautions

- (1) Corner chamfering and corner rounding can be commanded with "I", "K", "R" only when the 1st block of the corner chamfering/corner rounding command is linear.
- (2) Corner chamfering with "I", "K", and corner rounding with "R" can be commanded when the 1st block of the corner chamfering/corner rounding command is linear and the 2nd block is an arc command. "I", "K" in the 2nd block are the arc center commands.  
N100 G01 Xx Zz li ;..... li corner chamfering length  
N200 G02 Xx Zz li Kk ; ..... li, Kk arc center command
- (3) If ",C\_", ",R\_" or "I\_", "K\_", "C\_", "R\_" are commanded in the same block, ",C\_", ",R\_" will have the priority.

# 13. Program Support Functions

## 13.11 Corner chamfering/corner rounding II

### 13.11.2 Corner rounding ",R\_" (or "R\_")



#### Function and purpose

The corner is rounded by commanding ",R\_" (or "R\_") in the 1st block of two blocks containing continuous arcs.



#### Command format

```
N100 G03 X__ Z__ I__ K__ ,R__ (or R_);
N200 G01 X__ Z__ ;
,R_ / R_      Arc radius of corner rounding
```

Corner rounding is performed at the point where N100 and N200 intersect.



#### Example of program

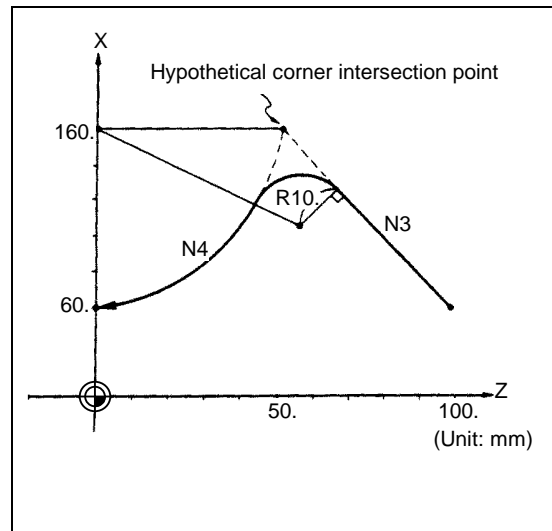
##### (1) Linear – arc

Absolute value command

```
N1 G28 X Z;
N2 G00 X60. Z100. ;
N3 G01 X160. Z50. ,R10. F100 ;
N4 G02 X60. Z0 I0 K-50. ;
:
```

Relative value command

```
N1 G28 X Z;
N2 G00 U30. W100. ;
N3 G01 U50. W-50. ,R10. F100 ;
N4 G02 U-50. W-50. I0 K-50. ;
:
```



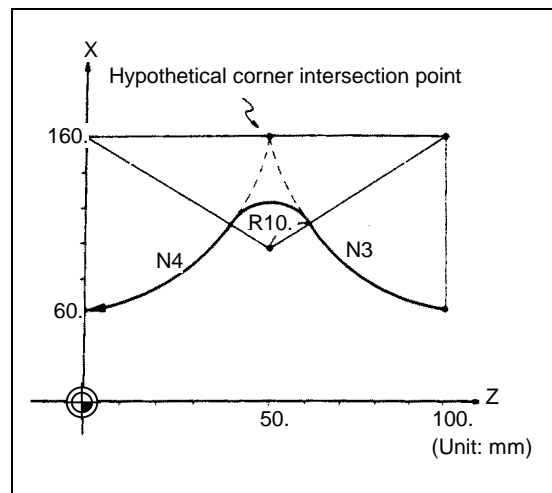
##### (2) Arc – arc

Absolute value command

```
N1 G28 X Z;
N2 G00 X60. Z100. ;
N3 G02 X160. Z50. R60 ,R10. F100 ;
N4 X60. Z0 R50. ;
:
```

Relative value command

```
N1 G28 X Z;
N2 G00 U30. W100. ;
N3 G02 U50. W-50. I50. K0 ,R10. F100 ;
N4 U-50. W-50. I0. K-50. ;
:
```



## 13. Program Support Functions

### 13.11 Corner chamfering/corner rounding II



#### Detailed description

- (1) The start point of the block following the corner rounding serves as the hypothetical corner intersection point.
- (2) If the base specification parameters "#1272 ext08/bit6" is "0", the ",R" command will be interpreted as a R command if there is no "," (comma).
- (3) When both corner chamfering and corner rounding are commanded in the same block the latter command is valid.
- (4) Tool offset is calculated for the shape which has already been subjected to corner rounding.
- (5) A program error (P385) results when positioning or thread cutting is commanded in the corner rounding command block or next block.
- (6) A program error (P382) results when the block following corner rounding contains a G command other than group 01 or another command.
- (7) Program error (P383) results when the movement amount in the block commanding corner rounding is less than the R value.
- (8) Program error (P384) results when the movement amount in the block following the block commanding corner rounding is less than the R value.
- (9) Even if a diameter is commanded, it will be handled as a radius value during corner rounding.
- (10) A program error (P382) results when a movement command is not issued in the block following corner rounding II.
- (11) Corner rounding cannot be commanded with "R" in an arc command block. "R" is the arc radius command.



#### Precautions

- (1) Corner chamfering and corner rounding can be commanded with "I", "K", "R" only when the 1st block of the corner chamfering/corner rounding command is linear.
- (2) Corner chamfering with "I", "K", and corner rounding with "R" can be commanded when the 1st block of the corner chamfering/corner rounding command is linear and the 2nd block is an arc command. "I", "K" in the 2nd block are the arc center commands.  

```
N100 G01 Xx Zz li ;..... li corner chamfering length  
N200 G02 Xx Zz li Kk ; ..... li, Kk arc center command
```
- (3) If ",C\_", ",R\_" or "I\_", "K\_", "C\_", "R\_" are commanded in the same block, ",C\_", ",R\_" will have the priority.

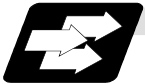
#### 13.11.3 Corner chamfering/corner rounding expansion

Refer to section "13.10.3 Corner chamfering/corner rounding expansion" for details.

#### 13.11.4 Interrupt during corner chamfering/corner rounding

Refer to section "13.10.4 Interrupt during corner chamfering/corner rounding" for details.

## 13.12 Linear Angle Command



## Function and purpose

The end point coordinates are calculated automatically by commanding the linear angle and one of the end point coordinate axes.

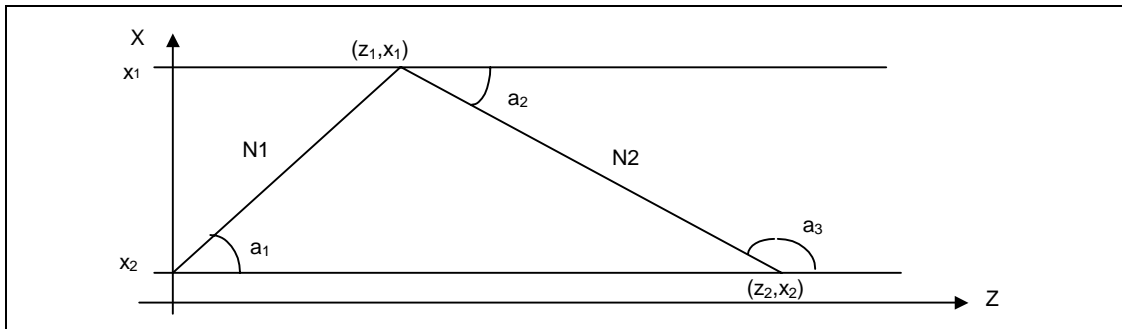


## Command format

```
N1 G01 Xx1 (Zz1) Aa1;
N1 G01 Xx2 (Zz2) A-a2; (A-a2 can also be set as As3.)
or
N1 G01 Xx1 (Zz1) ,Aa1;
N1 G01 Xx2 (Zz2) ,A-a2;
```

This designates the angle and the X or Z axis coordinates.

Select the command plane with G17 to G19.



## Detailed description

- (1) The angle is from the + direction of the horizontal axis on the selected plane. The counter-clockwise (CCW) direction is considered to be + and the clockwise direction (CW) -.
- (2) Either of the axes on the selected plane is commanded for the end point.
- (3) The angle is ignored when the angle and the coordinates of both axes are commanded.
- (4) When only the angle has been commanded, this is treated as a geometric command.
- (5) The angle of either the start point ( $a_1$ ) or end point ( $a_2$ ) may be used.
- (6) This function is valid only for the G01 command; it is not valid for other interpolation or positioning commands.
- (7) The range of slope "a" is  $-360.000 \leq a \leq 360.000$ .  
When a value outside this range is commanded, it will be divided by 360 (degrees) and the remainder will be commanded.  
**(Example)** If 400 is commanded,  $40^\circ$  (remainder of  $400/360$ ) will become the command angle.
- (8) If address A is used for the axis name or 2nd miscellaneous function, use ",A" as the angle.
- (9) If "A" and ",A" are commanded in the same block, ",A" will be interpreted as the angle.

### 13.13 Geometric

#### 13.13.1 Geometric I



#### Function and purpose

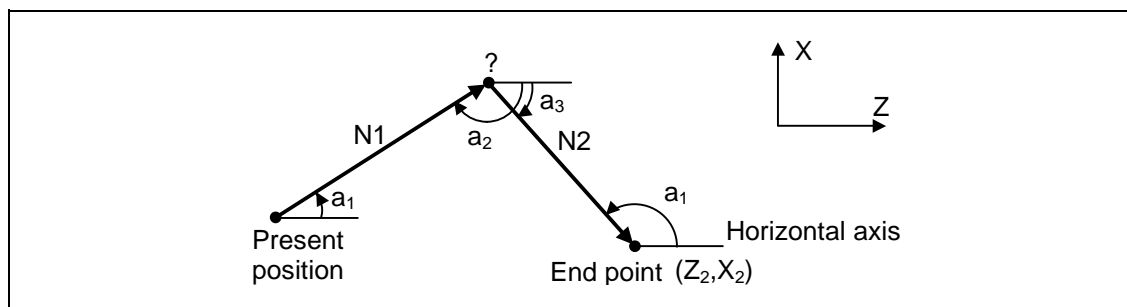
When it is difficult to determine the point at which the two straight lines intersect with a continuous linear interpolation command, the end point of the first straight line will be automatically calculated inside the CNC and the movement command will be controlled provided that the slope of the first straight line as well as the end point coordinates and slope of the second straight line are commanded.

**(Note)** If the parameter (#1082 Geomet) is set to 0, geometric I will not function.



#### Command format

<b>N1 G01 Aa<sub>1</sub> (A-a<sub>2</sub>) Ff<sub>1</sub> ;</b>	<b>This designates the angle and feedrate.</b>
<b>N2 Xx<sub>2</sub> Zz<sub>2</sub> A-a<sub>2</sub> (Aa<sub>3</sub>) Ff<sub>2</sub> ;</b>	<b>This designates the next block end point coordinates, angle and feedrate.</b>
Aa <sub>1</sub> , A-a <sub>2</sub> , Aa <sub>3</sub>	Angle
Ff <sub>1</sub> , Ff <sub>2</sub>	Feedrate
Xx <sub>2</sub> , Zz <sub>2</sub>	Next block end point coordinates





### Detailed description

- (1) Program error (P396) results when the geometric command is not on the selected plane.
- (2) The slope is expressed as the angle that is formed with the horizontal axis + direction on the selected plane. The counterclockwise (CCW) direction is considered to be + and the clockwise direction (CW) –.
- (3) Slope "a" ranges from  $-360.000 \leq a \leq 360.000$ .  
When a value outside this range is commanded, it will be divided by 360 (degrees) and the remainder will be commanded.  
**(Example)** If 400. is commanded,  $40^\circ$  (remainder of  $400/360$ ) will become the command angle.
- (4) The slope of the line can be commanded on the start or end point side. The start or end point side of the commanded slope is identified automatically inside the NC unit.
- (5) The end point coordinates of the second block should be commanded with absolute values. When incremental values are used, program error (P393) will result.
- (6) The feedrate can be commanded for each block.
- (7) When the angle where the two straight lines intersect is less than  $1^\circ$ , program error (P392) will result.
- (8) Program error (P396) results when the plane is selected in the 1st block and 2nd block.
- (9) This function is ignored when address A is used for the axis name or as the 2nd miscellaneous function.
- (10) Single block stop is possible at the end point of the 1st block.
- (11) Program error (P394) results when the 1st and 2nd blocks do not contain the G01 or G33 command.

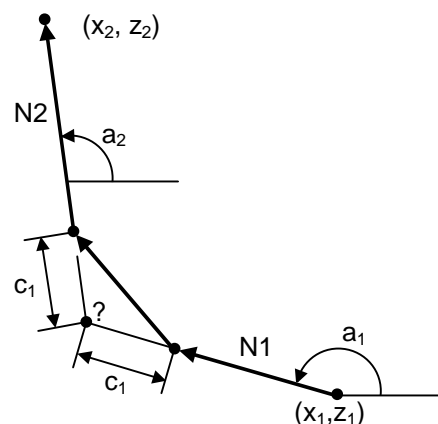


### Relationship with other functions

- (1) Corner chamfering or corner rounding can be commanded after the angle command in the 1st block.

#### (Example 1)

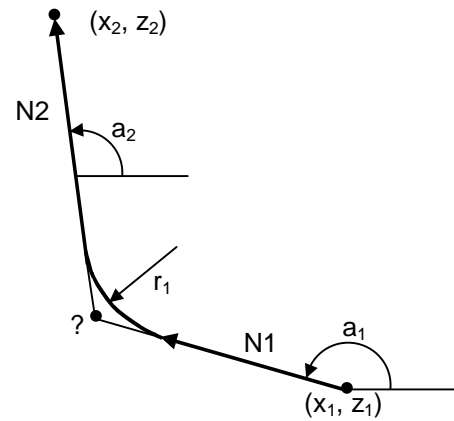
```
N1 Aa1, Cc1 ;
N2 Xx2 Zz2 Aa2 ;
```





**(Example 2)**

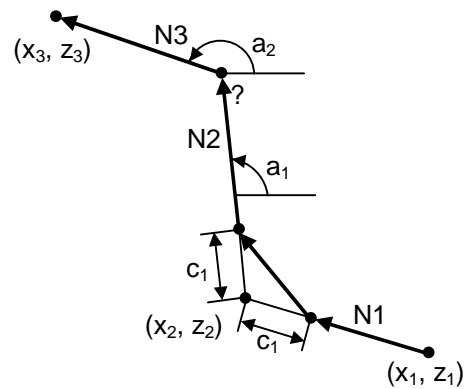
N1 Aa<sub>1</sub>, Rr<sub>1</sub> ;  
 N2 Xx<sub>2</sub> Zz<sub>2</sub> Aa<sub>2</sub> ;



(2) The geometric command I can be issued after the corner chamfering or corner rounding command.

**(Example 3)**

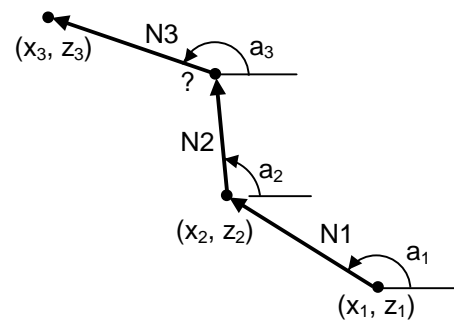
N1 Xx<sub>2</sub> Zz<sub>2</sub>, Cc<sub>1</sub> ;  
 N2 Aa<sub>1</sub> ;  
 N3 Xx<sub>3</sub> Zz<sub>3</sub> Aa<sub>2</sub> ;



(3) The geometric command I can be issued after the linear angle command.

**(Example 4)**

N1 Xx<sub>2</sub> Aa<sub>1</sub> ;  
 N2 Aa<sub>2</sub> ;  
 N3 Xx<sub>3</sub> Zz<sub>3</sub> Aa<sub>3</sub> ;

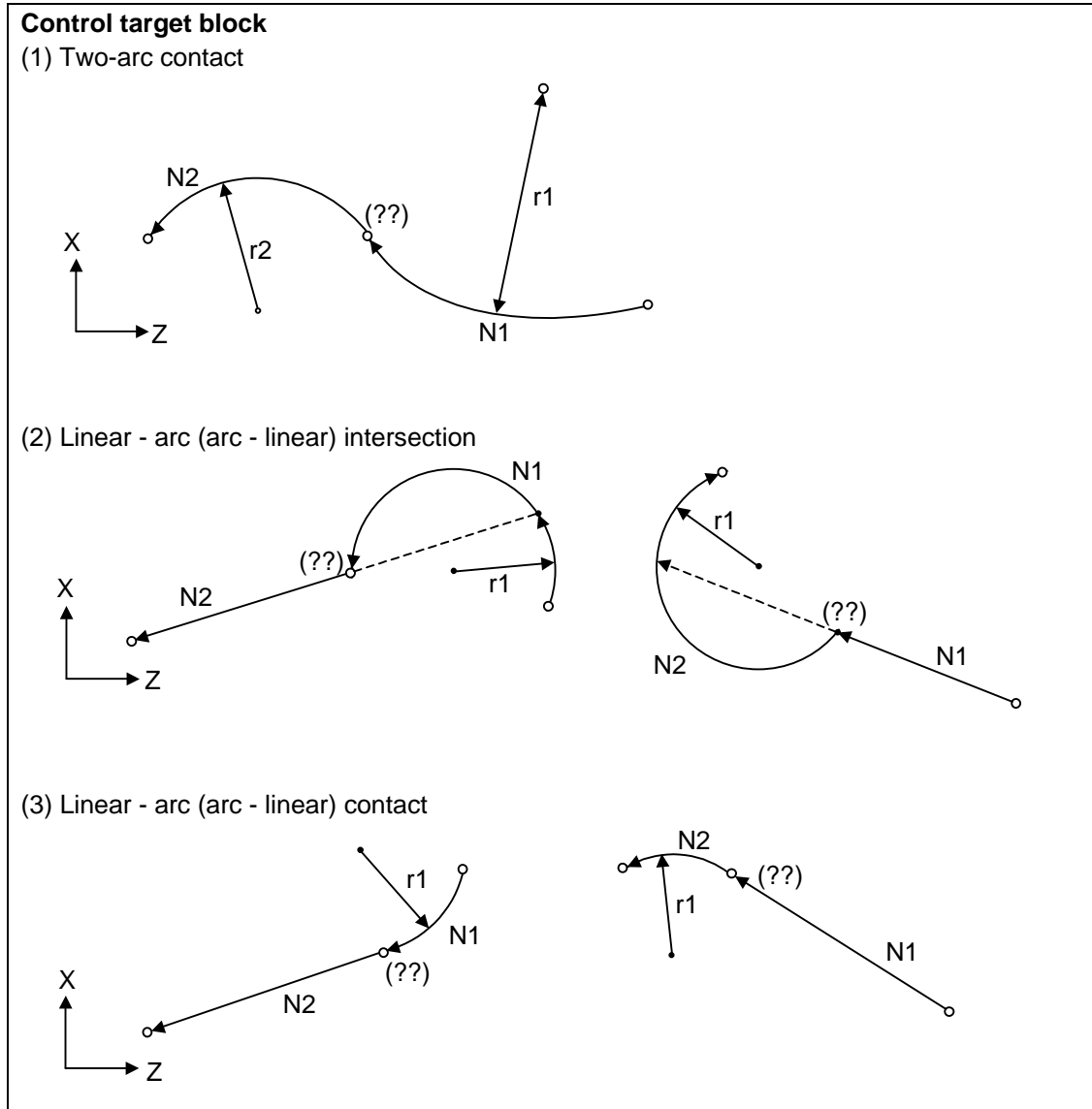


### 13.13.2 Geometric IB

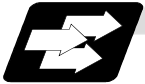


#### Function and purpose

With the geometric IB function, the contact and intersection are obtained by commanding an arc center point or linear angle in the movement commands of two continuous blocks (only for blocks having arc commands), instead of commanding the first block end point.



**(Note)** If the parameter (#1082 Geomet) is set to a value other than 2, geometric IB will not function.



### Function and purpose 1 (Automatic calculation of two-arc contact)

When the contact of two continuous contacting arcs is not indicated in the drawings, it can be automatically calculated by commanding the 1st arc center coordinate value or radius, and the 2nd arc end point absolute value and center coordinate value or radius.

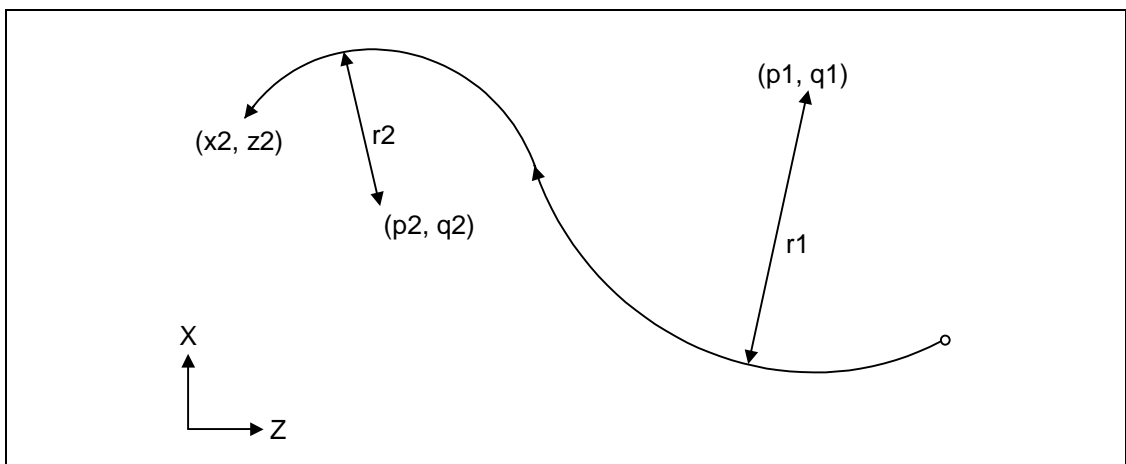


### Command format 1 (Automatic calculation of two-arc contact)

N1	G02 (G03)	Pp <sub>1</sub> Qq <sub>1</sub> Ff <sub>1</sub> ;
N2	G03 (G02)	Xx <sub>2</sub> Zz <sub>2</sub> Pp <sub>2</sub> Qq <sub>2</sub> Ff <sub>2</sub> ;
N1	G02 (G03)	Pp <sub>1</sub> Qq <sub>1</sub> Ff <sub>1</sub> ;
N2	G03 (G02)	Xx <sub>2</sub> Zz <sub>2</sub> Rr <sub>2</sub> Ff <sub>2</sub> ;
N1	G02 (G03)	Rr <sub>1</sub> Ff <sub>1</sub> ;
N2	G03 (G02)	Xx <sub>2</sub> Zz <sub>2</sub> Pp <sub>2</sub> Qq <sub>2</sub> Ff <sub>2</sub> ;
P, Q	X and Z axes arc center coordinate absolute value (diameter/radius value command) The center address for the 3rd axis is commanded with A.	
R	Arc radius (when a (-) sign is attached, the arc is judged to be 180° or more)	

\* I and K (X and Z axes arc center coordinate incremental value) commands can be issued instead of P and Q.

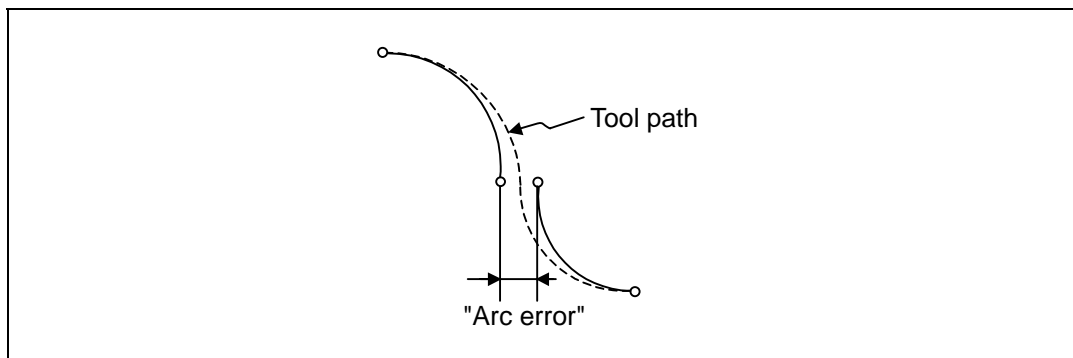
[ 1st block arc : Radius command incremental amount from the start point to the center ]  
 [ 2nd block arc : Radius command incremental amount from the end point to the center ]



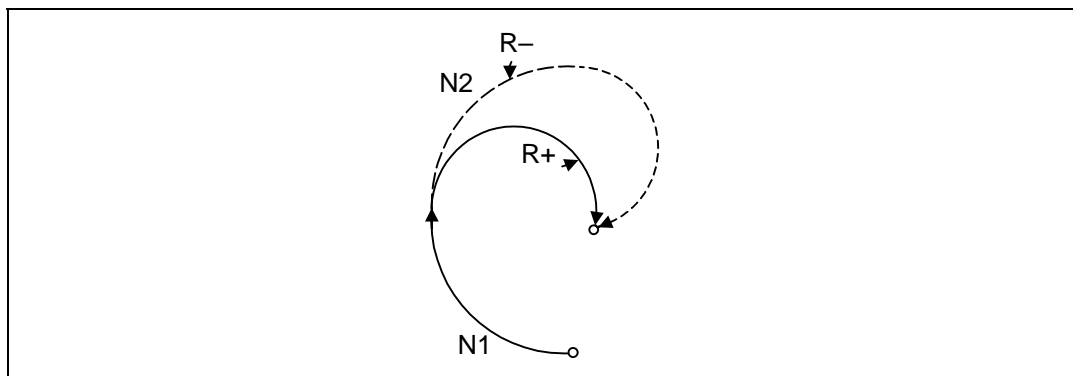


### Detailed description 1 (Automatic calculation of two-arc contact)

- (1) A program error (P393) will result before the 1st block if the 2nd block does not contain a coordinate absolute value command.
- (2) A program error (P398) will result before the 1st block if there are no geometric IB specifications.
- (3) A program error (P395) will result before the 1st block if there is no R (here, the 1st block is designated with P and Q (I, K)) or P and Q (I, K) designation.
- (4) A program error (P396) will result before the 1st block if another plane selection command (G17 to G19) is issued in the 2nd block.
- (5) A program error (P397) will result before the 1st block if two arcs that do not contact are commanded.
- (6) The contact calculation accuracy is  $\pm 1\mu\text{m}$  (fractions rounded up).
- (7) Single block operation stops at the 1st block.
- (8) When I or K is omitted, the values are regarded as I0 and K0. P and Q cannot be omitted.
- (9) The error range in which the contact is obtained is set in parameter "#1084 RadErr".



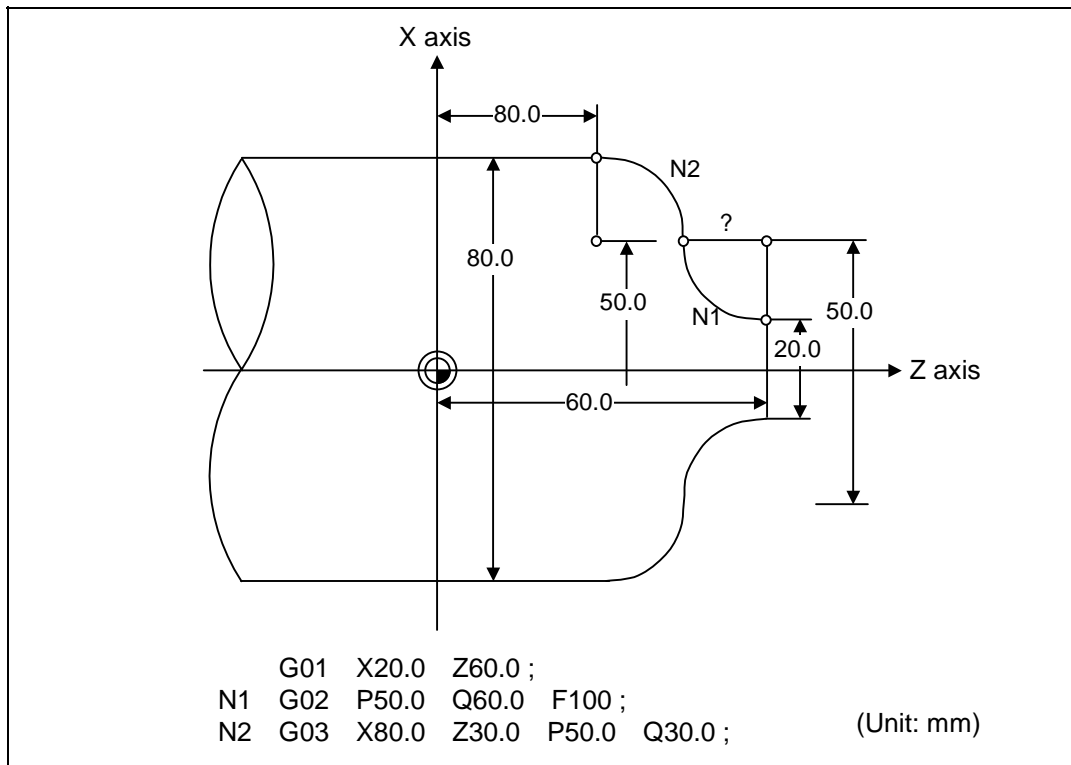
- (10) For an arc block perfect circle command (arc block start point = arc block end point), the R designation arc command finishes immediately, and there is no operation. Thus, use a PQ (IK) designation arc command.
- (11) G codes of the G modal group 1 in the 1st/2nd block can be omitted.
- (12) Addresses being used as axis names cannot be used as command addresses for arc center coordinates or arc radii.
- (13) If the 2nd block arc inscribes the 1st arc block when the 2nd block is an R designation arc, the R+ sign becomes the inward turning arc command, and the R- sign becomes the outward turning arc command.



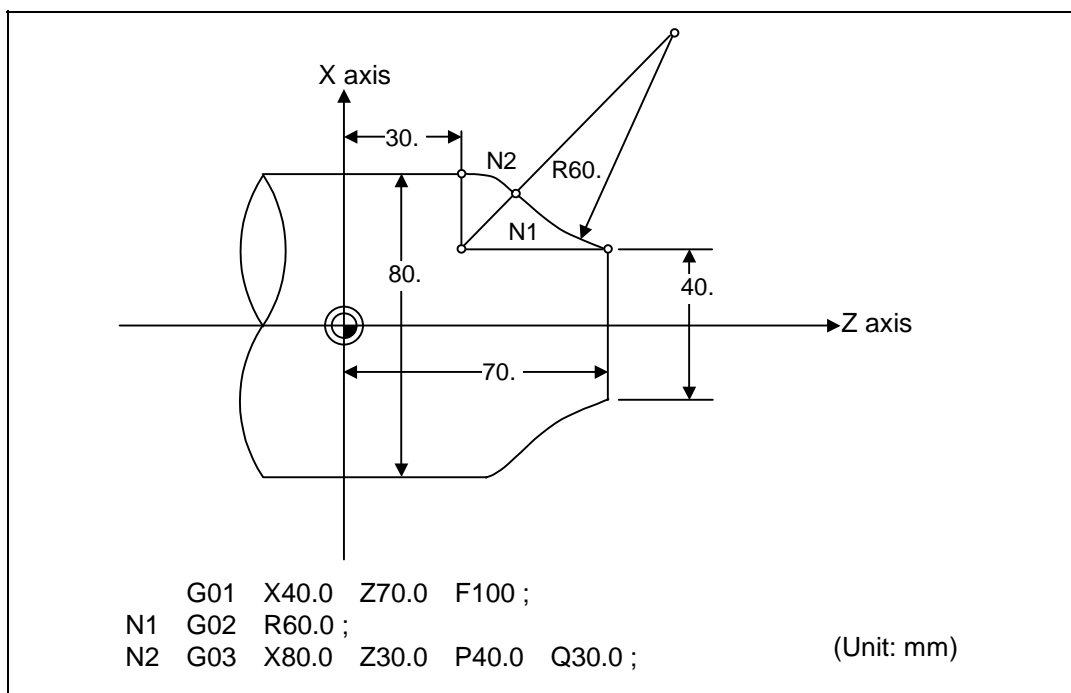


### Example of program 1 (Automatic calculation of two-arc contact)

#### (1) PQ, PQ command



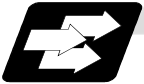
#### (2) PQ, R command





### Relation with other functions 1 (Automatic calculation of two-arc contact)

Command	Tool path
<p>Geometric IB + corner chamfering II</p> <pre>N1 G03 P__ Q__ ; N2 G02 X__ Z__ R__,C__ ; G02 X__ Z__ R__ ;</pre>	
<p>Geometric IB + corner rounding II</p> <pre>N1 G03 P__ Q__ ; N2 G02 X__ Z__ R__,R__ ; G02 X__ Z__ R__ ;</pre>	
<p>Geometric IB + corner chamfering II</p> <pre>N1 G03 P__ Q__ ; N2 G02 X__ Z__ R__,C__ ; G01 X__ Z__ ;</pre>	
<p>Geometric IB + corner rounding II</p> <pre>N1 G03 P__ Q__ ; N2 G02 X__ Z__ R__,R__ ; G02 X__ Z__ ;</pre>	



### Function and purpose 2 (Automatic calculation of linear - arc intersection)

When the intersection of a shape in which a line and arc intersect is not indicated in the drawings, it can be automatically calculated by commanding the following program.

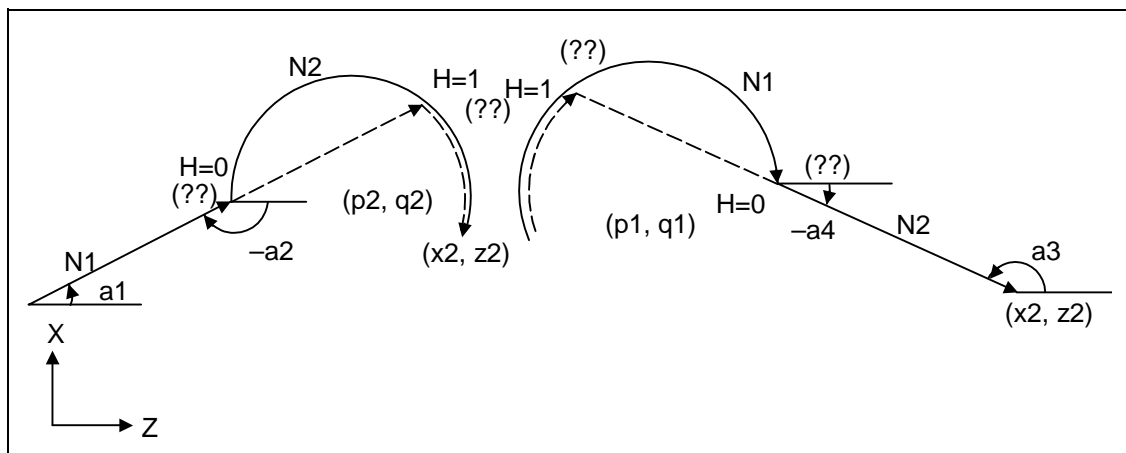


### Command format 2 (Automatic calculation of linear - arc intersection: For G18 plane)

N1	G01	Aa <sub>1</sub> (A-a <sub>2</sub> ) Ff <sub>1</sub> ;
N2	G02 (G03)	Xx <sub>2</sub> Zz <sub>2</sub> Pp <sub>2</sub> Qq <sub>2</sub> Hh <sub>2</sub> Ff <sub>2</sub> ;
N1	G02 (G03)	Pp <sub>1</sub> Qq <sub>1</sub> Hh <sub>1</sub> Ff <sub>1</sub> ;
N2	G01	Xx <sub>2</sub> Zz <sub>2</sub> Aa <sub>3</sub> (A-a <sub>4</sub> ) Ff <sub>2</sub> ;
A	Linear angle (-360.000° to 360.000°)	
P, Q	X and Z axes arc center coordinate absolute value (diameter/radius value command) The center address for the 3rd axis is commanded with A.	
H	Linear - arc intersection selection	[ 0 : Intersection of the shorter line 1 : Intersection of the longer line ]

\* I and K (X and Z axes arc center coordinate incremental value) commands can be issued instead of P and Q.

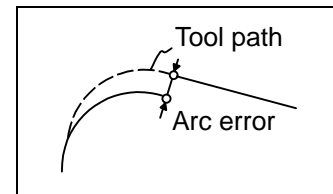
- [ 1st block arc : Radius command incremental amount from the start point to the center ]
- [ 2nd block arc: Radius command incremental amount from the end point to the center ]





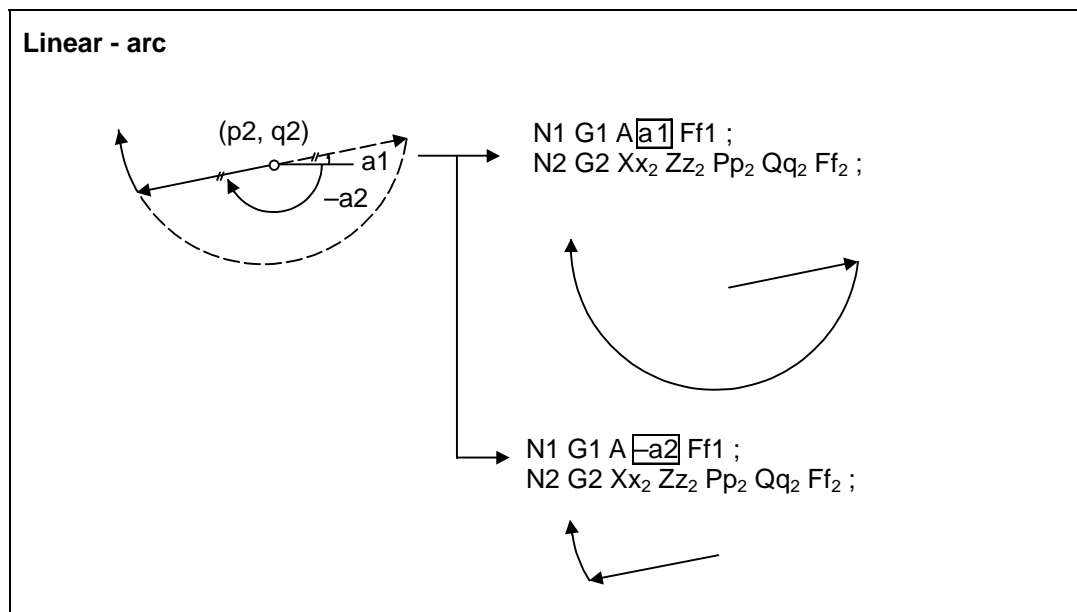
## Detailed description 2 (Automatic calculation of linear - arc intersection)

- (1) When the 2nd miscellaneous function address is A, the 2nd miscellaneous function is validated and this function is invalidated.
- (2) A program error (P393) will result before the 1st block if the 2nd block does not contain a coordinate absolute value command.
- (3) A program error (P398) will result before the 1st block if there are no geometric IB specifications.
- (4) A program error (P395) will result before the 1st block if there is no P and Q (I, K) designation. A program error (P395) will also result if there is no A designation for the line.
- (5) A program error (P396) will result before the 1st block if another plane selection command (G17 to G19) is issued in the 2nd block.
- (6) A program error (P397) will result before the 1st block if a straight line and arc that do not contact or intersect are commanded.
- (7) Single block operation stops at the 1st block.
- (8) When I or K is omitted, the values are regarded as I0 and K0. P and Q cannot be omitted.
- (9) When H is omitted, the value is regarded as H0.
- (10) The linear - arc contact is automatically calculated by designating R instead of P and Q (I, K).
- (11) The error range in which the intersection is obtained is set in parameter "#1084 RadErr".
- (12) The line slope is rotated at the angle regarded as its horizontal axis. Counterclockwise (CCW) is positive (+). Clockwise (CW) is negative (-).
- (13) The line slope can be commanded from either the start point side or the end point side. The judgment whether the designated slope is start point side or end point side is carried out automatically.





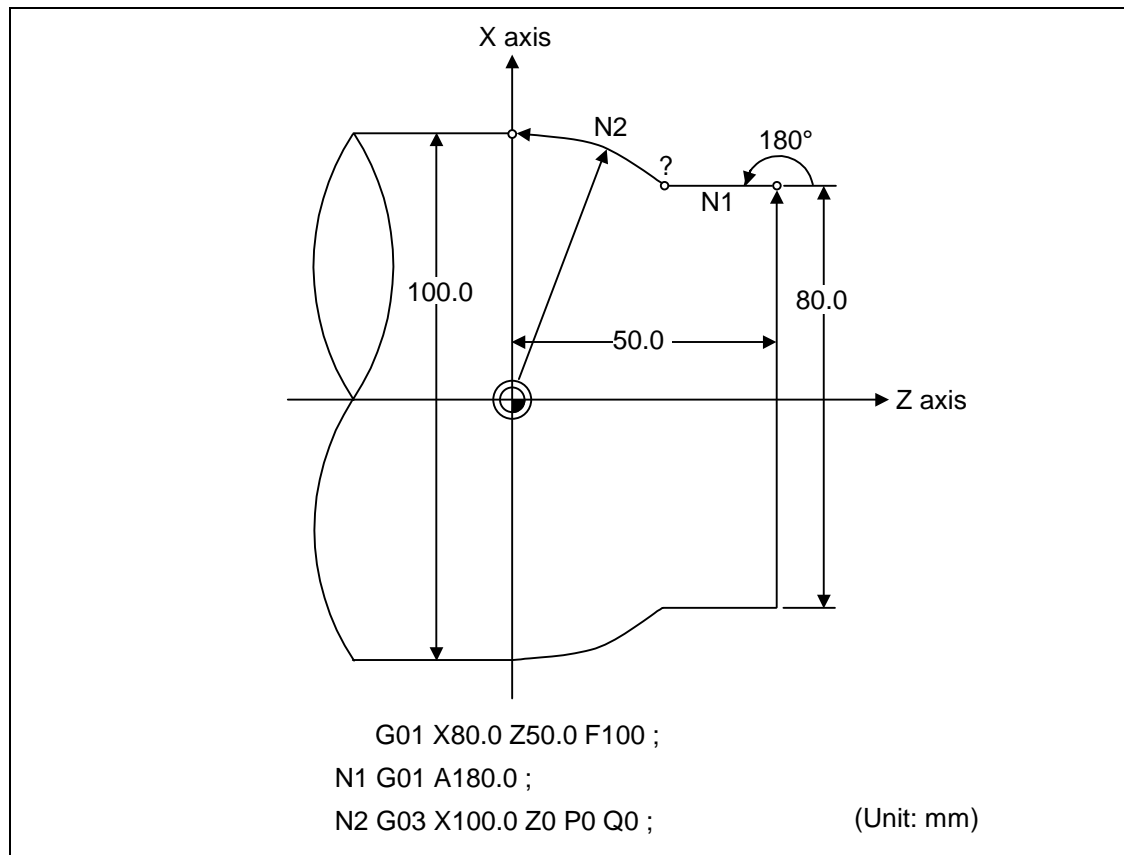
- (14) When the distance to the intersection of the line and arc is the same (as in the drawing below), the control by address H (short/long distance selection) is invalidated. In this case the judgment is carried out based on the angle of the line.



- (15) The intersection calculation accuracy is  $\pm 1\mu\text{m}$  (fractions rounded up).
- (16) In linear - arc intersections, the arc command can only be a PQ (IK) command. When the arc block start point = arc block end point, the arc is a perfect circle.
- (17) G codes of the G modal group in the 1st block can be omitted.
- (18) Addresses being used as axis names cannot be used as command addresses for angles, arc center coordinates or intersection selections.
- (19) When geometric IB is commanded, two blocks are pre-read.



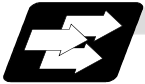
### Example of program 2 (Automatic calculation of linear - arc intersection)





### Relation with other functions 2 (Automatic calculation of linear - arc intersection)

Command	Tool path
<p>Geometric IB + corner chamfering II</p> <p>N1 G01 A__ ,C__ ; N2 G03 X__ Z__ P__ Q__ H__ ;</p>	
<p>Geometric IB + corner rounding II</p> <p>N1 G01 A__ ,R__ ; N2 G03 X__ Z__ P__ Q__ H__ ;</p>	
<p>Geometric IB + corner chamfering II</p> <p>N1 G01 A__ ; N2 G03 X__ Z__ P__ Q__ H__ ,C__ ; G01 X__ Z__ ;</p>	
<p>Geometric IB + corner rounding II</p> <p>N1 G01 A__ ; N2 G03 X__ Z__ P__ Q__ H__ ,R__ ; G01 X__ Z__ ;</p>	
<p>Geometric IB + corner chamfering II</p> <p>N1 G02 P__ Q__ H__ ; N2 G01 X__ Z__ A__ ,C__ ; G01 X__ Z__ ;</p>	
<p>Geometric IB + corner rounding II</p> <p>N1 G02 P__ Q__ H__ ; N2 G01 X__ Z__ A__ ,R__ ; G01 X__ Z__ ;</p>	



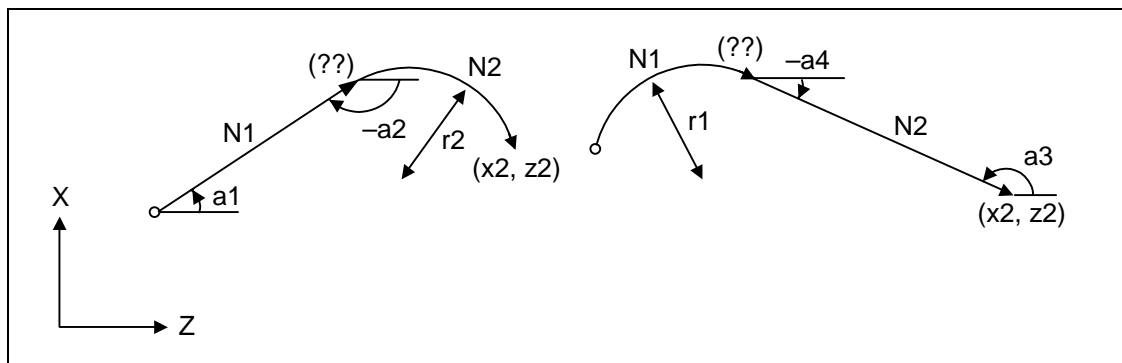
### Function and purpose 3 (Automatic calculation of linear - arc intersection)

When the contact of a shape in which a line and arc contact is not indicated in the drawings, it can be automatically calculated by commanding the following program.



### Command format 3 (Automatic calculation of linear - arc intersection: For G18 plane)

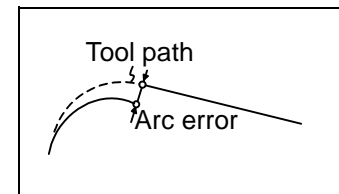
N1	G0	Aa <sub>1</sub> (A-a <sub>2</sub> ) Ff <sub>1</sub> ;
N2	G03 (G02)	Xx <sub>2</sub> Zz <sub>2</sub> Rr <sub>2</sub> ;
N1	G03 (G02)	Rr <sub>2</sub> Ff <sub>1</sub> ;
N2	G01	Xx <sub>2</sub> Zz <sub>2</sub> Aa <sub>3</sub> (A-a <sub>4</sub> ) Ff <sub>2</sub> ;
A		Linear angle (-360.000° to 360.000°)
R		Arc radius





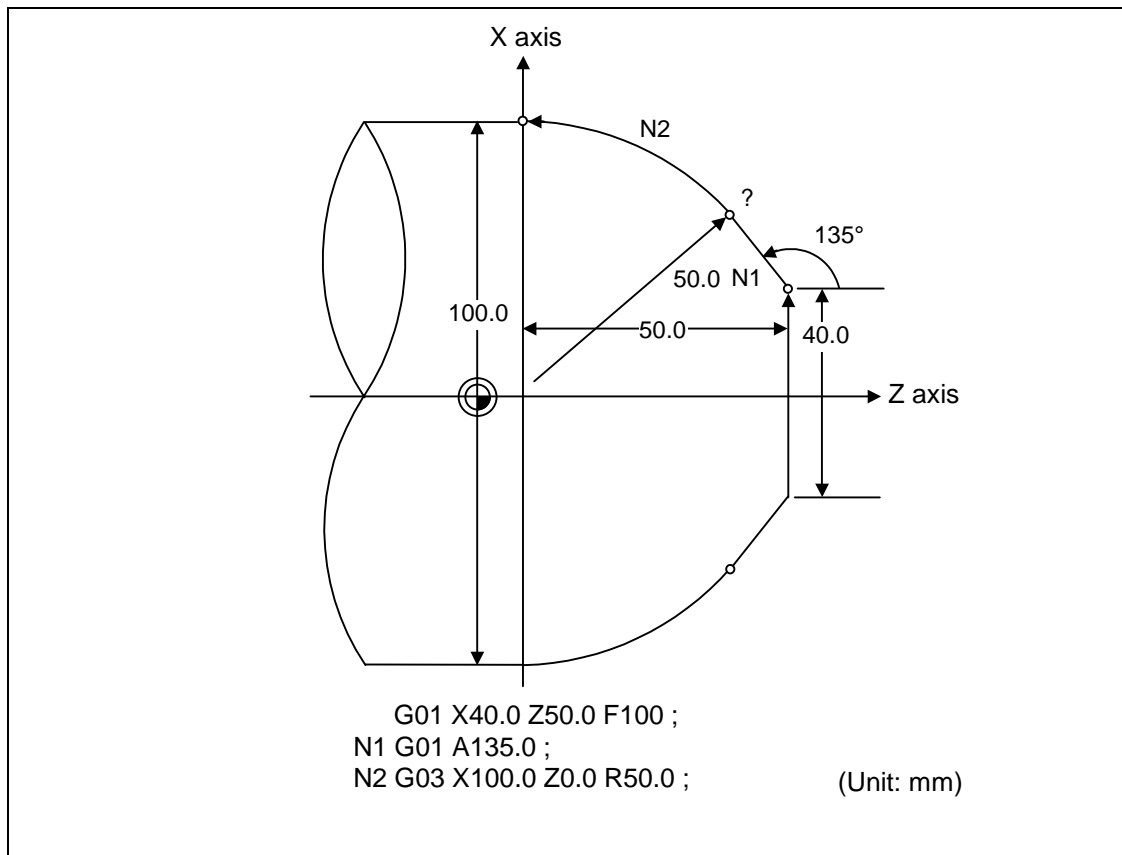
## Detailed description 3 (Automatic calculation of linear - arc intersection)

- (1) When the 2nd miscellaneous function address is A, the 2nd miscellaneous function is validated and this function is invalidated.
- (2) A program error (P393) will result before the 1st block if the 2nd block does not contain a coordinate absolute value command.
- (3) A program error (P398) will result before the 1st block if there are no geometric IB specifications.
- (4) A program error (P396) will result before the 1st block if another plane selection command (G17 to G19) is issued in the 2nd block.
- (5) A program error (P397) will result before the 1st block if a straight line and arc that do not contact are commanded.
- (6) A program error (P395) will result before the 1st block if there is no R designation for the 2nd block. A program error (P395) will also result if there is no A designation for the line.
- (7) Single block operation stops at the 1st block.
- (8) The linear - arc intersection is automatically calculated by designating R instead of P and Q (I, K).
- (9) The error range in which the contact is obtained is set in parameter "#1084 RadErr".
- (10) The line slope is rotated at the angle regarded as the positive (+) direction of its horizontal axis. Counterclockwise (CCW) is positive (+). Clockwise (CW) is negative (-).
- (11) The line slope can be commanded from either the start point side or the end point side. The judgment whether the designated slope is start point side or end point side is carried out automatically.
- (12) The intersection calculation accuracy is  $\pm 1\mu\text{m}$  (fractions rounded up).
- (13) In linear - arc contact, the arc command can only be an R command. Thus, when the arc block start point = arc block end point, the arc command finishes immediately, and there is no operation. (Perfect circle commands are not possible.)
- (14) G codes of the G modal group 1 in the 1st block can be omitted.
- (15) Addresses being used as axis names cannot be used as command addresses for angles or arc radii.
- (16) When geometric IB is commanded, two blocks are pre-read.





### Example of program 3 (Automatic calculation of linear - arc intersection)





### Relation with other functions 3 (Automatic calculation of linear - arc intersection)

Command	Tool path
<p>Geometric IB + corner chamfering</p> <pre>N1 G03 R__ ; N2 G01 X__ Z__ A__,C__ ;    G01 X__ Z__ R__ ;</pre>	
<p>Geometric IB + corner rounding</p> <pre>N1 G03 R__ ; N2 G01 X__ Z__ A__,R__ ;    G01 X__ Z__ ;</pre>	
<p>Geometric IB + corner chamfering II</p> <pre>N1 G01 A__ ; N2 G02 X__ Z__ R__,C__ ;    G01 X__ Z__ ;</pre>	
<p>Geometric IB + corner rounding II</p> <pre>N1 G01 A__ ; N2 G02 X__ Z__ R__,R__ ;    G01 X__ Z__ ;</pre>	

### 13.14 Program parameter input; G10, G11



#### Function and purpose

The parameters set from the setting and display unit can be changed in the machining programs. The data format used for the data setting is as follows.



#### Command format

##### G10 L50 ; Data setting command

**P major classification number N data number H□ bit type data ;**  
**P major classification number A axis number N data number D byte type data ;**  
**P major classification number A axis number N data number S word type data ;**  
**P major classification number A axis number N data number L 2 word type data ;**

##### G11 ; Data setting mode cancel (data setting completed)

There are 8 types of data formats according to the type of parameter (axis-common and axis-independent) and data type, as listed below.

##### With axis-common data

Axis-common bit-type parameter	P__ N__ H□__ ;
Axis-common byte-type parameter	P__ N__ D__ ;
Axis-common word-type parameter	P__ N__ S__ ;
Axis-common 2-word-type parameter	P__ N__ L__ ;

##### With axis-independent data

Axis-independent bit-type parameter	P__ A__ N__ H□__ ;
Axis-independent byte-type parameter	P__ A__ N__ D__ ;
Axis-independent word-type parameter	P__ A__ N__ S__ ;
Axis-independent 2-word-type parameter	P__ A__ N__ L__ ;

**(Note 1)** The sequence of addresses in a block must be as shown above.

**(Note 2)** Refer to Appendix Table 1 for the P, N number correspondence table.

**(Note 3)** For a bit type parameter, the data type will be H□ (□ is a value between 0 and 7).

**(Note 4)** The axis number is set in the following manner: 1st axis is 1, 2nd axis is 2, and so forth.



## 13.15 Macro interrupt



## Function and purpose

A user macro interrupt signal (UIT) is input from the machine to interrupt the program being currently executed and instead call another program and execute it. This is called the user macro interrupt function.

Use of this function allows the program to operate flexibly enough to meet varying conditions. For setting the parameters of this function, refer to the Alarm/Parameter Manual.



## Command format

<b>M96 P__ H__ ;</b>	<b>User macro interrupt enable</b>
<b>M97 ;</b>	<b>User macro interrupt disable</b>
P	Interrupt program number
H	Interrupt sequence number

The user macro interrupt function is enabled and disabled by the M96 and M97 commands programmed to make the user macro interrupt signal (UIT) valid or invalid. That is, if an interrupt signal (UIT) is input from the machine side in a user macro interrupt enable period from when M96 is issued to when M97 is issued or the NC is reset, a user macro interrupt is caused to execute the program specified by P\_\_ instead of the one being executed currently.

Another interrupt signal (UIT) is ignored while one user macro interrupt is being in service. It is also ignored in a user macro interrupt disable state such as after an M97 command is issued or the system is reset.

M96 and M97 are processed internally as user macro interrupt control M codes.



## Interrupt enable conditions

A user macro interrupt is enabled only during execution of a program. The requirements for the user macro interrupt are as follows:

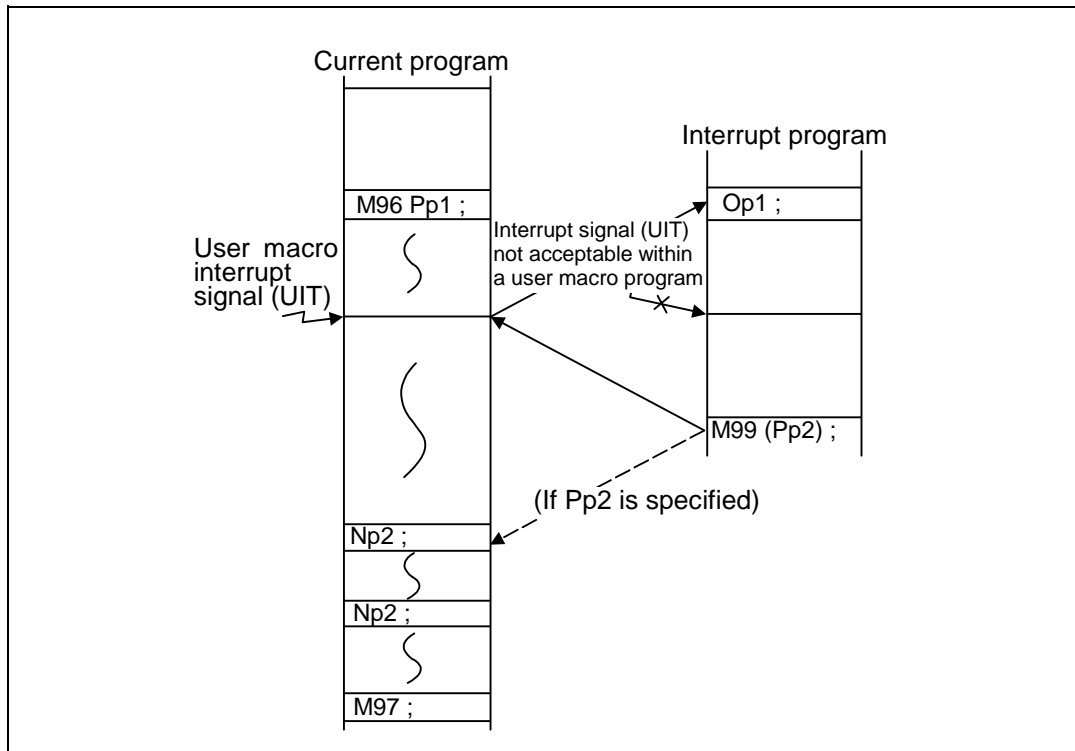
- (1) An automatic operation mode, memory or tape, or MDI has been selected.
- (2) The system is running in automatic mode.
- (3) No user macro interrupt is being processed.

**(Note 1)** A macro interrupt is disabled in manual operation mode (JOG, STEP, HANDLE, etc.)



### Outline of operation

- (1) When a user macro interrupt signal (UIT) is input after an M96 Pp1 ; command is issued by the current program, interrupt program Op1 is executed. When an M99 ; command is issued by the interrupt program, control returns to the main program.
- (2) If M99 Pp2 ; is specified, the blocks from the one next to the interrupted block to the last one are searched for the block with sequence number Np2 ;. Control thus returns to the block with sequence number Np2 ; that is found first in the above search.



**Interrupt type**

Interrupt types 1 and 2 can be selected by the parameter "#1113 INT\_2".

**[Type 1]**

- When an interrupt signal (UIT) is input, the system immediately stops moving the tool and interrupts dwell, then permits the interrupt program to run.
- If the interrupt program contains a move or miscellaneous function (MSTB) command, the commands in the interrupted block are lost. After the interrupt program completes, the main program resumes operation from the block next to the interrupted one.
- If the interrupted program contains no move and miscellaneous (MSTB) commands, it resumes operation, after completion of the interrupt program, from the point in the block where the interrupt was caused.

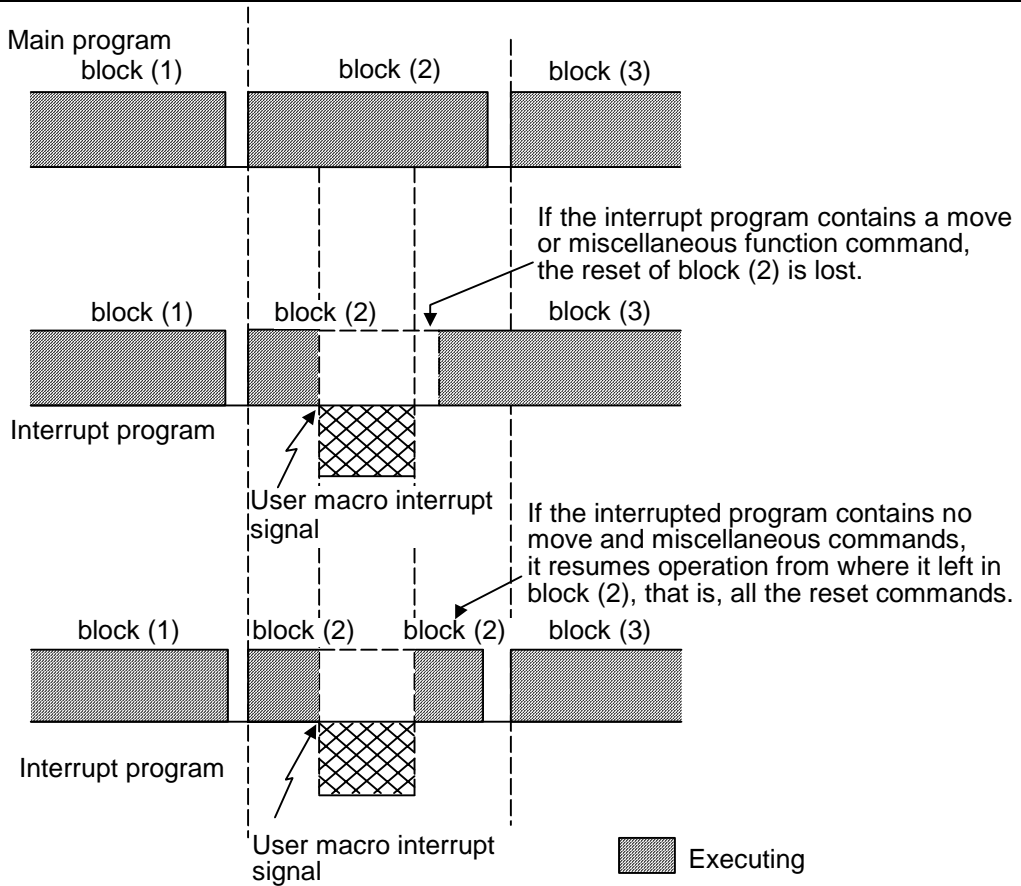
If an interrupt signal (UIT) is input during execution of a miscellaneous function (MSTB) command, the NC system waits for a completion signal (FIN). The system thus executes a move or miscellaneous function (MSTB) command in the interrupt program only after input of FIN.

**[Type 2]**

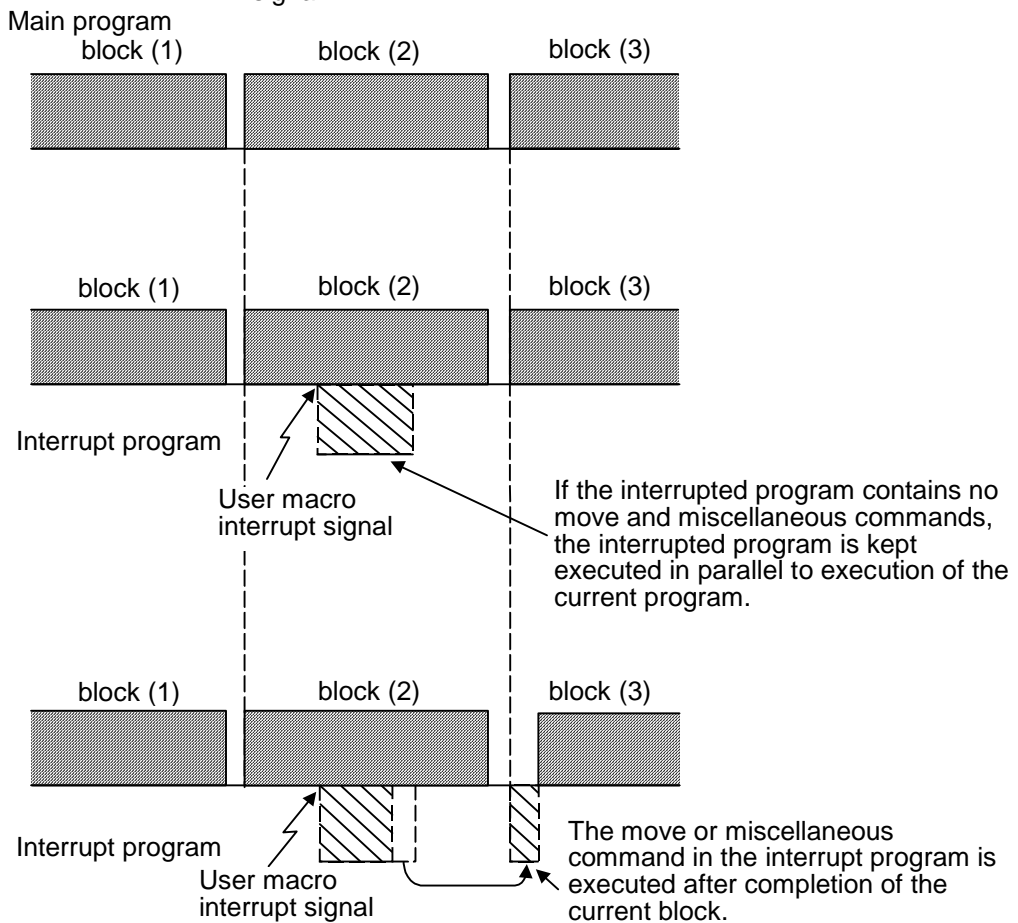
- When an interrupt signal (UIT) is input, the program completes the commands in the current block, then transfers control to the interrupt program.
- If the interrupt program contains no move and miscellaneous function (MSTB) commands, the interrupt program is executed without interrupting execution of the current program.

However, if the interrupt program has not ended even after the execution of the original block is completed, the system may stop machining temporarily.

[Type 1]



[Type 2]





### Calling method

User macro interrupt is classified into the following two types depending on the way an interrupt program is called. These two types of interrupt are selected by parameter "#1229 set01/bit0".

- a. Subprogram type interrupt
- b. Macro type interrupt

Both types of interrupt are included in calculation of the nest level. The subprograms and user macros called in the interrupt program are also included in calculation of the nest level.

Subprogram type interrupt	The user macro interrupt program is called as a subprogram. As with calling by M98, the local variable level remains unchanged before and after an interrupt.
Macro type interrupt	The user macro interrupt program is called as a user macro. As with calling by G65, the local variable level changes before and after an interrupt. No arguments in the main program can be passed to the interrupt program.



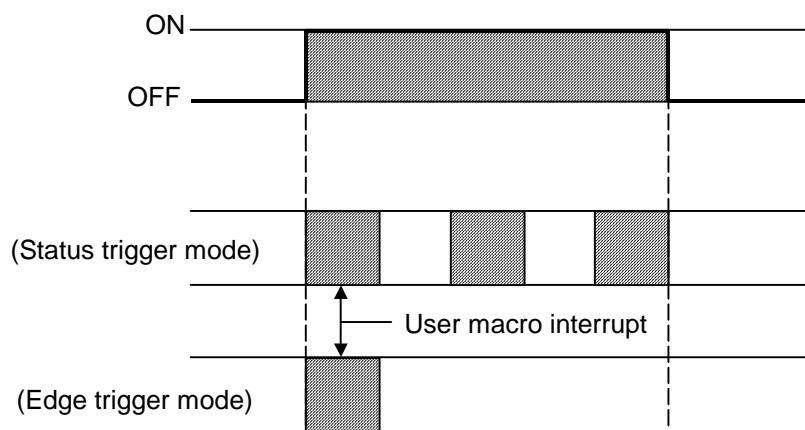
### Acceptance of user macro interrupt signal (UIT)

A user macro interrupt signal (UIT) is accepted in the following two modes. These two modes are selected by a parameter "#1112 S\_TRG".

- a. Status trigger mode
- b. Edge trigger mode

Status trigger mode	The user macro interrupt signal (UIT) is accepted as valid when it is ON. If the interrupt signal (UIT) is ON when the user macro interrupt function is enabled by M96, the interrupt program is activated. By keeping the interrupt signal (UIT) ON, the interrupt program can be executed repeatedly.
Edge trigger mode	The user macro interrupt signal (UIT) is accepted as valid at its rising edge, that is, at the instance it turns ON. This mode is useful to execute an interrupt program once.

#### User macro interrupt signal (UIT)



#### Accepting user macro interrupt signal (UIT)



### Returning from user macro interrupt

M99 (P\_\_);

An M99 command is issued in the interrupt program to return to the main program. Address P is used to specify the sequence number of the return destination in the main program. The blocks from the one next to the interrupted block to the last one in the main program are first searched for the block with sequence number Np2;. If it is not found, all the blocks before the interrupted one are then searched. Control thus returns to the block with sequence number that is found first in the above search.

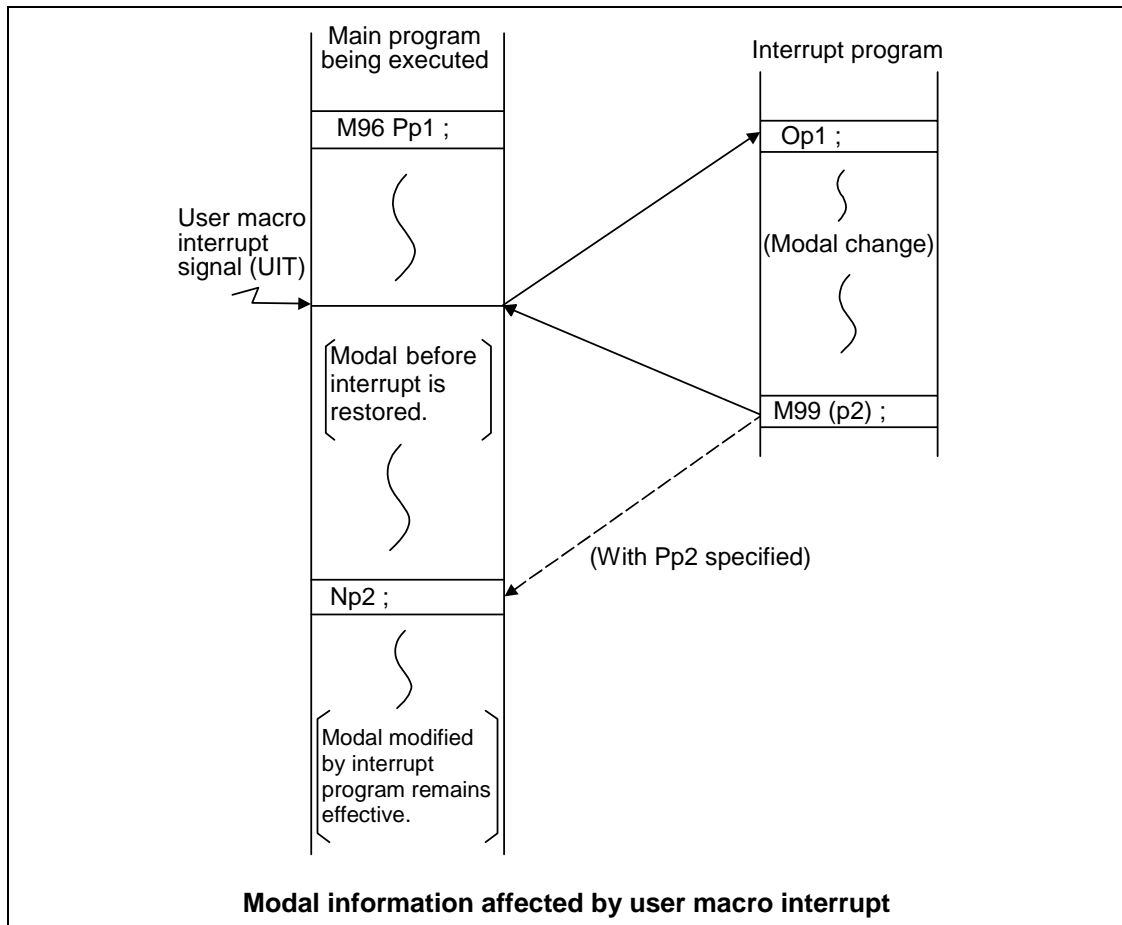
(This is equivalent to M99 P\_\_ used after M98 calling.)



### Modal information affected by user macro interrupt

If modal information is changed by the interrupt program, it is handled as follows after control returns from the interrupt program to the main program.

Returning with M99;	The change of modal information by the interrupt program is invalidated and the original modal information is not restored. With interrupt type 1, however, if the interrupt program contains a move or miscellaneous function (MSTB) command, the original modal information is not restored.
Returning with M99 P__;	The original modal information is updated by the change in the interrupt program even after returning to the main program. This is the same as in returning with M99 P__; from a program called by M98.



**Modal information variables (#4401 to #4520)**

Modal information when control passes to the user macro interrupt program can be known by reading system variables #4401 to #4520.

The unit specified with a command applies.

System variable	Modal information
#4401 ~ #4421	G code (group 01 to group 21)
#4507	D code
#4509	F code
#4511	H code
#4513	M code
#4514	Sequence number
#4515	Program number
#4519	S code
#4520	T code

Some groups are not used.

The above system variables are available only in the user macro interrupt program. If they are used in other programs, program error (P241) results.

**M code for control of user macro interrupt**

The user macro interrupt is controlled by M96 and M97. However, these commands may have been used for other operation. To be prepared for such case, these command functions can be assigned to other M codes.

(This invalidates program compatibility.)

User macro interrupt control with alternate M codes is possible by setting the alternate M code in parameters "#1110 M96\_M" and "#1111 M97\_M" and by validating the setting by selecting parameter "#1109 subs\_M".

(M codes 03 to 97 except 30 are available for this purpose.)

If the parameter "#1109 subs\_M" used to enable the alternate M codes is not selected, the M96 and M97 codes remain effective for user macro interrupt control.

In either case, the M codes for user macro interrupt control are processed internally and not output to the outside.



## Parameters

Refer to the Operating Manual for details on the setting methods.

- (1) Subprogram call validity "#1229 set01/bit 0"
  - 1 : Subprogram type user macro interrupt
  - 0 : Macro type user macro interrupt
- (2) Status trigger mode validity "#1112 S\_TRG"
  - 1 : Status trigger mode
  - 0 : Edge trigger mode
- (3) Interrupt type 2 validity "#1113 INT\_2"
  - 1 : The executable statements in the interrupt program are executed after completion of execution of the current block. (Type 2)
  - 0 : The executable statements in the interrupt program are executed before completion of execution of the current block. (Type 1)
- (4) Validity of alternate M code for user macro interrupt control "#1109 subs\_M"
  - 1 : Valid
  - 0 : Invalid
- (5) Alternate M codes for user macro interrupt
  - Interrupt enable M code (equivalent to M96) "#1110 M96\_M"
  - Interrupt disable M code (equivalent to M97) "#1111 M97\_M"

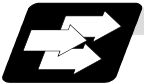


## Restrictions

- (1) If the user macro interrupt program uses system variables #5001 and after (position information) to read coordinates, the coordinates pre-read in the buffer are used.
- (2) If an interrupt is caused during execution of the tool diameter compensation, a sequence number (M99 P\_\_;) must be specified with a command to return from the user macro interrupt program. If no sequence number is specified, control cannot return to the main program normally.



### 13.16 Tool change position return; G30.1 ~ G30.5



#### Function and purpose

By specifying the tool change position in a parameter (#8206 tool change) and also specifying a tool change position return command in a machining program, the tool can be changed at the most appropriate position.

The axes that are going to return to the tool change position and the order in which the axes begin to return can be changed by commands.



#### Command format

- (1) The format of tool change position return commands is as follows.

**G30.n ;**  
 n = 1 to 5 Specify the axes that return to the tool change position and the order in which they return.

#### Commands and return order

Command	Return order
G30.1	X axis only ( → additional axis)
G30.2	Z axis only ( → additional axis)
G30.3	X axis → Z axis ( → additional axis)
G30.4	Z axis → X axis ( → additional axis)
G30.5	X axis • Z axis ( → additional axis)

**(Note 1)** An arrow ( → ) indicates the order of axes that begin to return. A period ( • ) indicates that the axes begin to return simultaneously.

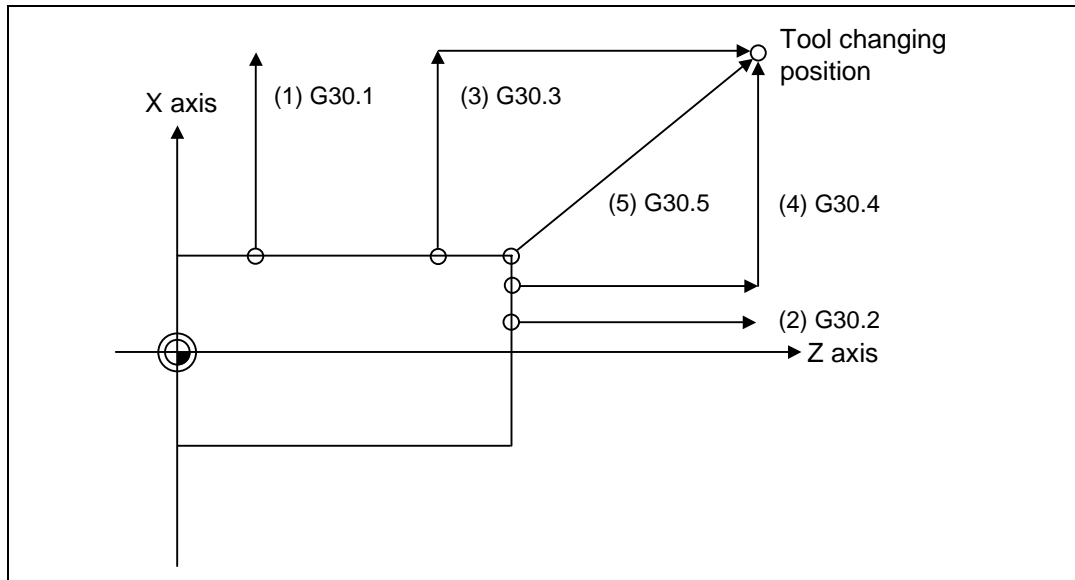
**Example :** "Z axis → X axis" indicate that the Z axis returns to the tool change position, then the X axis does.

- (2) The tool change position return on/off for the additional axis can be set with parameter "#1092 Tchg\_A" for the additional axis. For the order for returning to the tool change position, the axes return after the standard axis completes the return to the tool change position (refer to above table). For specifications having two additional axes, the additional axes return to the tool change position after the standard axis has finished its return to the tool change position. The additional axis alone cannot return to the tool change position.
- (3) When the axis address is commanded in the same block as the tool change position return command, the program error (P33) will occur.



## Detailed description

- (1) The following drawing shows an example of operation during a tool change position return command.



- (1) G30.1 command:

X axis only returns to the tool change position. (If the tool change position return is validated for the additional axis, the additional axis also returns to the tool change position after the X axis reaches the tool change position.)

- (2) G30.2 command:

Z axis only returns to the tool change position. (If the tool change position return is validated for the additional axis, the additional axis also returns to the tool change position after the Z axis reaches the tool change position.)

- (3) G30.3 command:

X axis returns to the tool change position, then the Z axis does the same thing. (If the tool change position return is validated for the additional axis, the additional axis also returns to the tool change position after the X and Z axes reaches the tool change position.)

- (4) G30.4 command:

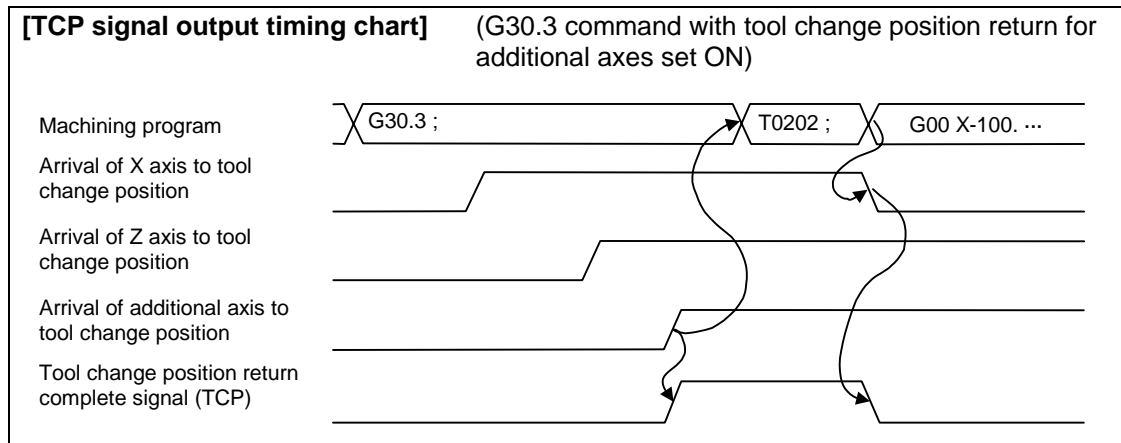
Z axis returns to the tool change position, then the X axis does the same thing. (If the tool change position return is validated for the additional axis, the additional axis also returns to the tool change position after the X and Z axes reaches the tool change position.)

- (5) G30.5 command:

X and Z axes return to the tool change position simultaneously. (If the tool change position return is validated for the additional axis, the additional axis also returns to the tool change position after the X and Z axes reaches the tool change position.)

- (2) After all necessary tool change position return is completed by a G30.n command, tool change position return complete signal TCP (X22B) is turned ON. When an axis out of those having returned to the tool change position by a G30.n command leaves the tool change position, the TCP signal is turned OFF.

With a G30.3 command, for example, the TCP signal is turned ON when the Z axis has reached the tool change position after the X axis did. The TCP signal is then turned OFF when the X or Z axis leaves the position. If tool change position return for additional axes is ON with parameter "#1092 Tchg\_A", the TCP signal is turned ON when the additional axis or axes have reached the tool change position after the standard axes did. It is then turned OFF when one of the X, Z, and additional axes leaves the position.



- (3) Tool compensation data such as tool length offset and tool nose wear compensation are temporarily canceled by the tool change position return command. The machine moves to the tool change position set in the parameters, but because the tool compensation amount is stored in the memory, it moves by the next movement command to a position with the tool compensation applied.
- (4) This command is executed by dividing blocks for every axis. If this command is issued during single-block operation, therefore, a block stop occurs each time one axis returns to the tool change position. To make the next axis tool change position return, therefore, a cycle start needs to be specified.

# 14. Coordinate System Setting Functions

## 14.1 Coordinate words and control axes

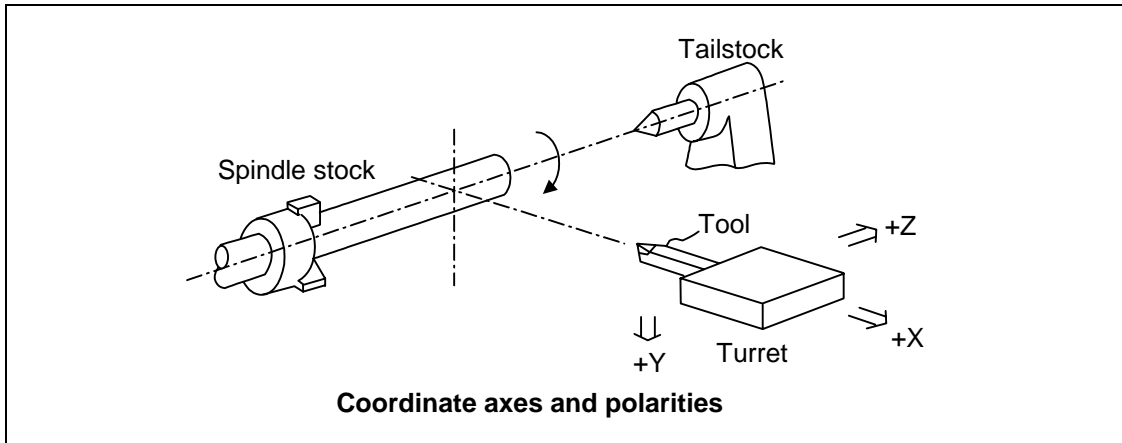
### 14. Coordinate System Setting Functions

#### 14.1 Coordinate words and control axes

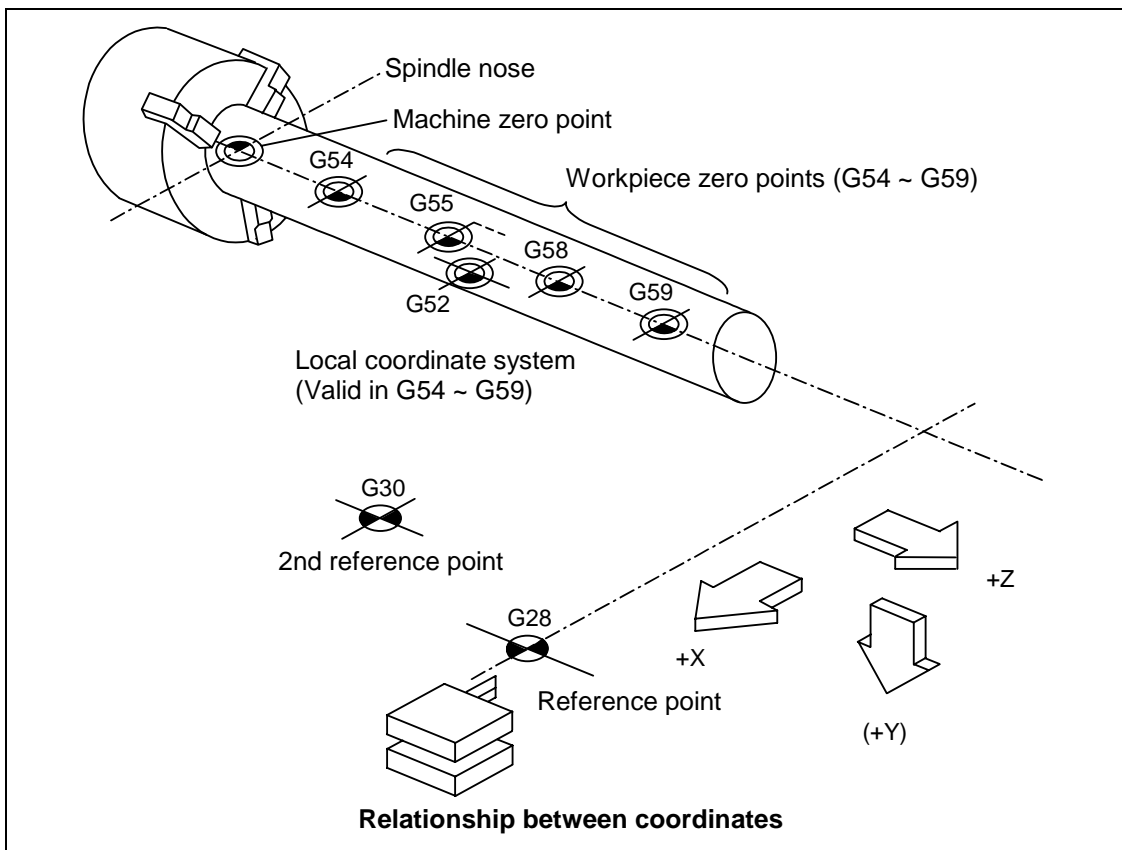


#### Function and purpose

In the case of a lathe, the axis parallel to the spindle is known as the Z axis and its forward direction is the direction in which the turret moves away from the spindle stock while the axis at right angle to the Z axis is the X axis and its forward direction is the direction in which it moves away from the Z axis, as shown in the figure below.



Since coordinates based on the right hand rule are used with a lathe, the forward direction of the Y axis in the above figure which is at right angles to the X-Z plane is downward. It should be borne in mind that an arc on the X-Z plane is expressed as clockwise or counterclockwise as seen from the forward direction of the Y axis. (Refer to the section on circular interpolation.)



## 14. Coordinate System Setting Functions

### 14.2 Basic machine, workpiece and local coordinate systems

#### 14.2 Basic machine, workpiece and local coordinate systems



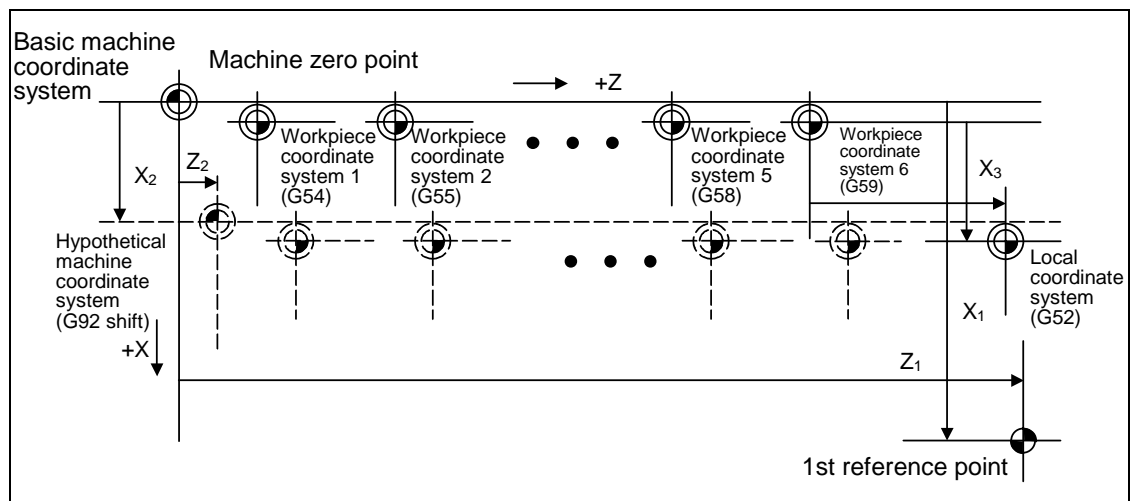
##### Function and purpose

The basic machine coordinate system is fixed in the machine and it denotes that position which is determined inherently by the machine.

The workpiece coordinate systems are used for programming and in these systems the reference point on the workpiece is set as the coordinate zero point.

The local coordinate systems are created on the workpiece coordinate systems and they are designed to facilitate the programs for parts machining.

Upon completion of the reference point return, the basic machine coordinate system and workpiece coordinate systems (G54 ~ G59) are automatically set with reference to the parameters. The basic machine coordinate system is set so that the first reference point is brought to the position specified by the parameter from the basic machine coordinate zero point (machine zero point).



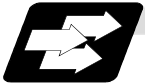
The local coordinate systems (G52) are valid on the coordinate systems designated by workpiece coordinate systems 1 to 6.

The hypothetical machine coordinate system can be set on the basic machine coordinate system using a G92 command. At this time, the workpiece coordinate system 1 to 6 is also simultaneously shifted.

## 14. Coordinate System Setting Functions

### 14.3 Machine zero point and 2nd reference point (Zero point)

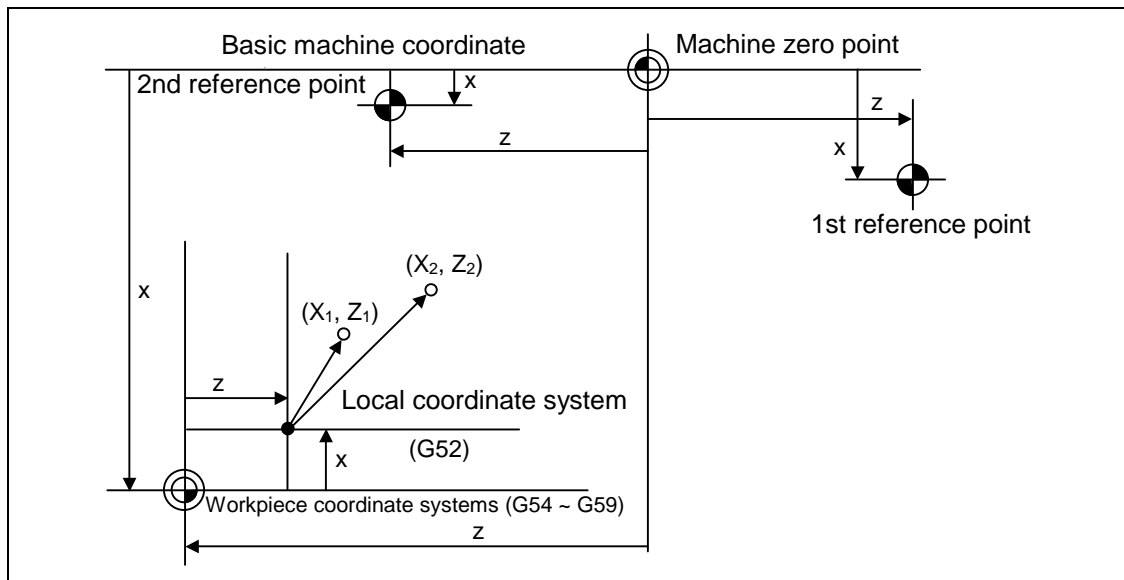
#### 14.3 Machine zero point and 2nd reference point (Zero point)



#### Function and purpose

The machine zero point serves as the reference for the basic machine coordinate system. It is inherent to the machine and is determined by the reference (zero) point return.

The 2nd reference (zero) point relates to the position of the coordinates which have been set beforehand by parameter from the basic machine coordinate system zero point.



## 14. Coordinate System Setting Functions

### 14.4 Automatic coordinate system setting

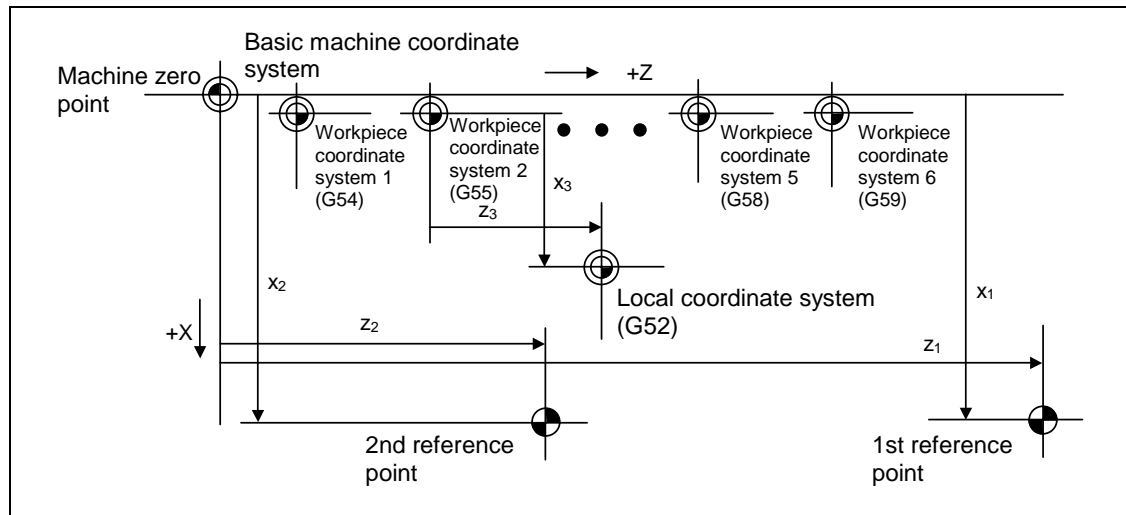
#### 14.4 Automatic coordinate system setting



##### Function and purpose

When the tool has arrived at the reference point by dog-type reference point return after the CNC's power has been turned ON, this function create the various coordinate systems in accordance with the parameter values input beforehand from the setting and display unit.


The actual machining program is programmed over the coordinate systems which have been set above.



##### Detailed description

- (1) The coordinate systems created by this function are as follows:
  - (a) Basic machine coordinate system
  - (b) Workpiece coordinate systems (G54 ~ G59)  
The local coordinate system (G52) is canceled.
- (2) The coordinate related parameters all provide the distance from the basic machine coordinate system zero point. Therefore it is decided at which position in the basic machine coordinate system the 1st reference point should be set and then the zero point positions of the workpiece coordinate systems are set.
- (3) When the automatic coordinate system setting function is executed, the following functions are canceled: workpiece coordinate system shift based on G92, local coordinate system setting based on G52, workpiece coordinate system shift based on origin setting and workpiece coordinate system shift based on manual interrupt.
- (4) When a parameter has been used to select the dog-type of first manual reference point return or automatic reference point return after the power has been turned ON, the dog-type reference point return will be executed for the 2nd and subsequent manual reference point returns or automatic reference point returns.

### CAUTION

-  If the workpiece coordinate offset amount is changed during automatic operation (including single block operation), the changes will be valid from the next block of the command several blocks later.

## 14. Coordinate System Setting Functions

### 14.5 Machine coordinate system selection

#### 14.5 Machine coordinate system selection; G53



##### Function and purpose

The tool is moved to a command position in the basic machine coordinate system, using a G53 command and feed mode command (G01 or G00) and the coordinate command following them.



##### Command format

**G53 G00 Xx Zz αα ;**

**G53 G00 Uu Ww ββ ;**

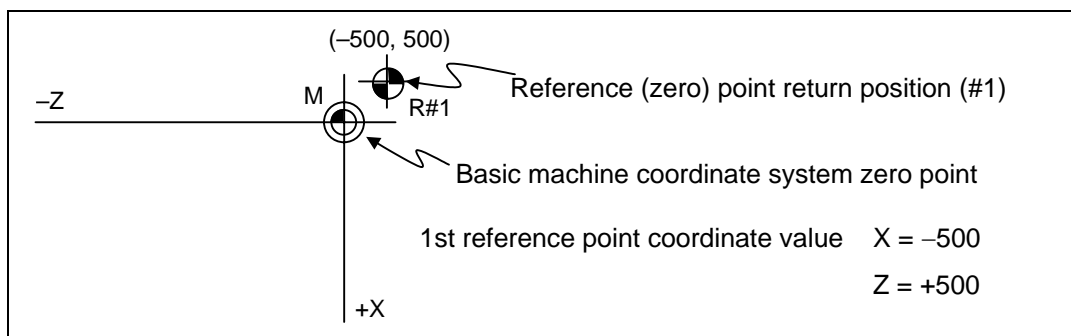
αα Additional axis

ββ Incremental command axis of the additional axis



##### Detailed description

- (1) When the power is turned ON, the basic machine coordinate system is automatically set as referenced to the reference (zero) point return position, which is determined by the automatic or manual reference (zero) point return.
- (2) The basic machine coordinate system is not changed by the G92 command.
- (3) The G53 command is valid only in the block in which it has been designated.
- (4) In the incremental value command mode (U,W,β), the G53 command provides movement with the incremental value in the coordinate system being selected.
- (5) The 1st reference point coordinates denote the distance from the basic machine coordinate system zero point to the reference (zero) point return position.

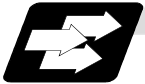




## 14. Coordinate System Setting Functions

### 14.6 Coordinate system setting

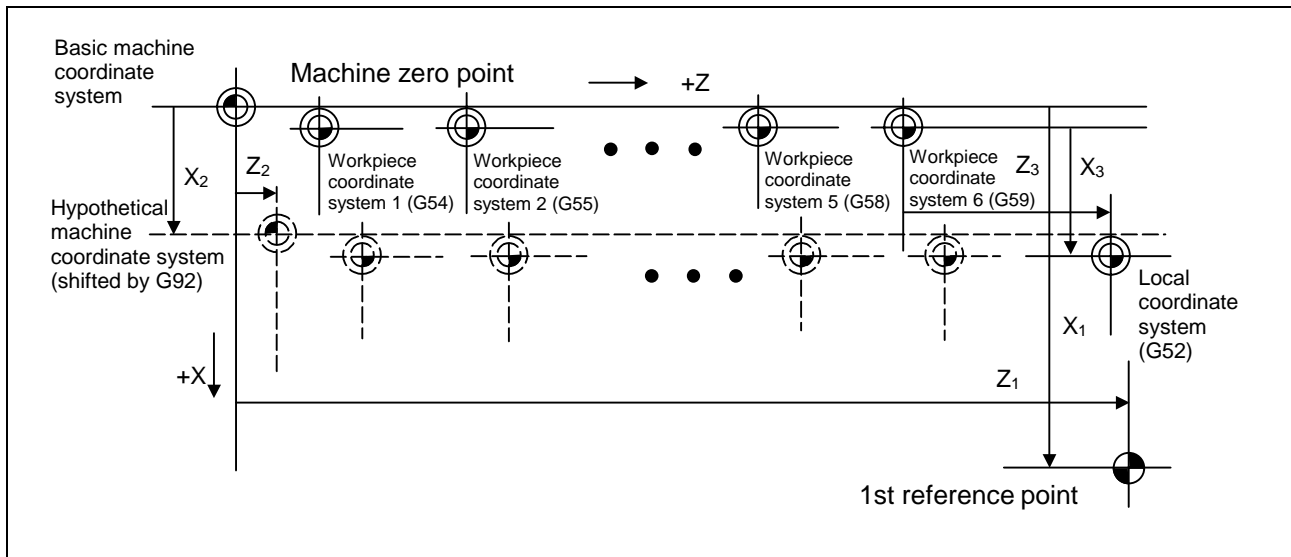
#### 14.6 Coordinate system setting; G92



##### Function and purpose

This function places the tool at the desired position, and the coordinate system is set by assigning the coordinate system setting command G92 at that position.

This system can be set as desired though normally the X and Y axes are set so that the workpiece center serves as the zero point and the Z axis is set so that the workpiece end serves as the zero point.



##### Command format

**G92 Xx2 Zz2 αα2 ;**

αα Additional axis



##### Detailed description

- (1) The basic machine coordinate system is shifted by the G92 command, the hypothetical machine coordinate system is created, and at the same time all workpiece coordinate systems 1 to 6 are also shifted.
- (2) When G92 and S or Q are assigned, the spindle clamp rotation speed is set. (Refer to the section on setting the spindle clamp rotation speed.)

## 14. Coordinate System Setting Functions

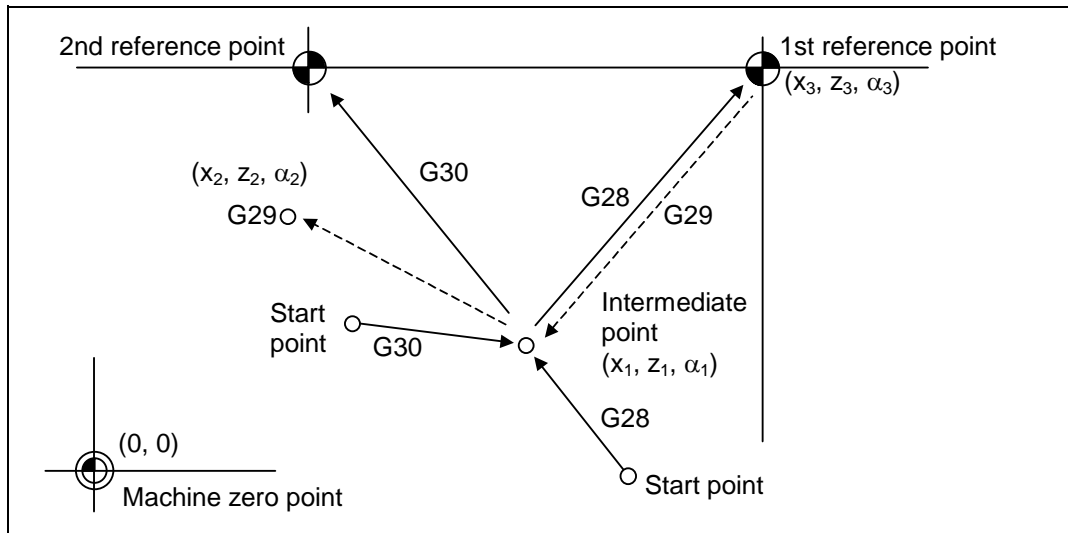
### 14.7 Reference point (Zero point) return

#### 14.7 Reference point (Zero point) return; G28, G29



##### Function and purpose

- (1) After the commanded axes have been positioned by G0, they are returned respectively at rapid traverse to the 1st reference (zero) point when G28 is commanded.
- (2) By commanding G29, the axes are first positioned independently at high speed to the G28 or G30 intermediate point and then positioned by G0 at the commanded position.



##### Command format

**G28**  $Xx_1 Zz_1 \alpha\alpha_1$  ; **Additional axis** Automatic reference point return

**G29**  $Xx_2 Zz_2 \alpha\alpha_2$  ; **Additional axis** Start position return

$\alpha\alpha_1/\alpha\alpha_2$  Additional axis

## 14. Coordinate System Setting Functions

### 14.7 Reference point (Zero point) return



#### Detailed description

- (1) The G28 command is equivalent to the following:

G00 X $x_1$  Z $z_1$   $\alpha_{\alpha_1}$  ;

G00 X $x_3$  Z $z_3$   $\alpha_{\alpha_3}$  ;

In this case,  $x_3$ ,  $z_3$  and  $\alpha_3$  are the reference point coordinates and they are set by parameters "#2037 G53ofs" as the distance from the basic machine coordinate system zero point.

- (2) After the power has been turned ON, the axes which have not been subject to manual reference (zero) point are returned by the dog-type of return just as with the manual type. In this case, the return direction is regarded as the command sign direction. For the second and subsequent returns, the return is made at high speed to the reference (zero) point which was stored at the first time.
- (3) When reference (zero) point return is completed, the zero point arrival output signal is output and also #1 appears at the axis name line on the setting and display unit screen.
- (4) The G29 command is equivalent to the following:

G00 X $x_1$  Z $z_1$   $\alpha_{\alpha_1}$  ; } The rapid traverse (non-interpolation type) independent  
G00 X $x_2$  Z $z_2$   $\alpha_{\alpha_2}$  ; } for each axis takes place.

In this case,  $x_1$ ,  $z_1$  and  $\alpha_1$  are the coordinate value of the G28 or G30 intermediate point.

- (5) Program error (P430) results when G29 is executed when automatic reference (zero) point return (G28) is not performed after the power has been turned ON.
- (6) The intermediate point coordinate value ( $x_1$ ,  $z_1$ ,  $\alpha_1$ ) of the positioning point are assigned by absolute/incremental value commands.
- (7) G29 is valid for either G28 or G30 but the commanded axes are positioned after a return has been made to the latest intermediate point.
- (8) The tool offset will be temporarily canceled during reference point return unless it is already canceled, and the intermediate point will be the offset position.
- (9) The intermediate point can also be ignored by parameter "#1091 Mpoint" setting.
- (10) Control from the intermediate point to the reference (zero) point is ignored for reference (zero) point return in the machine lock status. The next block is executed when the commanded axis survives as far as the intermediate point.
- (11) Mirror image is valid from the start point to the intermediate point during reference (zero) point return in the mirror image mode and the tool will move in the opposite direction to that of the command. However, mirror image is ignored from the intermediate point to the reference (zero) point and the tool will move to the reference (zero) point.

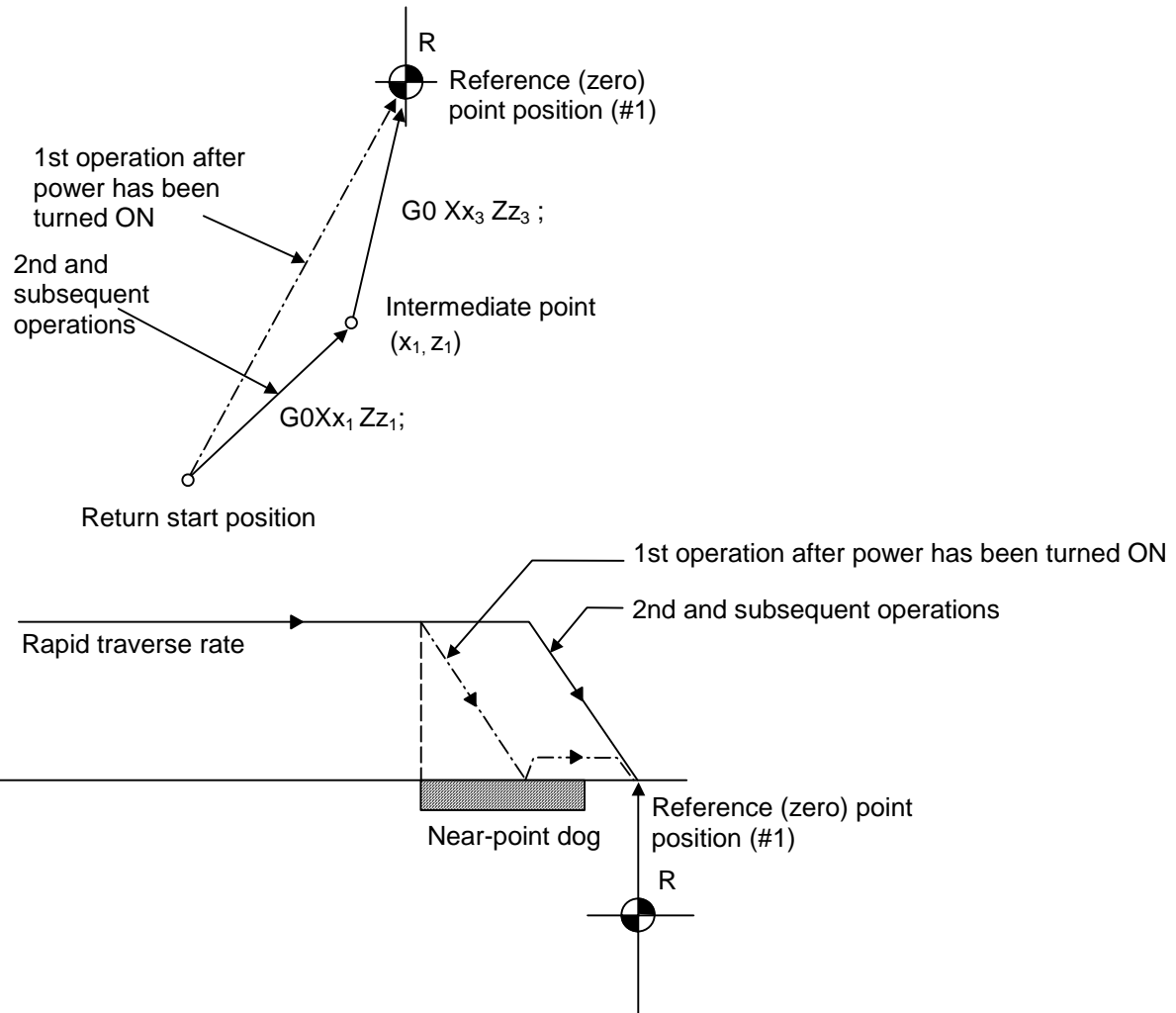
# 14. Coordinate System Setting Functions

## 14.7 Reference point (Zero point) return



### Example of program

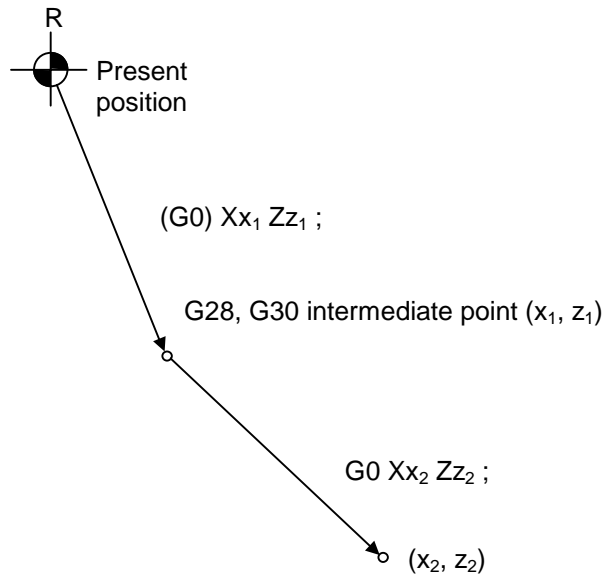
(Example 1) G28 Xx<sub>1</sub> Zz<sub>1</sub> ;



# 14. Coordinate System Setting Functions

## 14.7 Reference point (Zero point) return

(Example 2) G29 Xx<sub>2</sub> Zz<sub>2</sub> ;



(Example 3) G28 Xx<sub>1</sub> Zz<sub>1</sub> ;

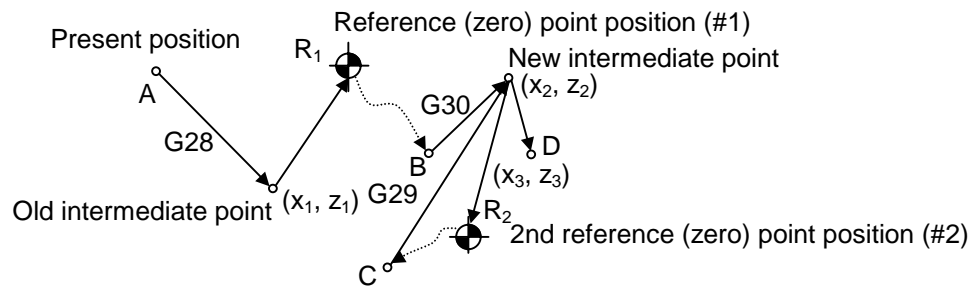
: (From point A to reference (zero) point)

G30 Xx<sub>2</sub> Zz<sub>2</sub> ;

: (From point B to 2nd reference (zero) point)

G29 Xx<sub>3</sub> Zz<sub>3</sub> ;

(From point C to point D)



## 14. Coordinate System Setting Functions

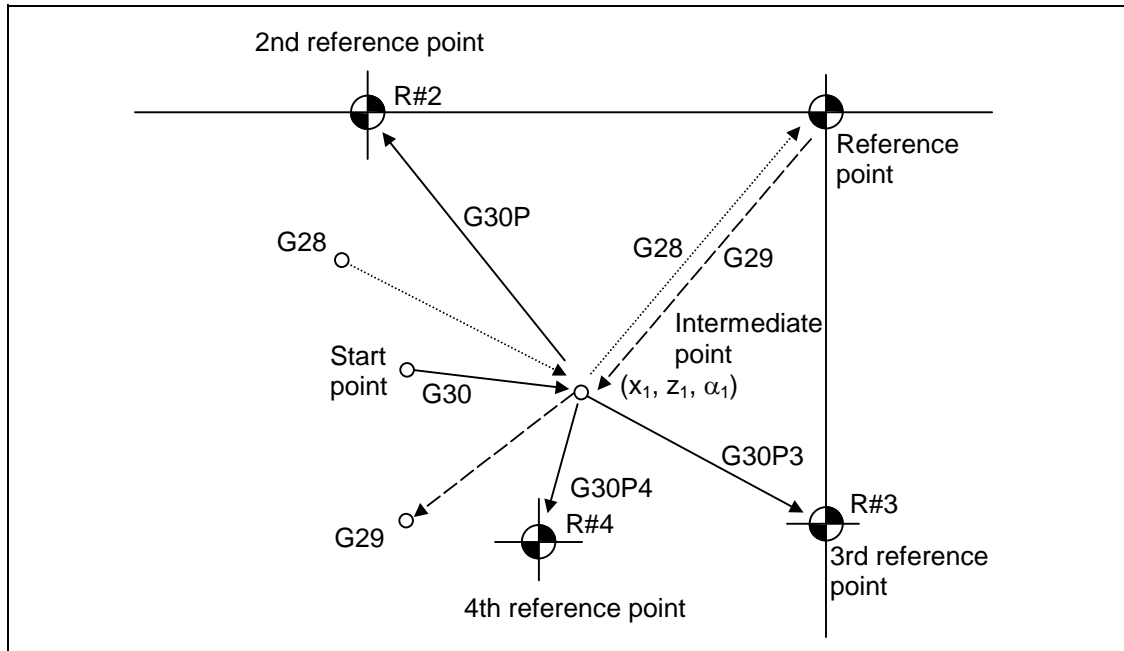
### 14.8 2nd, 3rd, And 4th reference point (Zero point) return

#### 14.8 2nd, 3rd, and 4th reference point (Zero point) return; G30



##### Function and purpose

The tool can return to the 2nd, 3rd, or 4th reference (zero) point by specifying G30 P2 (P3 or P4).



##### Command format

**G30 P2 (P3, P4) Xx<sub>1</sub> Zz<sub>1</sub> αα<sub>1</sub> ;**

αα<sub>1</sub> Additional axis

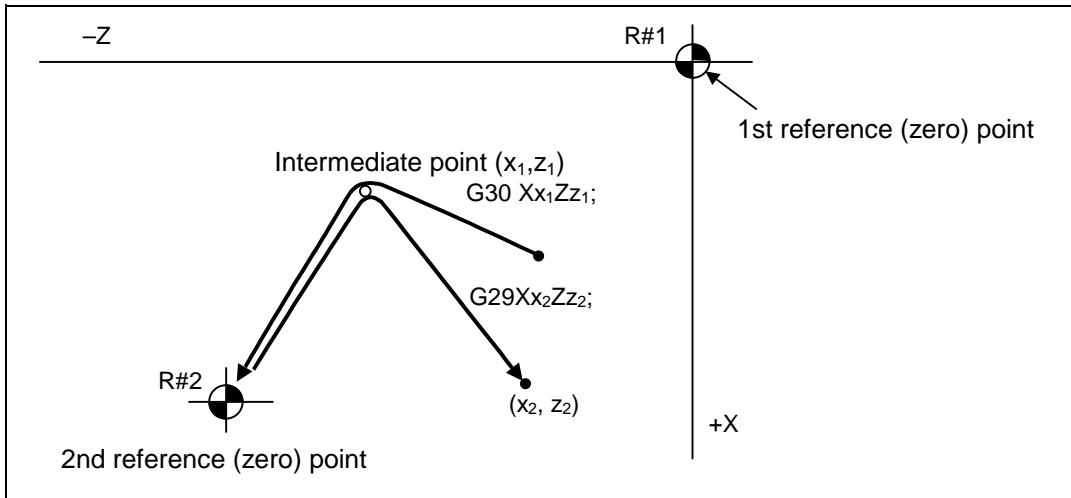
# 14. Coordinate System Setting Functions

## 14.8 2nd, 3rd, And 4th reference point (Zero point) return

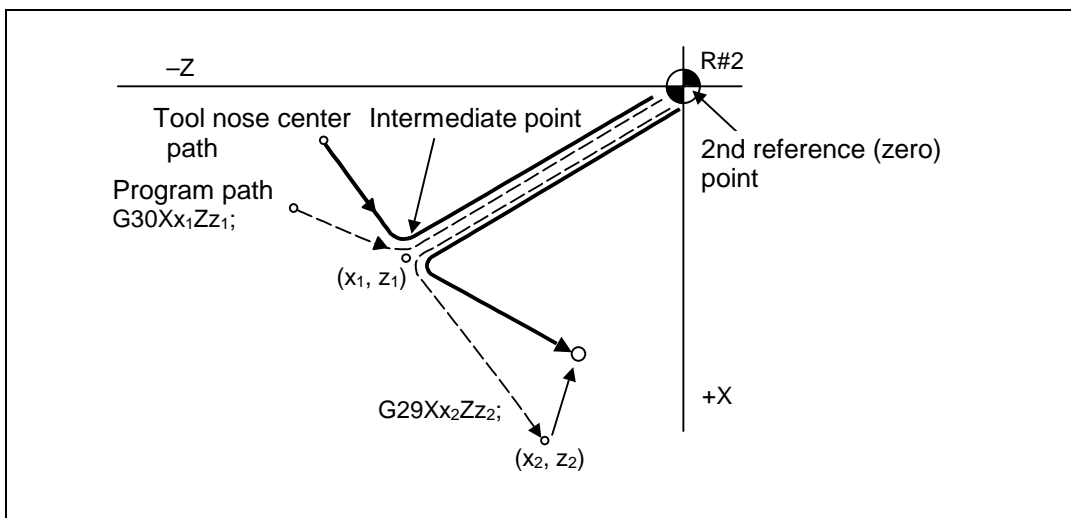


### Detailed description

- (1) The 2nd, 3rd, or 4th reference (zero) point return is specified by P2, P3, or P4. A command without P or with P0, P1, P5 or a greater P number is ignored, returning the tool to the 2nd reference (zero) point.
- (2) In the 2nd, 3rd, or 4th reference (zero) point return mode, as in the 1st reference (zero) point return mode, the tool returns to the 2nd, 3rd, or 4th reference (zero) point via the intermediate point specified by G30.
- (3) The 2nd, 3rd, and 4th reference (zero) point coordinates refer to the positions specific to the machine, and these can be checked with the setting and display unit.
- (4) If G29 is specified after completion of returning to the 2nd, 3rd, and 4th reference (zero) points, the intermediate position used last is used as the intermediate position for returning by G29.



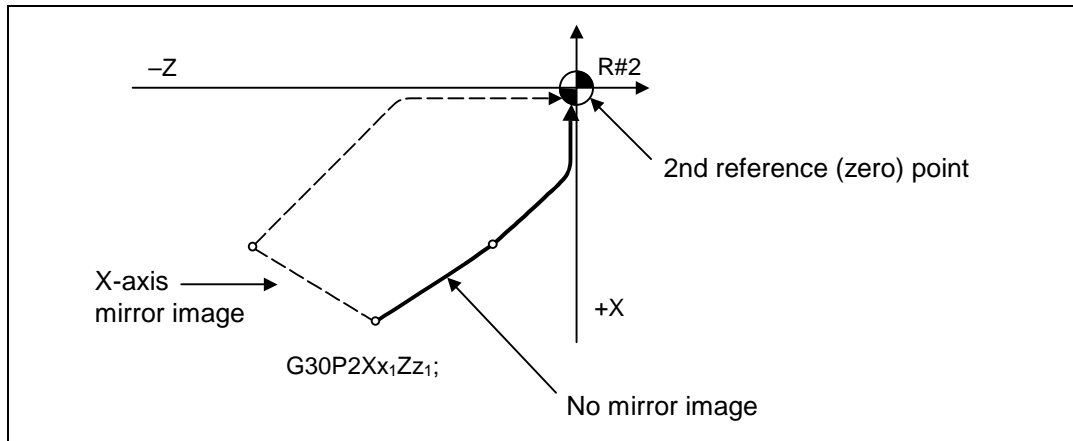
- (5) With reference (zero) point return on a plane during compensation, the tool moves without tool nose radius compensation (zero compensation) from the intermediate point as far as the reference (zero) point. With a subsequent G29 command, the tool move without tool nose radius compensation from the reference (zero) point to the intermediate point and it move with such compensation until the G29 command from the intermediate point.



## 14. Coordinate System Setting Functions

### 14.8 2nd, 3rd, And 4th reference point (Zero point) return

- (6) The tool length offset amount for the axis involved is canceled temporarily after the 2nd, 3rd or 4th reference (zero) point return.
- (7) With 2nd, 3rd or 4th reference (zero) point return in the machine lock status, control from the intermediate point to the reference (zero) point will be ignored. When the designated axis reaches as far as the intermediate point, the next block will be executed.
- (8) With 2nd, 3rd or 4th reference (zero) point return in the mirror image mode mirror image will be valid from the start point to the intermediate point and the tool will move in the opposite direction to that of the command. However, mirror image is ignored from the intermediate point to the reference (zero) point and the tool moves to the reference (zero) point.





# 14. Coordinate System Setting Functions

## 14.9 Simple zero point return

### 14.9 Simple zero point return



#### Function and purpose

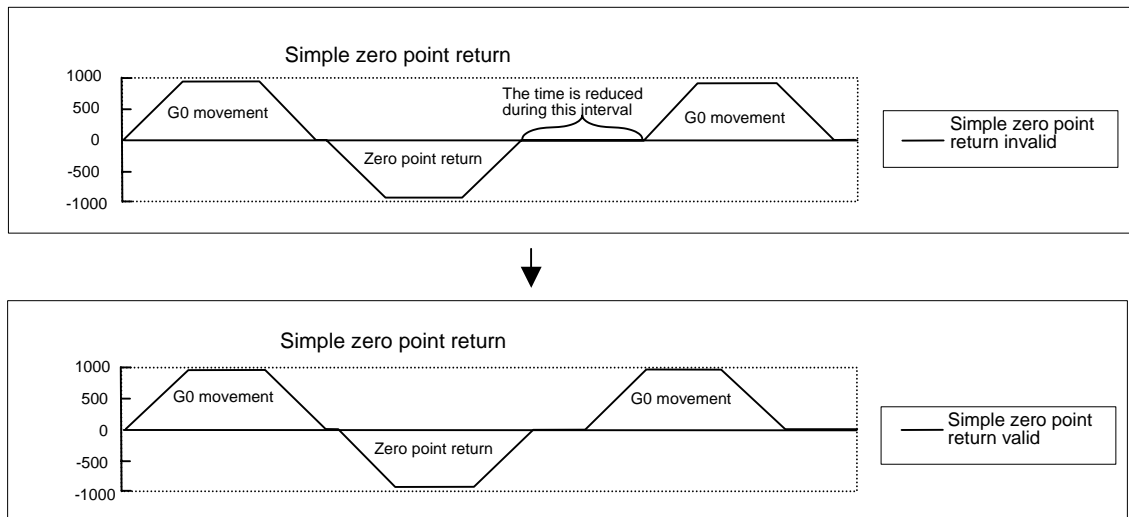
This is the improvement function. That can simplify the G28, G29 and G30 positioning operations and shorten the time by setting the parameter (#1222 aux06/bit7). The time is shortened by approx. several ten ms. When this function is validated, the positioning accuracy may drop. Do not use this function when using an inclined axis.



#### Methods of shortening time

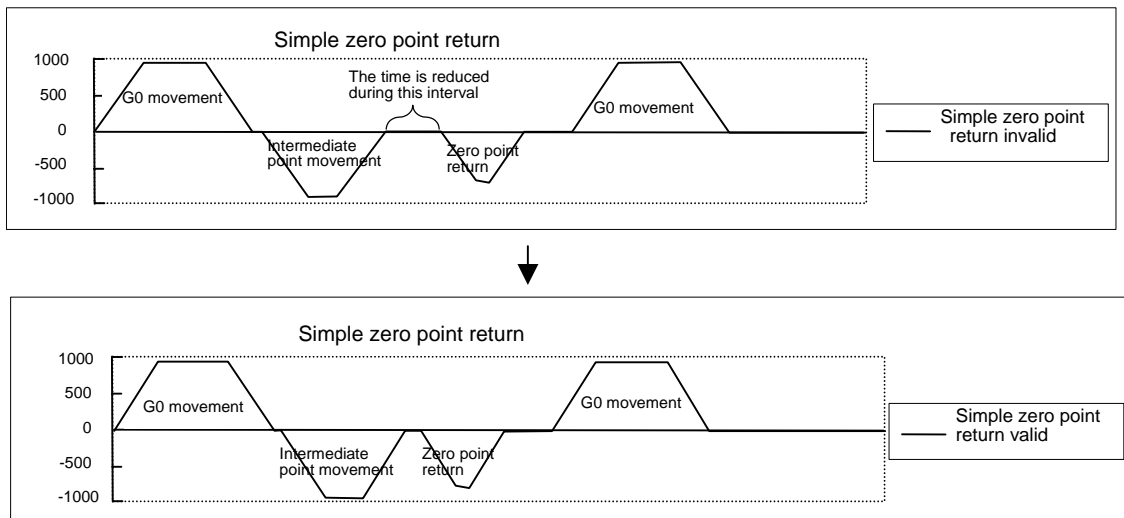
##### (1) G90 without intermediate point

Fig. 1 The graph shows the commanded speed during movement.



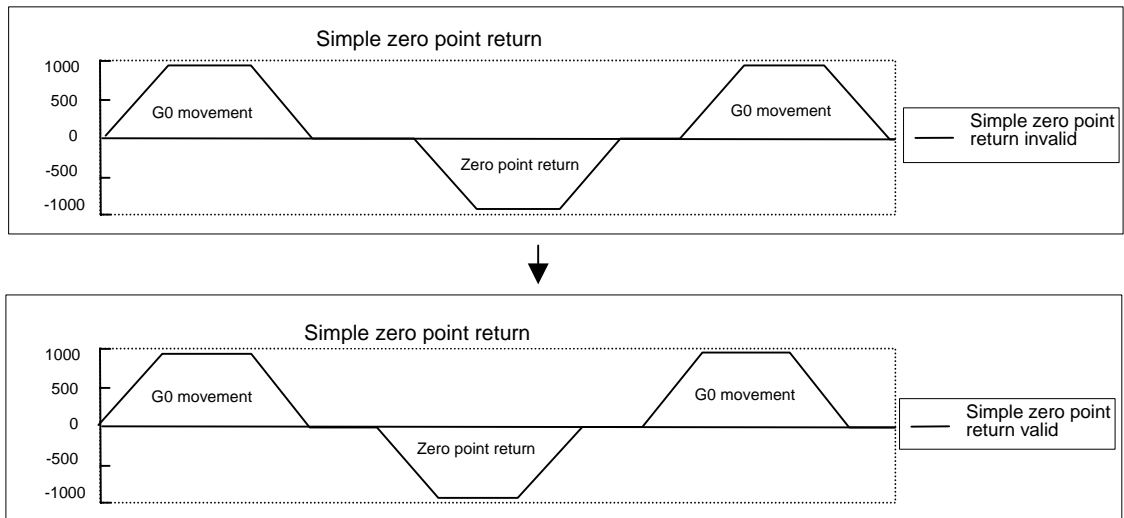
##### (2) G90 with intermediate point

Fig. 2 The graph shows the commanded speed during movement.



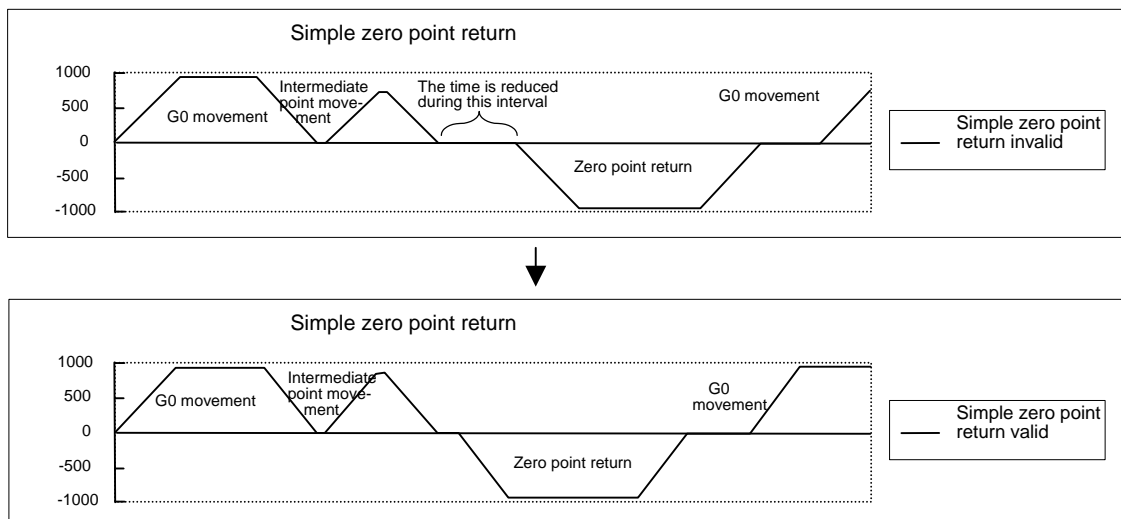
### (3) G91 without intermediate point

**Fig. 3** The graph shows the commanded speed during movement.



### (4) G91 with intermediate point

**Fig. 4** The graph shows the commanded speed during movement.





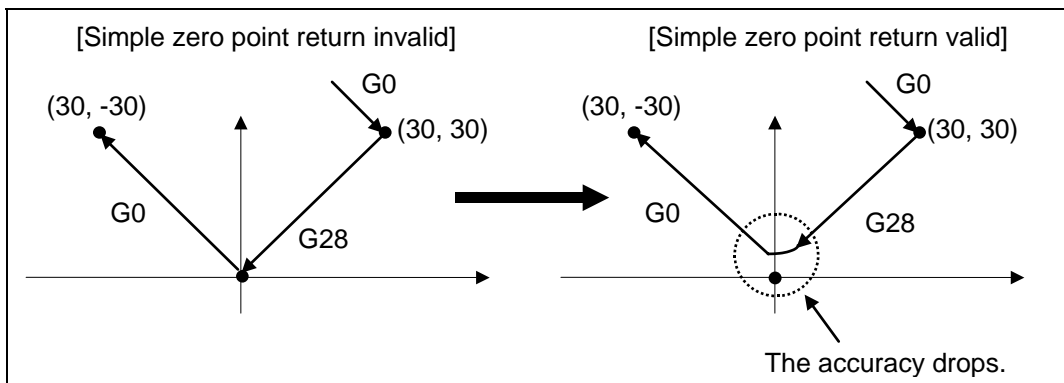
### Accuracy

#### (1) Without intermediate point (Absolute value command)

```
(Program example)
G00 X30.0 Z30.0 ;
G28 XZ ;
G00 X-30.0 Z-30.0 ;
M30 ;
```

The accuracy of the movement before G28 is maintained.

The G28 zero point return accuracy will drop. (The accuracy is affected by the positioning speed and movement distance.)

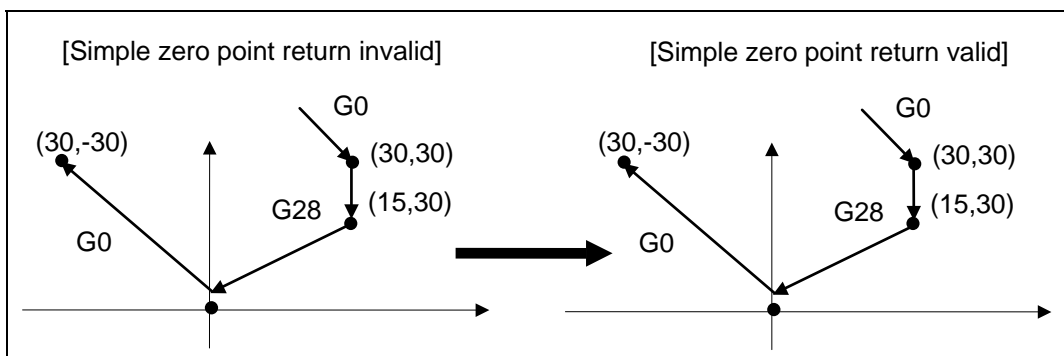


#### (2) With intermediate point (Absolute value command)

```
(Program example)
G00 X30.0 Z30.0 ;
G28 X15.0 Z30.0 ;
G00 X30.0 Z-30.0 ;
M30 ;
```

The accuracy of the movement before G28 is maintained.

The accuracy of the movement to the intermediate point is maintained.



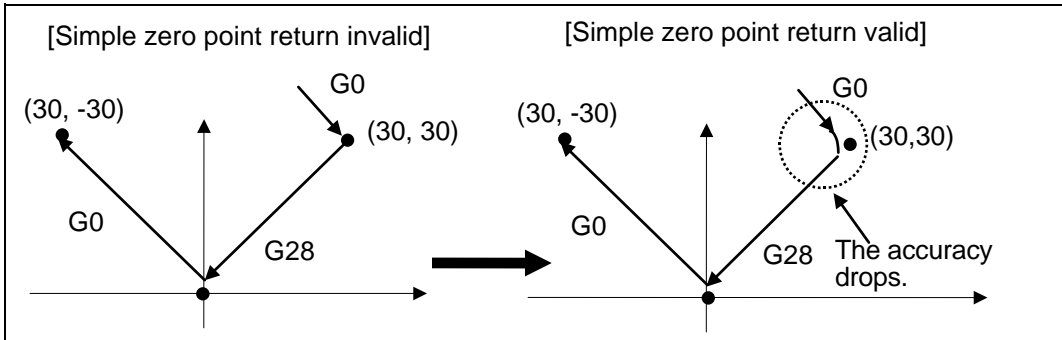
# 14. Coordinate System Setting Functions

## 14.9 Simple zero point return

### (3) Without intermediate point (Incremental value command)

```
(Program example)  
G00 X30.0 Z30.0 ;  
G28 UW ;  
G00 X30.0 Z-30.0 ;  
M30 ;
```

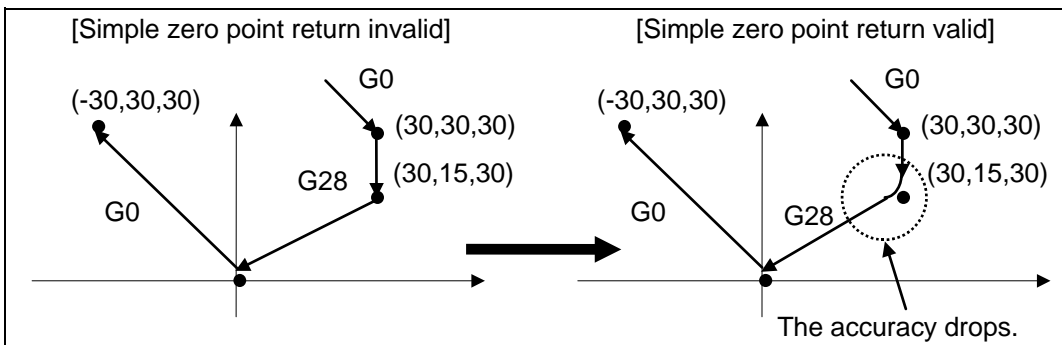
The accuracy of the movement before G28 will drop. (The accuracy is affected by the positioning speed and movement distance.)



### (4) With intermediate point (Incremental value command)

```
(Program example)  
G00 X30.0 Z30.0 ;  
G28 U-15.0 W0.0 ;  
G00 X30.0 Z-30.0 ;  
M30 ;
```

The accuracy of the movement before G28 is maintained.  
The accuracy of the movement to the intermediate point will drop. (The accuracy is affected by the positioning speed and movement distance.)





#### Precautions

- (1) Even when this function's parameter is ON and simple zero point return is validated, if an inclined axis is validated, this function's parameter will be forcibly turned OFF, and this function will be invalidated.
- (2) If this function's parameter is turned ON while an inclined axis is valid, a "setting error" will occur, and this function will be invalidated.
- (3) If parameters which "validate this function" and "validate the inclined axis" are input, this function's parameter will be forcibly turned OFF, and this function will be invalidated.
- (4) When simple zero point return is valid, the accuracy of the rapid traverse command, cutting feed command, and movement command to the intermediate point before G28 may drop. If the rapid traverse command and cutting feed command before G28 must be accurate, use one of the following methods.
  - Invalidate this function.
  - Carry out an in-position check in the block before G28/G29/G30, and confirm the accuracy. (G09: Exact stop check, etc.)
  - Input the required dwell command (time) in the block before G28/G29/G30.

## 14. Coordinate System Setting Functions

### 14.10 Reference point check

#### 14.10 Reference point check; G27



##### Function and purpose

This command first positions the tool at the position assigned by the program and then, if that positioning point is the 1st reference point, it outputs the reference point arrival signal to the machine in the same way as with the G28 command. Therefore, when a machining program is prepared so that the tool will depart from the 1st reference point and return to the 1st reference point, it is possible to check whether the tool has returned to the reference point after the program has been run.



##### Command format

**G27 Xx<sub>1</sub> Zz<sub>1</sub> αα<sub>1</sub> Pp<sub>1</sub> ;**

G27	Check command
Xx <sub>1</sub> Zz <sub>1</sub> αα <sub>1</sub>	Return control axis
Pp <sub>1</sub>	Check number
	P1 : 1st reference point check
	P2 : 2nd reference point check
	P3 : 3rd reference point check
	P4 : 4th reference point check



##### Detailed description

- (1) If the P command has been omitted, the 1st reference point will be checked.
- (2) The number of axes whose reference points can be checked simultaneously depends on the number of axes which can be controlled simultaneously.
- (3) An alarm will occur if the reference point is not reached after the command is completed.

## 14. Coordinate System Setting Functions

### 14.11 Workpiece coordinate system setting and offset

#### 14.11 Workpiece coordinate system setting and offset; G54 ~ G59 (G54.1)



##### Function and purpose

- (1) The workpiece coordinate systems facilitate the programming on the workpiece, serving the basic point of the machining workpiece as the zero point.
- (2) These commands enable the tool to move to the positions in the workpiece coordinate system. There are 6 workpiece coordinate systems which are used by the programmer for programming. (G54 ~ G59)  
In addition to the six sets of workpiece coordinate systems between G54 and G59, there are 48 additional workpiece coordinate system sets. (The 48 sets are options.)
- (3) Among the workpiece coordinate systems currently selected by these commands, any workpiece coordinate system with coordinates which have been commanded by the present position of the tool is reset. (The "present position of the tool" includes the offset amounts for nose radius and tool length.)
- (4) An hypothetical machine coordinate system with coordinates which have been commanded by the present position of the tool is set by this command. (The "present position of the tool" includes the offset amounts for nose radius, tool length.) (G54 ~ G92)



##### Command format

#### (1) Workpiece coordinate system selection (G54 ~ G59)

G54 Xx<sub>1</sub> Zz<sub>1</sub> α<sub>α1</sub> ;

α<sub>α1</sub> Additional axis

#### (2) Workpiece coordinate system setting (G54 ~ G59)

(G54) G92 Xx<sub>1</sub> Zz<sub>1</sub> α<sub>α1</sub> ;

α<sub>α1</sub> Additional axis

#### (3) Workpiece coordinate system selection (P1 ~ P48)

G54.1 Pn ;

#### (4) Workpiece coordinate system setting (P1 ~ P48)

G54.1 Pn ;

G92 Xx Zz ;

#### (5) Workpiece coordinate system offset amount setting (P1 ~ P48)

G10 L20 Pn Xx Zz ;

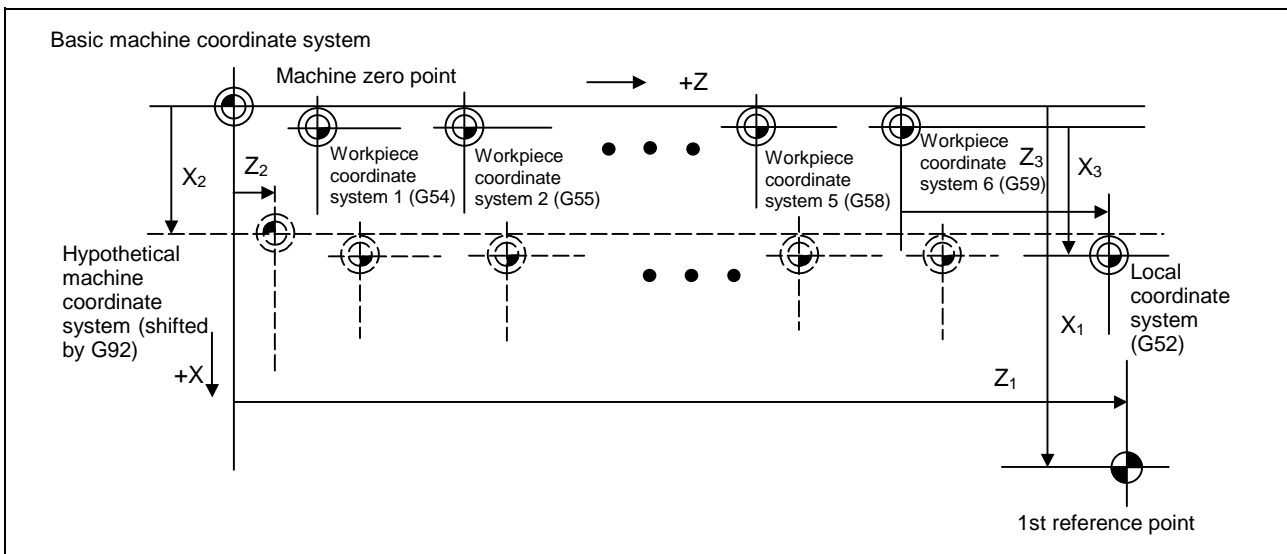
# 14. Coordinate System Setting Functions

## 14.11 Workpiece coordinate system setting and offset



### Detailed description

- (1) With any of the G54 to G59 commands, the nose radius offset amounts for the commanded axes will not be canceled even if workpiece coordinate system selection is commanded.
- (2) The G54 workpiece coordinate system is selected when the power is turned ON.
- (3) Commands G54 to G59 are modal commands (group 12).
- (4) The coordinate system will move with G92 in a workpiece coordinate system.
- (5) The offset setting in a workpiece coordinate system denotes the distance from the basic machine coordinate system zero point.



- (6) The offset settings of workpiece coordinate systems can be changed any number of times. (They can also be changed by G10 L2 Pp<sub>1</sub> Xx<sub>1</sub> Zz<sub>1</sub>.)

#### Handling when L or P is omitted

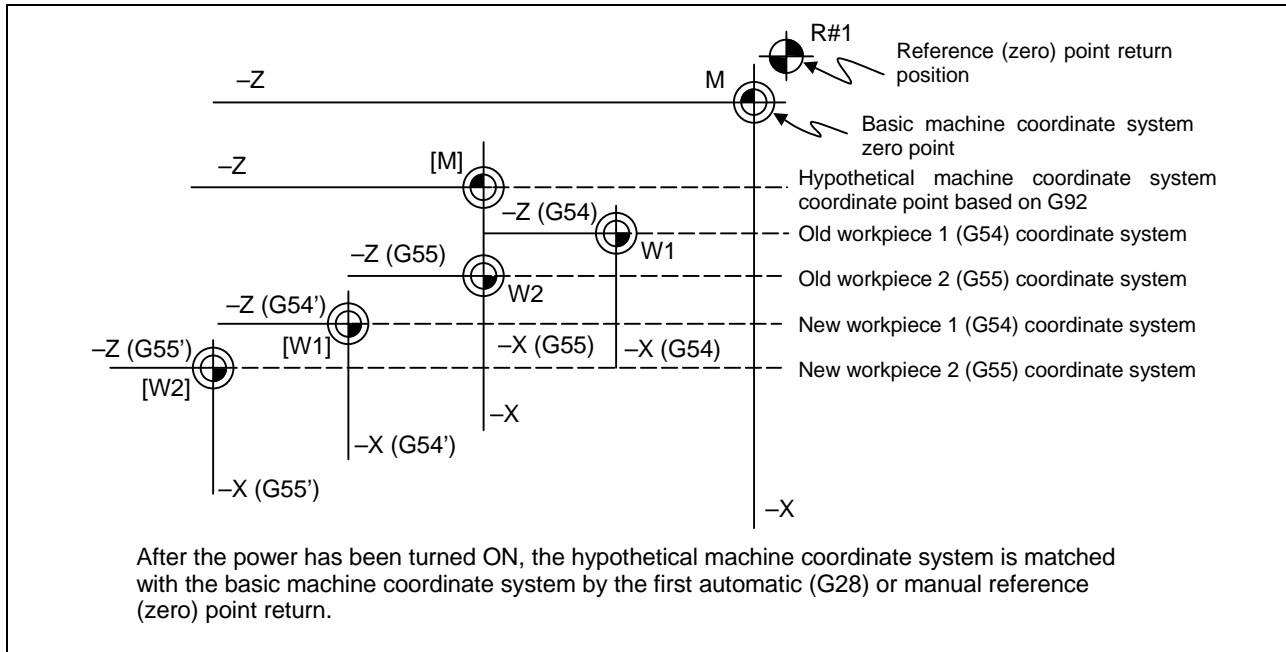
G10 L2 Pn Xx Zz ;	n=0	: Set the offset amount in the external workpiece coordinate system.
	n=1 to 6	: Set the offset amount in the designated workpiece coordinate system.
	Others	: The program error (P35) will occur.
G10 L2 Xx Zz ;		Set the offset amount in the currently selected workpiece coordinate system. When in G54.1 modal, the program error (P33) will occur.
G10 L20 Pn Xx Zz ;	n=1 to 48	: Set the offset amount in the designated workpiece coordinate system.
	Others	: The program error (P35) will occur.
G10 L20 Xx Zz ;		Set the offset amount in the currently selected workpiece coordinate system. When in G54 to G59 modal, the program error (P33) will occur.
G10 Pn Xx Zz ;	}	L10 (tool offset) will be judged if there is no L value.
G10 Xx Zz ;		



## 14. Coordinate System Setting Functions

### 14.11 Workpiece coordinate system setting and offset

- (7) A new workpiece coordinate system 1 is set by issuing the G92 command in the G54 (workpiece coordinate system 1) mode. At the same time, the other workpiece coordinate systems 2 to 6 (G55 ~ G59) will move in parallel and new workpiece coordinate systems 2 to 6 will be set.
- (8) A hypothetical machine coordinate system is formed at the position which deviates from the new workpiece reference (zero) point by an amount equivalent to the workpiece coordinate system offset amount.



- (9) By setting the hypothetical machine coordinate system, the new workpiece coordinate system will be set at a position which deviates from that hypothetical machine coordinate system by an amount equivalent to the workpiece coordinate system offset.
- (10) When the first automatic (G28) or manual reference (zero) point return is completed after the power has been turned ON, the basic machine coordinate system and workpiece coordinate systems are set automatically in accordance with the parameter settings.
- (11) If G54 X- ; is commanded after the reference point return (both automatic or manual) executed after the power is turned ON, the program error (P62) will occur. (A speed command is required as the movement will be controlled with the G01 speed.)
- (12) Do not command a G code for which a P code is used in the same block as G54.1. The P code will be used in the prioritized G command.
- (13) When number of workpiece offset sets additional specifications is not added, the program error (P39) will occur when the G54.1 command is executed.

## 14. Coordinate System Setting Functions

### 14.11 Workpiece coordinate system setting and offset

- (14) When number of workpiece offset sets additional specifications is not added, the program error (P172) will occur when the G10 L20 command is executed.
- (15) The local coordinate system cannot be used during G54.1 modal. The program error (P438) will occur when the G52 command is executed during G54.1 modal.
- (16) A new workpiece coordinate system P1 can be set by commanding G92 in the G54.1 P1 mode. However, the workpiece coordinate system of the other workpiece coordinate systems G54 to G59, G54.1, and P2 to P48 will move in parallel with it, and a new workpiece coordinate system will be set.
- (17) The offset amount of the extended workpiece coordinate system is assigned to the variable number as shown in Table 1.

**Table 1 Variable numbers of the extended workpiece coordinate offset system**

	1st axis to nth axis		1st axis to nth axis
P1	#7001 to #700n	P25	#7481 to #748n
P2	#7021 to #702n	P26	#7501 to #750n
P3	#7041 to #704n	P27	#7521 to #752n
P4	#7061 to #706n	P28	#7541 to #754n
P5	#7081 to #708n	P29	#7561 to #756n
P6	#7101 to #710n	P30	#7581 to #758n
P7	#7121 to #712n	P31	#7601 to #760n
P8	#7141 to #714n	P32	#7621 to #762n
P9	#7161 to #716n	P33	#7641 to #764n
P10	#7181 to #718n	P34	#7661 to #766n
P11	#7201 to #720n	P35	#7681 to #768n
P12	#7221 to #722n	P36	#7701 to #770n
P13	#7241 to #724n	P37	#7721 to #772n
P14	#7261 to #726n	P38	#7741 to #774n
P15	#7281 to #728n	P39	#7761 to #776n
P16	#7301 to #730n	P40	#7781 to #778n
P17	#7321 to #732n	P41	#7801 to #780n
P18	#7341 to #734n	P42	#7821 to #782n
P19	#7361 to #736n	P43	#7841 to #784n
P20	#7381 to #738n	P44	#7861 to #786n
P21	#7401 to #740n	P45	#7881 to #788n
P22	#7421 to #742n	P46	#7901 to #790n
P23	#7441 to #744n	P47	#7921 to #792n
P24	#7461 to #746n	P48	#7941 to #794n

#### CAUTION



If the workpiece coordinate system offset amount is changed during single block stop, the new setting will be valid from the next block.

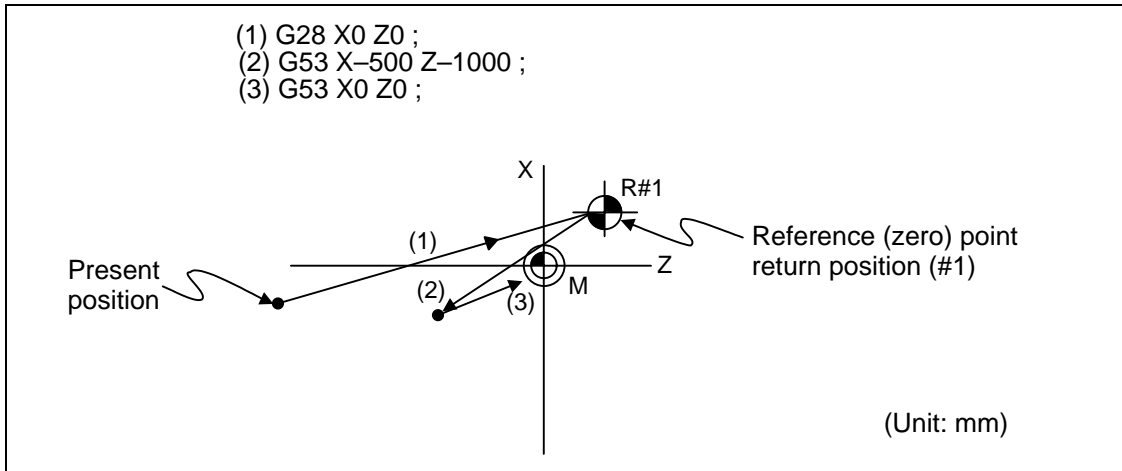
# 14. Coordinate System Setting Functions

## 14.11 Workpiece coordinate system setting and offset



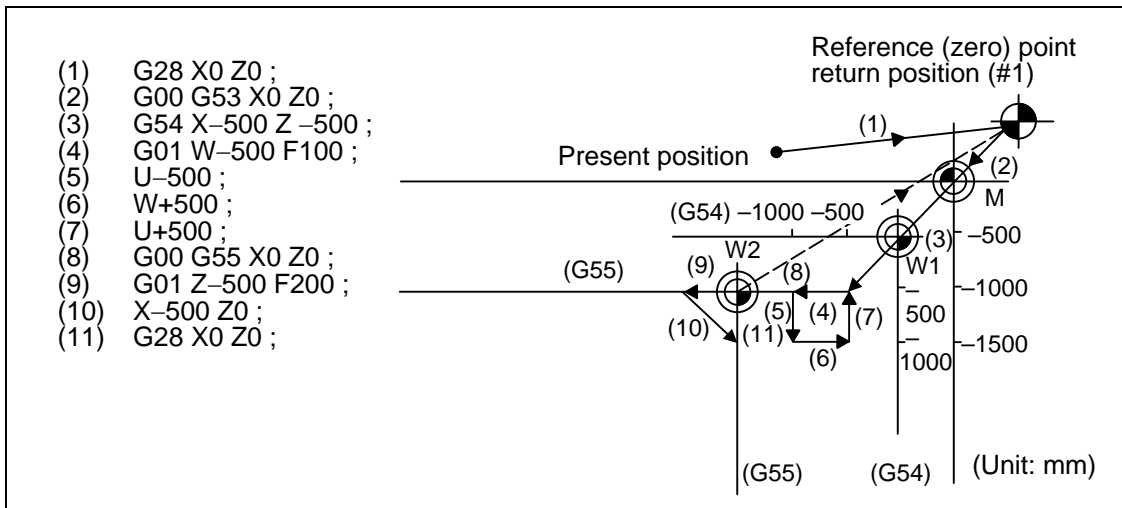
### Example of program

#### (Example 1)



When the 1st reference point coordinate is zero, the basic machine coordinate system zero point and reference (zero) point return position (#1) will coincide.

#### (Example 2)

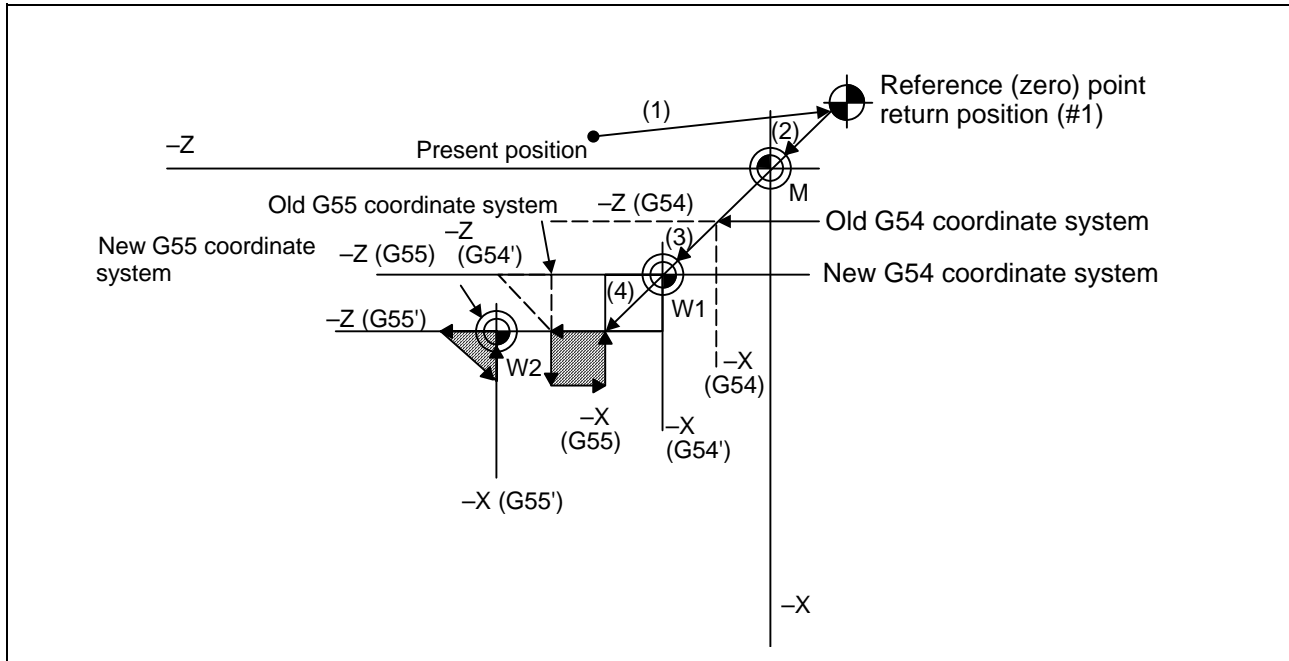


## 14. Coordinate System Setting Functions

### 14.11 Workpiece coordinate system setting and offset

**(Example 3)** When workpiece coordinate system G54 has deviated (-500, -500) in Example 2 (It is assumed that (3) to (10) in Example 2 have been entered in subprogram O1111.)

(1) G28 X0 Z0 ;	
(2) G00 G53 X0 Z0 ;	(Not required when there is no basic machine coordinate system offset.)
(3) G54 X-500 Z-500 ;	Amount by which workpiece coordinate system deviates
(4) G92 X0 Z0 ;	New workpiece coordinate system is set.
(5) M98 P1111 ;	



**(Note)** The workpiece coordinate system will deviate each time steps (3) to (5) are repeated. The reference point return (G28) command should therefore be issued upon completion of the program.

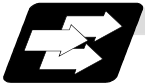
#### **CAUTION**

**!** If the workpiece coordinate system offset amount is changed during single block stop, the changes will be valid from the next block.

## 14. Coordinate System Setting Functions

### 14.12 Local coordinate system setting

#### 14.12 Local coordinate system setting; G52



##### Function and purpose

The local coordinate systems can be set independently on the G54 ~ G59 workpiece coordinate systems using the G52 command so that the commanded position serves as the programmed zero point.

The G52 command can also be used instead of the G92 command to change the deviation between the zero point in the machining program and the machining workpiece zero point.



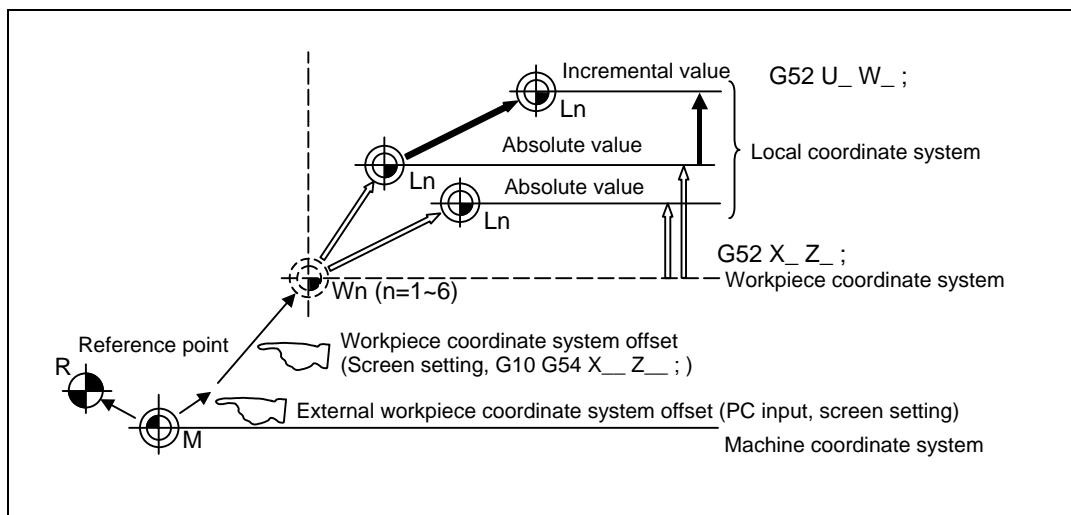
##### Command format

G54 (G54 to G59) G52 Xx1 Zz1 ;



##### Detailed description

- (1) The G52 command is valid until a new G52 command is issued and the tool does not move. This command comes in handy for employing another coordinate system without changing the zero point positions of the workpiece coordinate systems (G54 ~ G59).
- (2) The local coordinate system offset is cleared by the reference point (zero point) return or dog-type manual reference (zero) point return after the power is turned ON.
- (3) The local coordinate system is canceled by (G54 ~ G59) G52 X0 Z0 ;.
- (4) Coordinate commands in the absolute value mode cause the tool to move to the local coordinate system position.



**(Note)** If the program is executed repeatedly, the workpiece coordinate system will deviate each time. Thus, when the program is completed, the reference point return operation must be commanded.

## 14. Coordinate System Setting Functions

### 14.13 Workpiece Coordinate System Preset

#### 14.13 Workpiece Coordinate System Preset; G92.1

[E68]



#### Function and purpose

- (1) This function presets the workpiece coordinate system shifted with the program command during manual operation to the workpiece coordinate system offset from the machine zero point by the workpiece coordinate offset amount by the program command (G92.1).
- (2) The set workpiece coordinate system will be shifted from the machine coordinate system when the following type of operation or program command is executed.
  - When manual interrupt is executed while manual absolute is OFF
  - When movement command is issued in machine lock state
  - When axis is moved with handle interrupt
  - When operation is started with mirror image
  - When local coordinate system is set with G52
  - When workpiece coordinate system is shifted with G92

This function presets the shifted workpiece coordinate system to the workpiece coordinate system offset from the machine zero point by the workpiece coordinate offset amount. This takes place in the same manner as manual reference point return. Whether to preset the relative coordinates or not can be selected with the parameters.



#### Command format

<b>G92.1</b> X0 Y0 Z0 $\alpha$ 0 ; <b>(G50.3)</b> $\alpha$ 0 Additional axis
------------------------------------------------------------------------------------

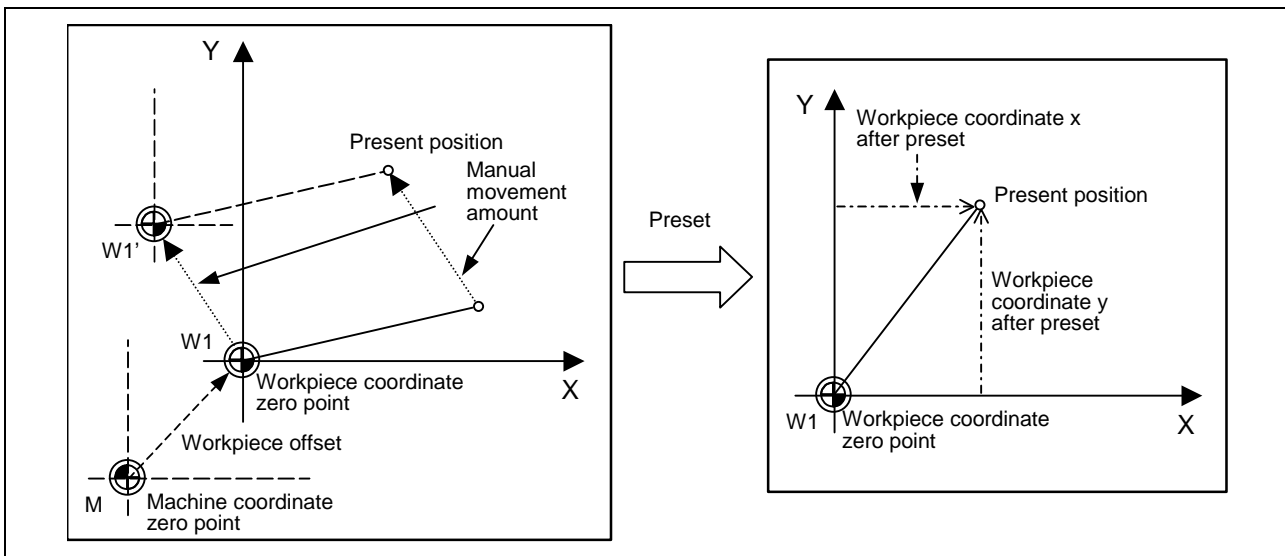
## 14. Coordinate System Setting Functions

### 14.13 Workpiece Coordinate System Preset



#### Detailed description

- (1) Command the address of the axis to be preset. The axis will not be preset unless commanded.
- (2) A program error (P35) will occur if a value other than "0" is commanded.
- (3) This can be commanded in the following G code systems.
  - G code system 2, 3, 6, 7
- (4) Depending on the G code system, the G code will be "G50.3".
- (5) When manual operation is carried out when manual absolute is set to OFF, or if the axis is moved with handle interrupt

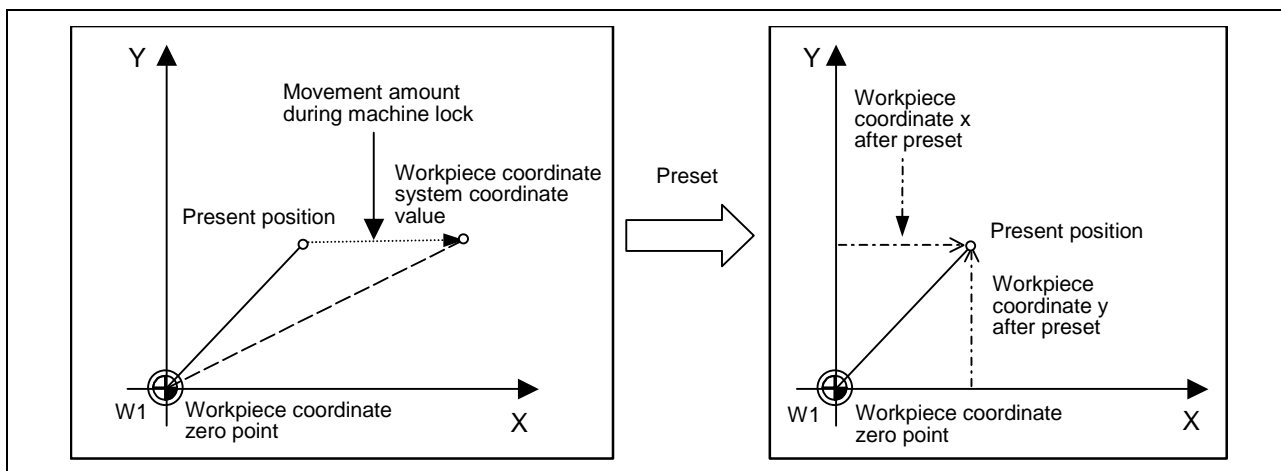


If manual operation is carried out when manual absolute is set to OFF, or if the axis is moved with handle interrupt, the workpiece coordinate system will be shifted by the manual movement amount. This function returns the shifted workpiece coordinate zero point W1' to the original workpiece coordinate zero point W1, and sets the distance from W1 to the present position as the workpiece coordinate system's present position.

## 14. Coordinate System Setting Functions

### 14.13 Workpiece Coordinate System Preset

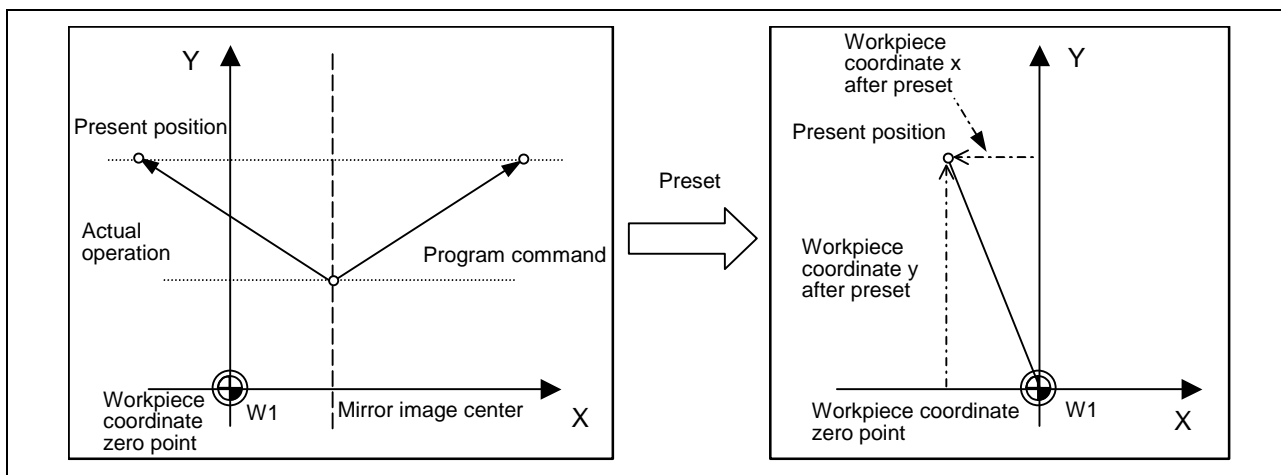
(6) When movement command is issued in machine lock state



If the movement command is issued in the machine lock state, the present position will not move, and only the workpiece coordinates will move.

This function returns the moved workpiece coordinates to the original present position, and sets the distance from W1 to the present position as the workpiece coordinate system's present position.

(7) When operation is carried out with mirror image



If operation is carried out with mirror image, only the NC internal coordinates are used as the program command coordinates. The other coordinates are the present position coordinates.

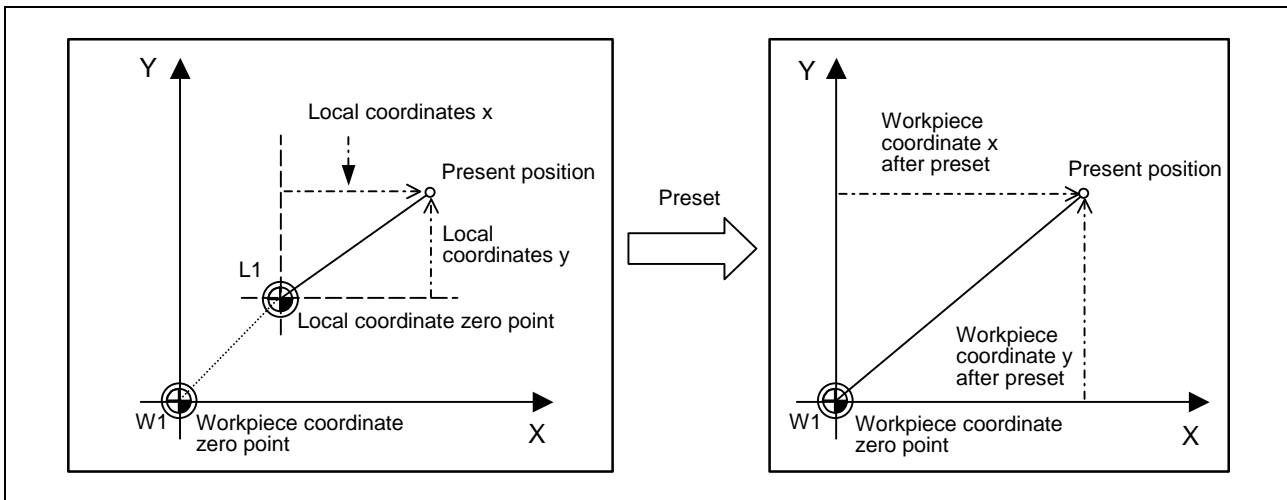
This function sets the NC internal coordinates as the present position coordinates.



## 14. Coordinate System Setting Functions

### 14.13 Workpiece Coordinate System Preset

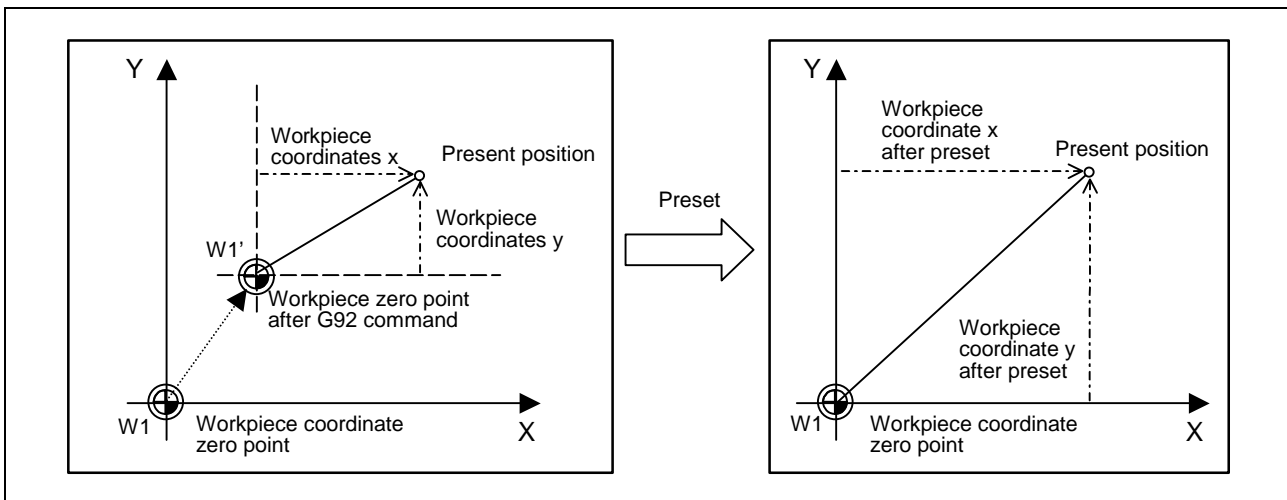
#### (8) Setting local coordinate system with G52



The local coordinate system is set with the G52 command, and the program commands, etc., are issued with the local coordinate system.

With this function, the set local coordinate system is canceled, and the program commands, etc., use the workpiece coordinate system which has W1 as the zero point. The canceled local coordinate system is only the selected workpiece coordinate system.

#### (9) Shifting the workpiece coordinate system with G92



The workpiece coordinate system shifts with the G92 command, and the distance between W1' and the present position is set as the present position of the workpiece coordinate system.

This function returns the shifted workpiece coordinate zero point to W1, and sets the distance from W1 to the present position as the workpiece coordinate system's present position. This is valid for all workpiece coordinate systems.

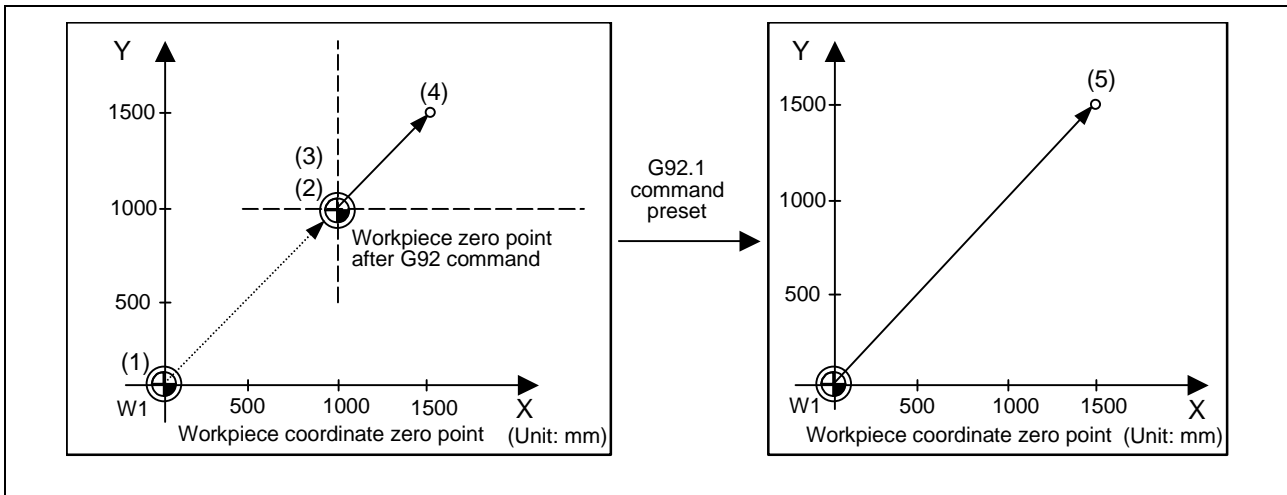
## 14. Coordinate System Setting Functions

### 14.13 Workpiece Coordinate System Preset



#### Example of program

The workpiece coordinate system shifted with G92 is preset with G92.1.



#### (Example)

```
G28 X0 Y0 ;..... (1)
G00 G90 X1. Y1. ;..... (2)
G92 X0 Y0 ;..... (3)
G00 X500 Y500 ;..... (4)
G92.1 X0 Y0 ;..... (5)
```



#### Precautions

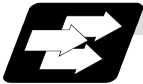
- (1) Cancel tool radius compensation, tool length offset and tool position offset before using this function. If these are not canceled, the workpiece coordinates will be at a position obtained by subtracting the workpiece coordinate offset amount from the machine value. Thus, the compensation vector will be temporarily canceled.
- (2) This function cannot be executed while the program is being resumed.
- (3) Do not command this function during the scaling, coordinate rotation or program mirror image modes.  
A program error (P34) will occur if commanded.

# 14. Coordinate System Setting Functions

## 14.14 Coordinate system for rotary axis

### 14.14 Coordinate system for rotary axis

[E68]



#### Function and purpose

The axis designated as the rotary axis with the parameters is controlled with the rotary axis' coordinate system.

The rotary axis includes the rotating type (short-cut valid/invalid) and linear type.

The range of each coordinate system is 0 to 359.999° for the rotating type, and 0 to ±99999.999° for the linear type.

The rotary axis is commanded with a degree (°) unit regardless of the inch or metric designation.

The rotary axis type is common for all rotary axes and can be set with the parameters "#1089 Cut\_RT" and "#1090 Lin\_RT".

	Rotary axis			
	Rotating type rotary axis		Linear type rotary axis	Linear axis
	Short-cut invalid	Short-cut valid		
#1089 setting value	0	1	--	--
#1090 setting value	0		1	--
Workpiece coordinate value	Displayed in the range of 0° to 359.999°.		Displayed in the range of 0° to ±99999.999°.	
Machine coordinate value/relative position	Displayed in the range of 0° to 359.999°.		Displayed in the range of 0° to ±99999.999°.	
ABS command	The incremental amount from the end point to the current position is divided by 360, and the axis moves by the remainder amount according to the sign.	The incremental amount from the end point to the current position is divided by 360°, and the axis takes a shortcut to moves by the remainder amount.	In the same manner as the normal linear axis, moves according to the sign by the amount obtained by subtracting the current position from the end point (without rounding up to 360 degrees).	
INC command	Moves in the direction of the commanded sign by the commanded incremental amount starting at the current position.			
Reference position return	Until the intermediate point: Depends on the absolute command or the incremental command.			
	From the intermediate point to the reference position: Returns with movement within 360 degrees.		Moves and returns in the reference position direction by the difference from the intermediate point to the reference position.	

# 14. Coordinate System Setting Functions

## 14.14 Coordinate system for rotary axis



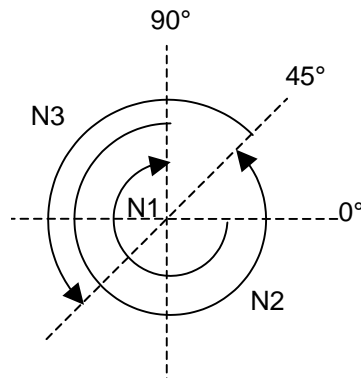
### Example of operation

Examples of differences in the operation and counter displays according to the type of rotation coordinate are given below.

(The workpiece offset is set as 0°.)

#### (1) Rotary type (#1090 = 0) + short-cut invalid (#1089 = 0)

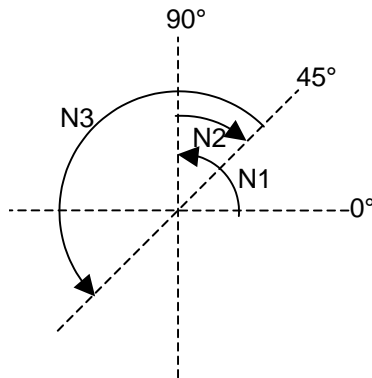
- (a) The machine coordinate position, workpiece coordinate position and current position are displayed in the range of 0 to 359.999°.
- (b) For the absolute command, the axis moves according to the sign.



Program	Workpiece coordinate counter	Machine coordinate counter
G28 C0.		
N1 G90 C-270.	90.000	90.000
N2 C405.	45.000	45.000
N3 G91 C180	225.000	225.000

#### (2) Rotation type (When #1090 = 0) + short-cut valid (#1089 = 1)

- (a) The machine coordinate position, workpiece coordinate position and current position are displayed in the range of 0 to 359.999°.
- (b) For the absolute command, the axis rotates to the direction having less amount of movement to the end point.



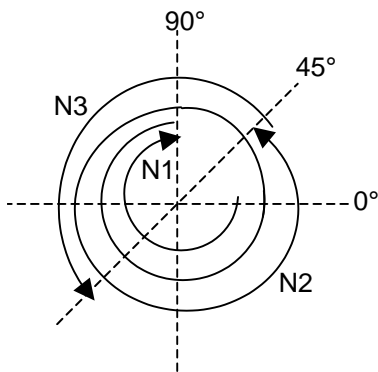
Program	Workpiece coordinate counter	Machine coordinate counter
G28 C0.		
N1 G90 C-270.	90.000	90.000
N2 C405.	45.000	45.000
N3 G91 C180	225.000	225.000

# 14. Coordinate System Setting Functions

## 14.14 Coordinate system for rotary axis

### (3) Linear type (When #1090 = 1)

- (a) The machine coordinate position, workpiece coordinate position and current position are displayed in the range of 0 to 99999.999°.
- (b) For absolute value commands that exceed 360°, the axis will move in the same manner as a linear axis.  
**(Example)** If the command is 420°, the axis will pass the 360° position and will move to the 60° position.
- (c) During reference position return, the axis moves in the same manner as the linear axis until the intermediate point.  
 The axis returns with a rotation within 360° from the intermediate point to the reference position.
- (d) During absolute position detection, the system starts up at the position where the power was turned OFF when the power is turned ON again.



Program	Workpiece coordinate counter	Machine coordinate counter	POSITION counter
G28 C0.			
N1 G90 C-270.	-270.000	-270.000	-270.000
N2 C405.	405.000	405.000	405.000
N3 G91 C180	585.000	585.000	585.000

When power is turned ON again ↓

Workpiece	Machine
585.000	585.000

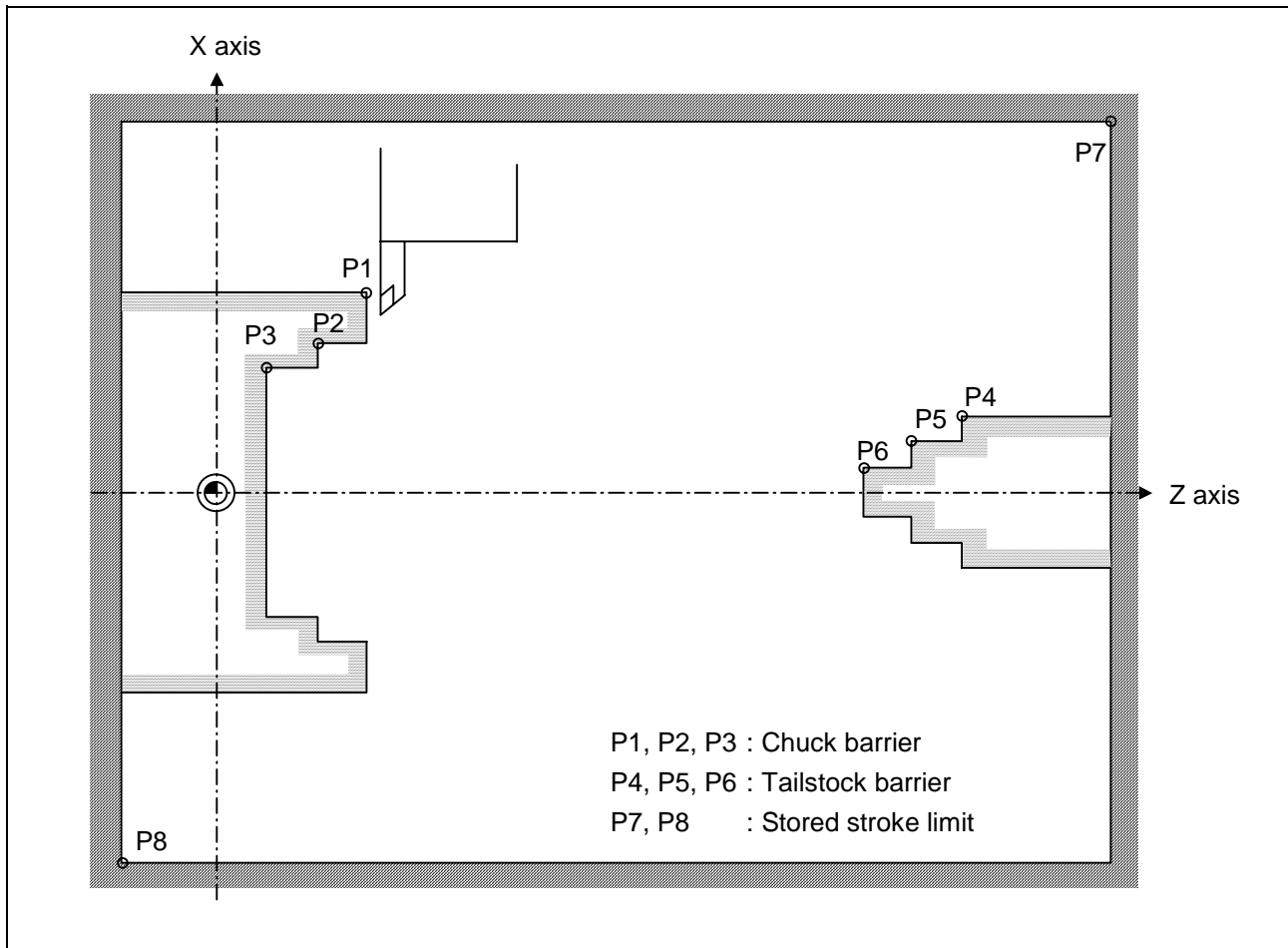
### 15. Protection Function

#### 15.1 Chuck barrier/tailstock barrier; G22, G23



#### Function and purpose

By limiting the tool nose movement range, the chuck barrier and tailstock barrier prevent collision with the chuck and tailstock due to programming errors. If movement is commanded which exceeds the region set in the parameters, the tool will automatically stop at the barrier boundary.



#### Command format

G22;	Barriers valid
G23;	Barriers invalid

Set G22 and G23 commands in independent blocks.

## 15. Protection Function

### 15.1 Chuck barrier/tailstock barrier



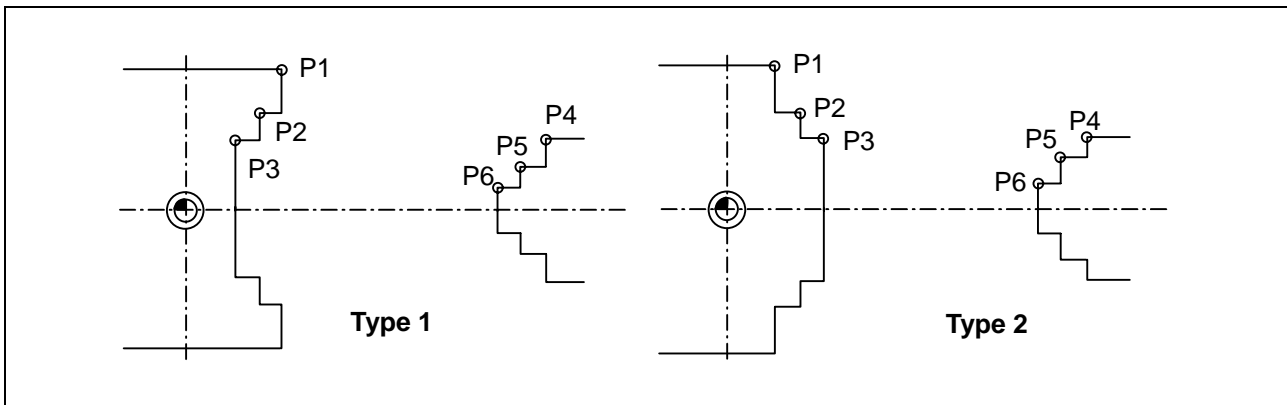
#### Detailed description

- (1) An alarm will appear if the machine stops because it was about to exceed the set region. Reset to cancel this alarm.
- (2) This function is also valid during machine lock.
- (3) This function is validated when all axes in which chuck barrier and tailstock barrier are set have finished their reference point returns.
- (4) When there is a stored stroke check function, and the stored stroke limit region is set, the chuck barrier/tailstock barrier function is validated simultaneously with the stored stroke check function.
- (5) A detailed barrier can be set on the left and right by using the PLC external signal.



#### Setting the chuck barrier/tailstock barrier

##### (1) Using G22 and G23



- (a) Three points can be input as parameters for both the chuck barrier and tailstock barrier. Set in the machine coordinate system.  
Points P1, P2 and P3 are for the chuck barrier. Points P4, P5 and P6 are for the tailstock barrier.
- (b) The barrier region should be a symmetric shape regarding the Z axis. When the X axis coordinates of barrier point P\_ are a negative value, reverse the sign to the positive side, then convert and check.  
The absolute value of each barrier point's X axis coordinates must be set as follows.  
 $P1 \geq P2 \geq P3, P4 \geq P5 \geq P6$   
(Note that the Z axis coordinates do not have to follow this setting.)

### (2) Using the PLC external signal input

Set the coordinate values for each point between "#8300 P0" to "#8314 P10".

P0 is the basic X coordinate of the chuck and tailstock barrier. Set the workpiece center coordinate in the basic machine coordinate system as a radius value.

The coordinate value from the workpiece center (P0) is set as a radius value for the P1 to P10 X axes. The Z axis is set as a basic machine coordinate system coordinate.

The barrier region is symmetrical in respect to P0.

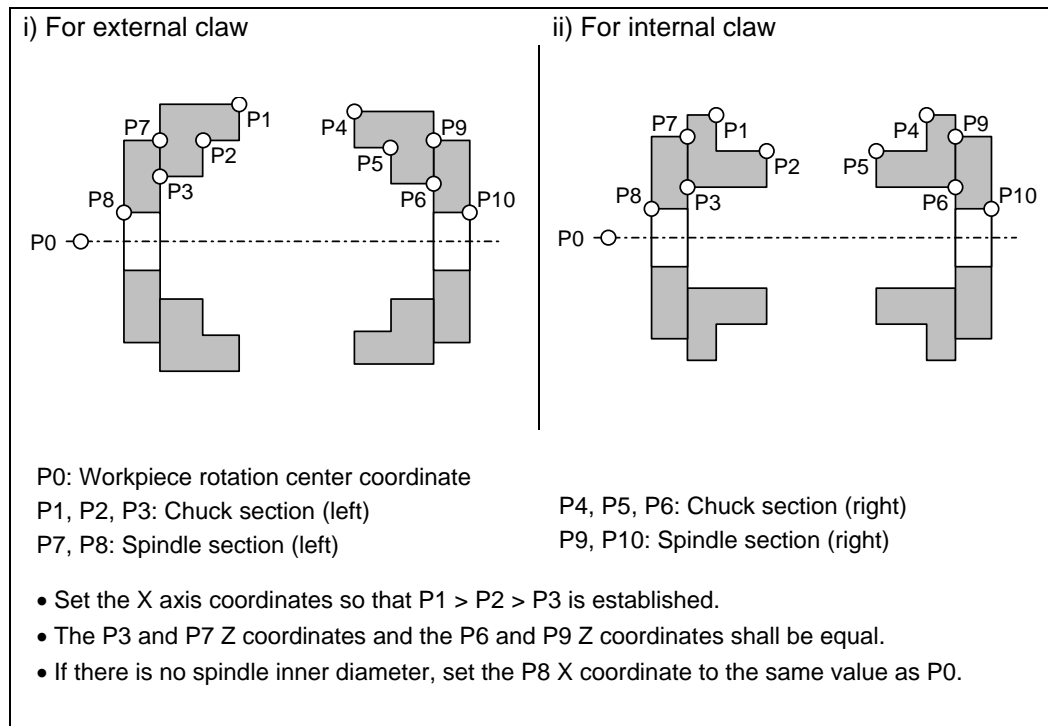
The X axis coordinate of each point must be set in the following manner.

$$\begin{array}{ll} P1 \geq P2 \geq P3 & P4 \geq P5 \geq P6 \\ P7 \geq P8 & P9 \geq P10 \end{array}$$

If the P8 Z axis coordinate must be less than P1 to P3, and the P10 Z axis coordinate must be more than P4 to P6.

The positions of the P0 to P10 points in the chuck/tailstock are as follow.

#### (a) Setting the chuck

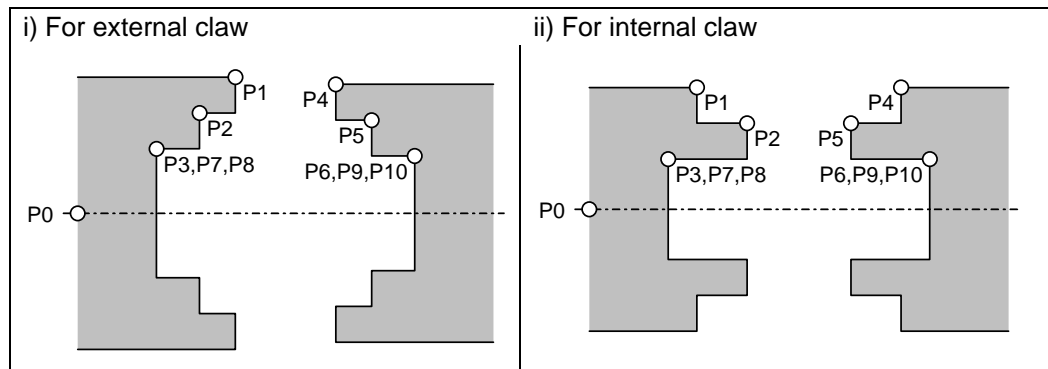




## 15. Protection Function

### 15.1 Chuck barrier/tailstock barrier

If a spindle section is not to be set in the manner of the conventional barrier, set the P3, P7 and P8 values to the same values, and the P6, P9 and P10 values to the same values. The barrier range will be as shown below in this case.



#### (b) Setting the tailstock

The diagram shows a tailstock assembly with several vertical bars. A horizontal dashed line represents the P0 coordinate. Points P1, P2, and P3 are marked on the top of the bars from left to right. Points P8 and P10 are marked on the horizontal dashed line. Points P4, P5, and P6 are marked on the top of the bars from right to left.

- P7 and P9 are not used.
- The tailstock end section angle is set with the parameters.
  - "#8318 STOCK ANGLE (L)"
  - "#8319 STOCK ANGLE (R)"
- If "0" is set, the angle will be 90°. (Default)
- Set the P8 X coordinate to the same value as P0. (No spindle inner diameter)

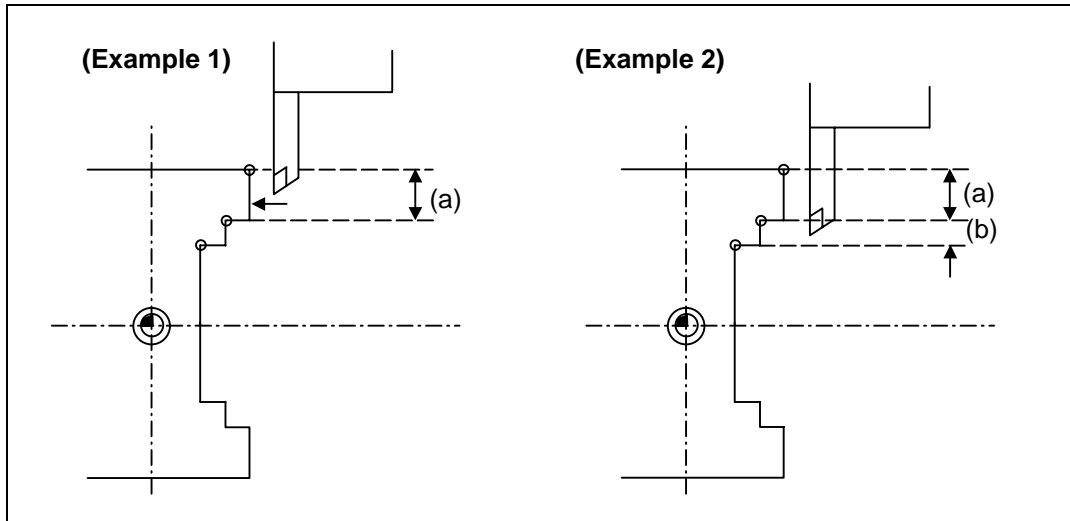


### Restrictions

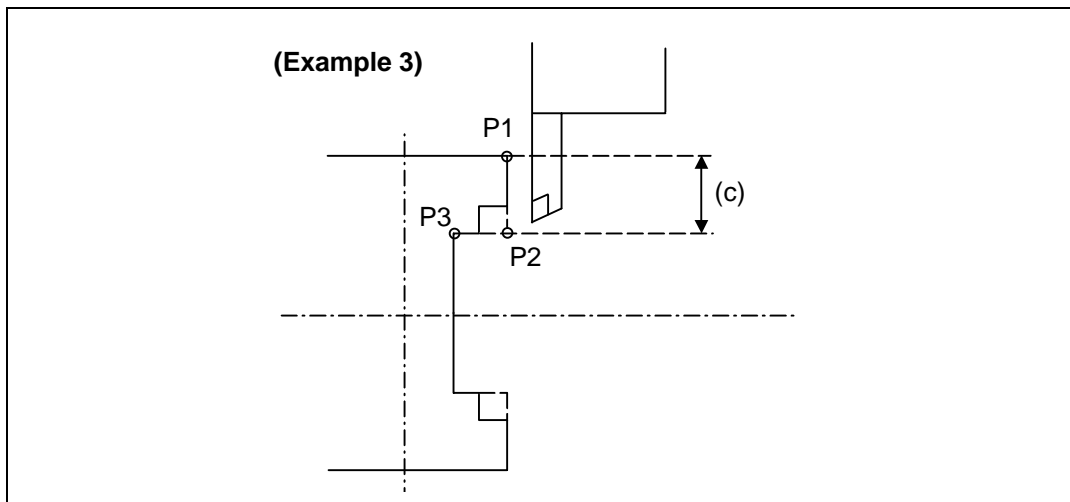
- (1) There is only one checkpoint from the tool regarding the chuck barrier/tailstock barrier. Therefore, the following cautions must be observed.

In the following examples, when the barrier points are set to be checked by the hypothetical tool nose point and the tool moves in the direction of the arrow in the drawing, the following situation may occur.

In Example 1, there is a checkpoint in the range "a", so the tool will automatically stop at the barrier boundary. However, in Example 2 there is a checkpoint in range "b", so the chuck and tool may collide in range "a".



To avoid this, Example 3 is given. In this example, if the barrier points P1, P2 and P3 are set and the checkpoint is set in range "c", the tool can be stopped at the barrier boundary.



- (2) When the tool enters the barrier region and an alarm occurs, the tool may move in the opposite direction from which it came, once the alarm is canceled by resetting.
- (3) There is no barrier region for axes without a reference point return function. Thus, there is no barrier alarm for that axis.
- (4) When the tool enters a canceled barrier region, and that barrier is then validated, an alarm will result immediately if the tool is moved.  
If the alarm is canceled with reset in this manner, invalidate the barrier (G23) before escaping or change the value set for each barrier point.
- (5) The soft limit is valid even if the barrier (G23) is invalid.

### 15.2 Stored stroke limit (6 and 7 only in G code list)



#### Function and purpose

Tool entry prohibited ranges can be set with stored stroke limit I and stored stroke limit II. These areas are set with parameters.

Using G22 and G23, the prohibited range (parameters) of the stored stroke limit II function can be changed and turned ON and OFF with the program commands.



#### Command format

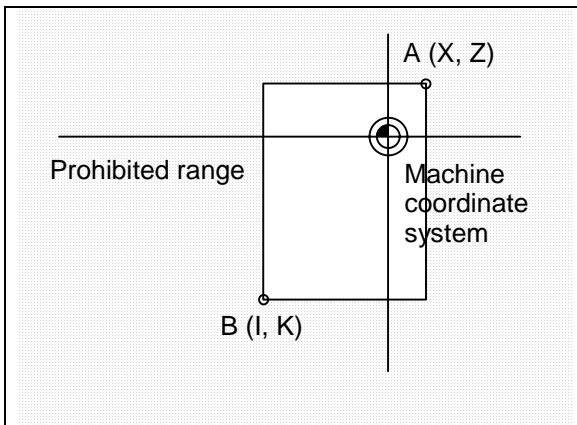
**G22 X\_ Z\_ C\_ I\_ J\_ K\_ ;**

G22	Stored stroke limit II function parameter change
G23	Stored stroke limit II function ON (all axes)
X, Z, C (1st to 3rd axis name)	Stored stroke limit + side designation
I, J, K (Plane selection axis name)	Stored stroke limit - side designation

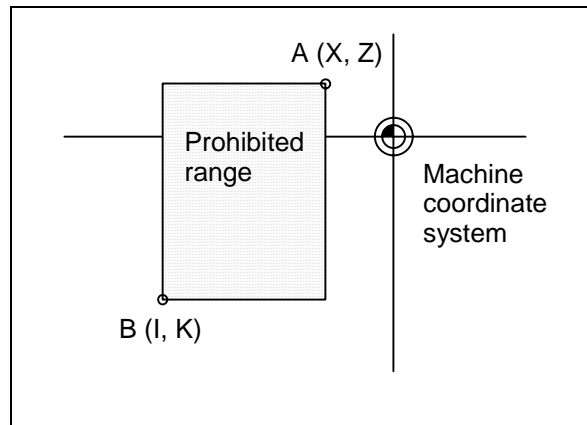


#### Detailed description

Command G22 and G23 in independent blocks.



When prohibited range is on outside



When prohibited range is on inside  
(For only E68)



#### Precautions

- (1) If the maximum value and minimum value of the stored stroke limit's prohibited range are set to the same value, the following will occur.
  - (a) When the maximum value and minimum value are set to 0, if the outside is the prohibited range, the entire range will be prohibited. If the inside is the prohibited range, the entire range will be the moveable range.
  - (b) If data other than 0 is set for the maximum value and minimum value, the entire range will be the moveable range.
- (2) The settings (G22, G23) with the stored stroke limit II program command are valid at the G code system 6 and 7 (cmdtyp = 7, 8).
- (3) G22 and G23 are unmodal (group 0), and are valid only in the commanded block.

## 16. Measurement Support Functions

### 16.1 Automatic tool length measurement

## 16. Measurement Support Functions

### 16.1 Automatic tool length measurement; G37



#### Function and purpose

These functions issue the command values from the measuring start position as far as the measurement time, move the tool in the direction of the measurement position, stop the machine once the tool has arrived at the sensor, cause the NC system to calculate automatically the difference between the coordinate values at that time and the coordinate values of the commanded measurement position and provide this difference as the tool offset amount.

When offset is already being applied to a tool, it moves the tool toward the measurement position with the offset still applied, and if a further offset amount is generated as a result of the measurement and calculation, it provides further compensation of the present offset amount.



#### Command format

**G37  $\alpha$ \_ R\_ D\_ F\_ ;**

$\alpha$	Measuring axis address and coordinates of measurement position ..... X, Z
R	This commands the distance between the measurement position and point where the movement is to start at the measuring speed (Radius value fixed, incremental value)
D	This commands the range within which the tool is to stop. (Radius value fixed, incremental value)
F	This commands the measuring speed. When R_, D_ or F_ is omitted, the value set in the parameter is used instead. <Parameter> ("TLM" on machining parameter screen)
	• #8004 Measuring feedrate (Fp) : 0 ~ 60000 (mm/min)
	• #8005 Deceleration range r : 0 ~ 99999.999 (mm)
	• #8006 Measurement range d : 0 ~ 99999.999 (mm)

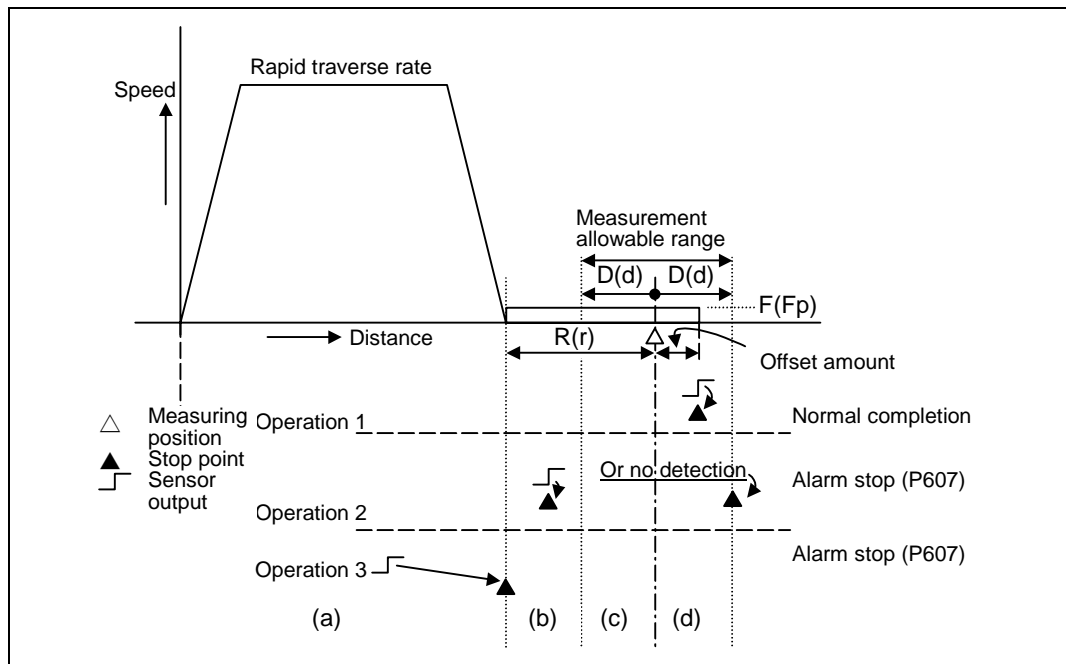
## 16. Measurement Support Functions

### 16.1 Automatic tool length measurement



#### Detailed description

##### (1) Operation with G37 command



- (2) The sensor signal (measuring position arrival signal) is used in common with the skip signal.
- (3) The feedrate will be 1mm/min if the F command and parameter measurement speed are 0.
- (4) During the synchronous feed mode, the axis will move at the synchronous feedrate [mm/rev].
- (5) An updated offset amount is valid unless it is assigned from the following T command of the G37 command.
- (6) Excluding the corresponding values at the PLC side, the delay and fluctuations in the sensor signal processing range from 0 to 0.2ms.  
As a result, the measuring error shown below is caused.

$$\text{Maximum measuring error [mm]} = \text{Measuring speed [mm/min]} \times 1/60 \times 0.2 \text{ [ms]}/1000$$

- (7) The machine position coordinates at that point in time are ready by sensor signal detection, and the machine will overtravel and stop at a position equivalent to the servo droop.

$$\text{Maximum overtravel [mm]} = \text{Measuring speed [mm/min]} \times 1/60 \times 1/\text{Position loop gain [1/s]}$$

The standard position loop gain is 33 [1/s].

## 16. Measurement Support Functions

### 16.1 Automatic tool length measurement



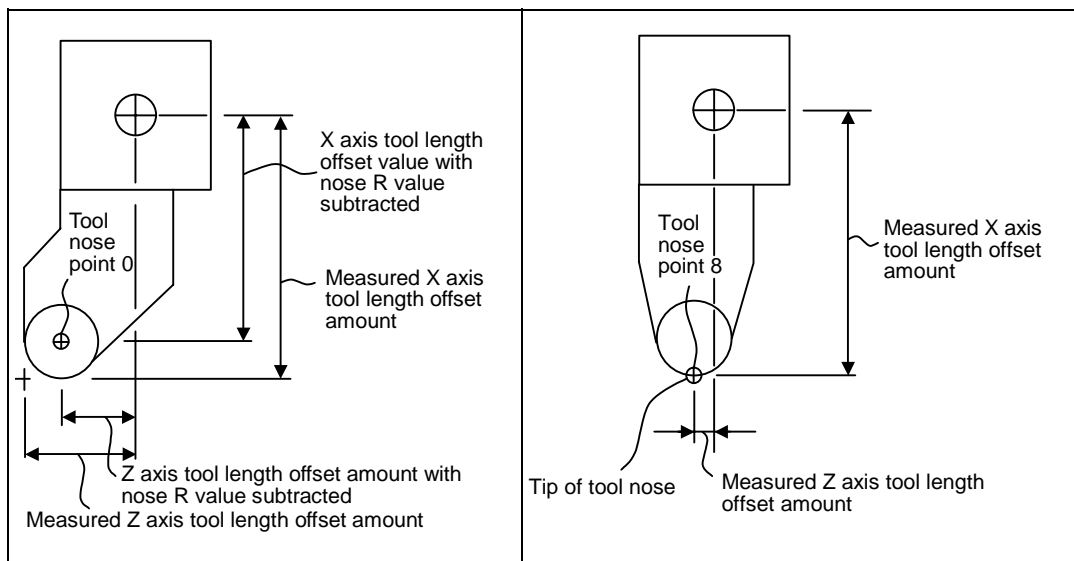
#### Precautions

- (1) Program error (P600) results if G37 is commanded when the automatic tool length measurement function is not provided.
- (2) Program error (P604) results when no axis has been commanded in the G37 block or when two or more axes have been commanded.
- (3) Program error (P605) results when the T code is commanded in the G37 block. If the low-order digit or two digits is 0, the (4) error will occur.
- (4) Program error (P606) results when T code is not commanded prior to the G37 block. When the low-order digit or two digits is 0, the program error (P606) will occur even if T is commanded.
- (5) Program error (P607) results when the sensor signal was input outside the allowable measuring range or when the sensor signal was not detected even upon arrival at the end point. If the sensor signal stays ON during operation 3 in "Detailed description" (1), it will be judged as a normal measurement even when the (b) region is not present.
- (6) When a manual interrupt is applied while the tool is moving at the measuring speed, a return must be made to the position prior to the interrupt and then operation must be resumed.
- (7) The data commanded in G37 or the parameter setting data must meet the following conditions:  
|Measurement point - start point| > R command or parameter r > D command or parameter d
- (8) When the D command and parameter d in (7) above are zero, operation will be completed normally only when the commanded measurement point and sensor signal detection point coincide. Otherwise, program error (P607) will result.
- (9) When the R and D commands as well as parameters r and d in (7) above are all zero, program error (P607) will result regardless of whether the sensor signal is present or not after the tool has been positioned at the commanded measurement point.
- (10) When the measurement command distance < measurement allowable range, it becomes the measurement allowable range for all axes.

## 16. Measurement Support Functions

### 16.1 Automatic tool length measurement

- (11) When the measurement command distance < measurement speed movement distance, all axes move at the measurement speed.
- (12) When the measurement allowable range > measurement speed movement distance, the axis moves in the measurement allowable range at the measurement speed.
- (13) Always cancel nose R compensation before commanding G37.
- (14) Calculate the tool length offset amount without regard for the nose R value and tool nose point No.,  
To set the tool nose point No. to 0, subtract the nose R value from the measured tool length offset amount and set.  
When the tool nose point No. (tool nose shape) is 5, 6, 7, or 8, measure the tool length at the tool tip.





## 16.2 Skip function; G31



## Function and purpose

When the skip signal is input externally during linear interpolation based on the G31 command, the machine feed is stopped immediately, the remaining distance is discarded and the command in the following block is executed.



## Command format

**G31 Xx/Uu Zz/Ww Ff ;**

x, z, u, w

Axis coordinate value; they are commanded as absolute or incremental values.

f

Feedrate (mm/min)

Linear interpolation can be executed using this function. If the skip signal is input externally while this command is being executed, the machine will stop, the remaining commands will be canceled and operation will be executed from the next block.



## Detailed description

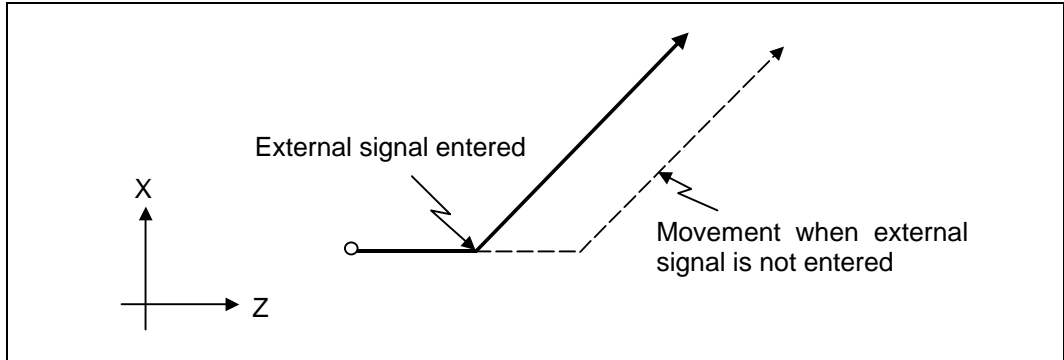
- (1) If Ff is commanded as the feedrate, command speed f will apply; if it not commanded, the value set in the parameter "#1174 skip\_F" will serve as the feedrate. In either case, the F modal will not be updated.
- (2) The machine will not automatically accelerate and decelerate with the G31 block.
- (3) Override is invalid with the G31 command, and it is fixed at 100%. Dry run is also invalid. The stop conditions (feed hold, interlock, override zero and stroke end) are valid. External deceleration is also valid.
- (4) The G31 command is unmodal and so it needs to be commanded each time.
- (5) If the skip signal is input during G31 command start, the G31 command will be completed immediately.  
When a skip signal has not been input until the G31 block completion, the G31 command will also be completed upon completion of the movement commands.
- (6) When the G31 command is issued during nose R compensation, program error (P608) will result.
- (7) When there is no F command in the G31 command and the parameter speed is also zero, program error (P603) will result.
- (8) If only the Z axis is commanded when the machine lock is ON or the Z axis cancel switch is ON, the skip signal will be ignored and execution will continue as far as the end of the block.



### Execution of G31

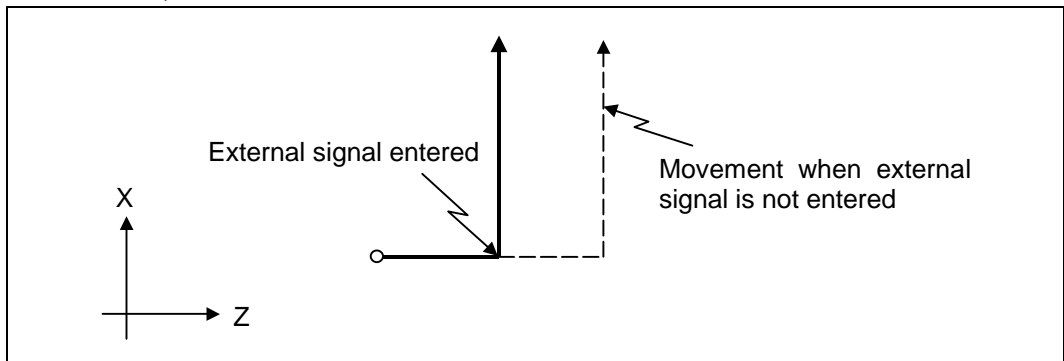
**(Example 1)** When the next block is an incremental value command

```
G31 Z1000 F100 ;
G01 U2000 W1000 ;
```



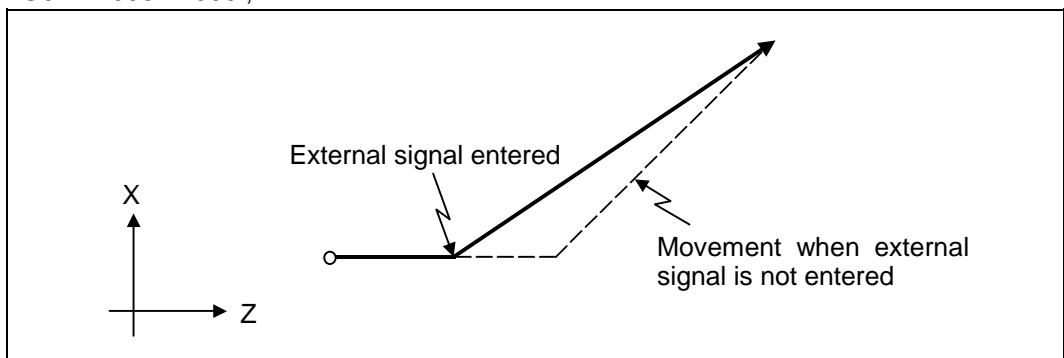
**(Example 2)** When the next block is a one axis movement command with absolute value

```
G31 Z1000 F100 ;
G01 X1000 ;
```



**(Example 3)** When the next block is a two axes movement command with absolute value

```
G31 Z1000 F100 ;
G01 X1000 Z2000 ;
```





### Detailed description (Readout of skip coordinates)

As the skip signal inputting coordinate positions have been stored in the system variables #5061 (1st axis) ~ #5063 (3rd axis), it is possible to use them with user macro.

<pre>                 } G00 X-100. ; G31 X-200. F60 ; #101 = #5061                 }             </pre>	<p>— Skip command</p> <p>— Skip signal input coordinate values (workpiece coordinate system) are readout to #101.</p>
---------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------



### Detailed description (G31 coasting)

The amount of coasting from when the skip signal is input during the G31 command until the machine stops differs according to the parameter "#1174 skip\_F" or F command in G31. The time to start deceleration to a stop after responding to the skip signal is short, so the machine can be stopped precisely with a small coasting amount. The coasting amount can be calculated from the following formula.

$$\delta_0 = \frac{F}{60} \times T_p + \frac{F}{60} \times (t_1 \pm t_2) = \frac{F}{60} \times \underbrace{(T_p + t_1)}_{\delta_1} \pm \underbrace{\frac{F}{60} \times t_2}_{\delta_2}$$

$\delta_0$  : Coasting amount (mm)

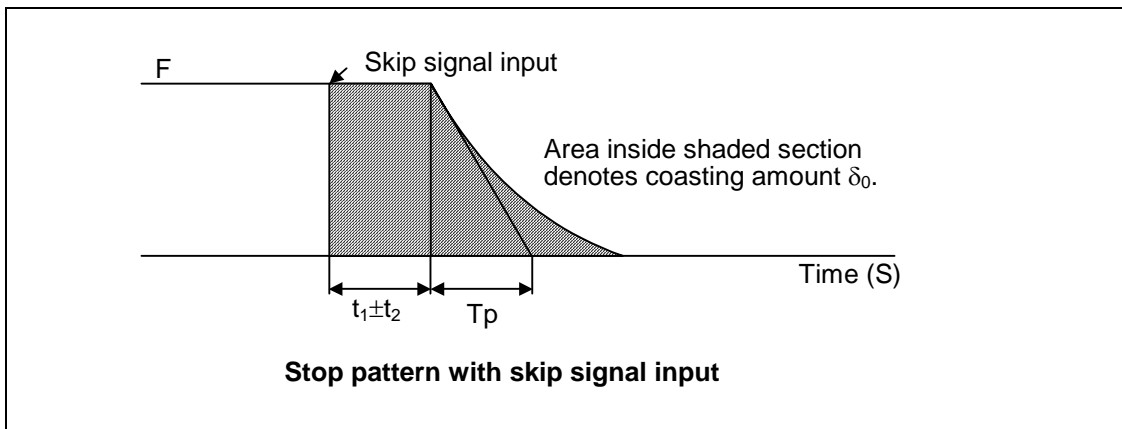
F : G31 skip speed (mm/min)

$T_p$  : Position loop time constant (s) = (position loop gain)

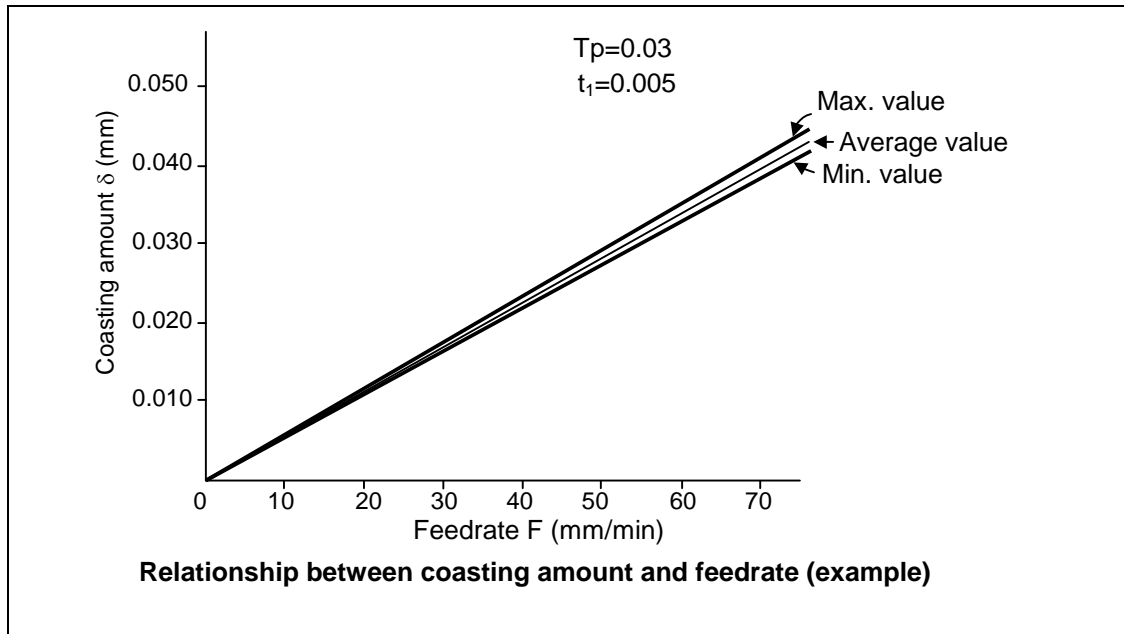
$t_1$  : Response delay time (s) = (Time from when skip signal is detected, PC is passed, and control unit is reached.)

$t_2$  : Response error time 0.001 (s)

When G31 is used for calculation, the value calculated from the section indicated by  $\delta_1$  in the above equation can be compensated, however,  $\delta_2$  results in calculation error.



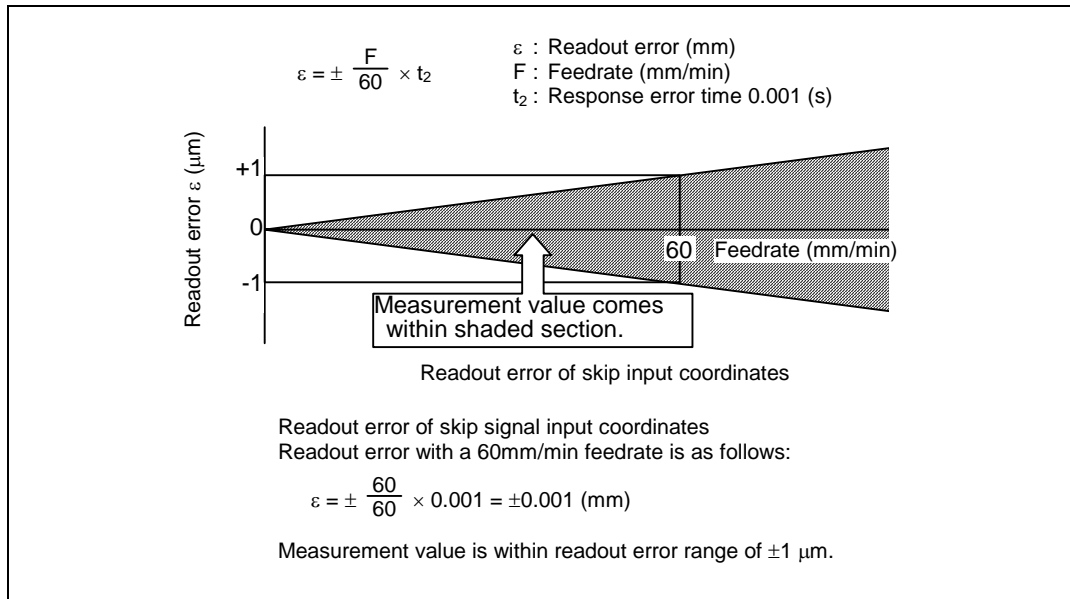
The relationship between the coasting amount and speed when  $T_p$  is 30ms and  $t_1$  is 5ms is shown in the following figure.





### Detailed description (Skip coordinate readout error mm)

- (1) Skip signal input coordinate readout  
 The coasting amount based on the position loop time constant  $T_p$  and cutting feed time constant  $T_s$  is not included in the skip signal input coordinate values. Therefore, the workpiece coordinate values applying when the skip signal is input can be read out across the error range in the following formula as the skip signal input coordinate values. However, coasting based on response delay time  $t_1$  results in a measurement error and so compensation must be provided.

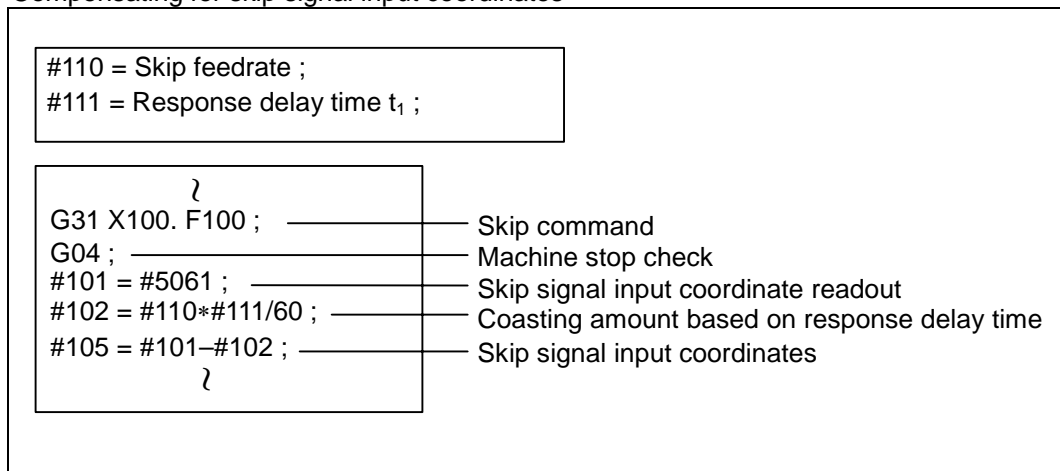


- (2) Readout of other coordinates  
 The readout coordinate values include the coasting amount. Therefore, when coordinate values are required with skip signal input, refer to the section on the G31 coasting amount to compensate the coordinate value. As in the case of (1), the coasting amount based on the delay error time  $t_2$  cannot be calculated, and this generates a measuring error.

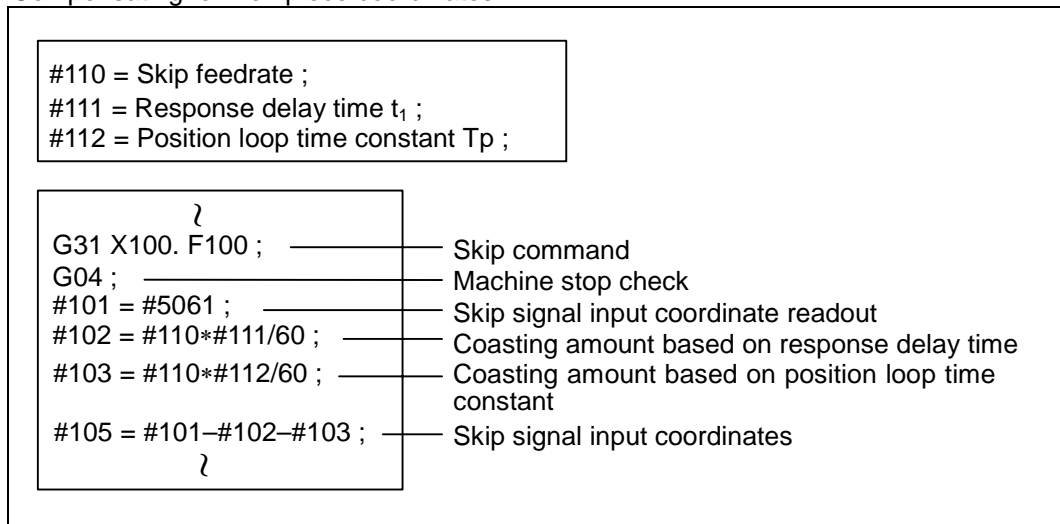


### Examples of compensating for coasting

#### (1) Compensating for skip signal input coordinates



#### (2) Compensating for workpiece coordinates



### 16.3 Multi-stage skip function; G31.n, G04



#### Function and purpose

The setting of combinations of skip signals to be input enables skipping under various conditions. The actual skip operation is the same as with G31.

The G commands which can specify skipping are G31.1, G31.2, G31.3, and G04, and the correspondence between the G commands and skip signals can be set by parameters.



#### Command format

**G31.1 Xx Zz αα Ff ;**

Xx Zz αα	Command format axis coordinate word and target coordinates
Ff	Feedrate (mm/min)

Same with G31.2 and G31.3; Ff is not required with G04.

As with the G31 command, this command executes linear interpolation and when the preset skip signal conditions have been met, the machine is stopped, the remaining commands are canceled, and the next block is executed.



#### Detailed description

- (1) Feedrate G31.1 set with the parameter corresponds to "#1176 skip1f", G31.2 corresponds to "#1178 skip2f", and G31.3 corresponds to "#1180 skip3f".
- (2) A command is skipped if it meets the specified skip signal condition.
- (3) The G31.n and G04 commands work the same as the G31 command for other than (1) and (2) above.
- (4) The feedrates corresponding to the G31.1, G31.2, and G31.3 commands can be set by parameters.
- (5) The skip conditions (logical sum of skip signals which have been set) corresponding to the G31.1, G31.2, G31.3 and G04 commands can be set by parameters.

Parameter setting	Valid skip signal		
	1	2	3
1	○		
2		○	
3	○	○	
4			○
5	○		○
6		○	○
7	○	○	○

(Skip when ○ signal is input.)



### Operation

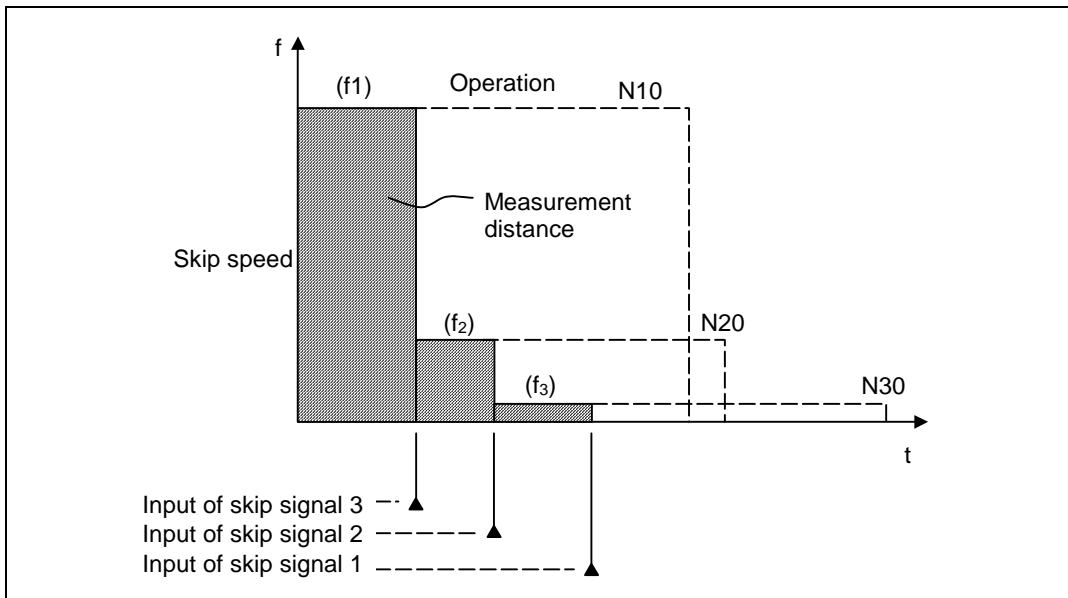
- (1) The multi-stage skip function enables the following control, thereby improving measurement accuracy and shortening the time required for measurement.

Parameter settings :

Skip condition	Skip speed
G31.1 : 7	20.0mm/min (f1)
G31.2 : 3	5.0mm/min (f2)
G31.3 : 1	1.0mm/min (f3)

Program example

```
N10 G31.1 X200.0 ;
N20 G31.2 X40.0 ;
N30 G31.3 X1.0 ;
```



**(Note 1)** If skip signal 1 is input before skip signal 3 in the above operation, N20 is skipped at that point and N30 is also ignored.

- (2) If a skip signal with the condition set during G04 (dwell) is input, the remaining dwell time is canceled and the following block is executed.



### 16.4 Multi-stage skip function 2



#### Function and purpose

During linear interpolation, command operation is skipped if skip signal parameter Pp specified with a skip command (G31), which indicates external skip signals 1 to 4, is met.

If the skip condition specified by the parameter "#1173 dwlskp" (indicating external skip signals 1 to 4) is met during execution of a dwell command (G04), the remaining dwell time is canceled and the following block is executed.



#### Command format

**G31 Xx Zz αα Pp Ff ;**

Xx Zz αα	Command format axis coordinate word and target coordinates
Pp	Skip signal command (see table below)
Ff	Feedrate (mm/min)



#### Detailed description

- (1) The skip is specified by command speed f. Note that the F modal is not updated.
- (2) The skip signal is specified by skip signal command p. p can range from 1 to 15. If p is specified outside the range, program error (P35) occurs.

Skip signal command P	Valid skip signals			
	4	3	2	1
1				○
2			○	
3			○	○
4		○		
5		○		○
6		○	○	
7		○	○	○
8	○			
~	~	~	~	~
13	○	○		○
14	○	○	○	
15	○	○	○	○

(Skip when ○ signal is input.)

## 16. Measurement Support Functions

### 16.4 Multi-stage skip function 2

- (3) The specified skip signal command is a logical sum of the skip signals.

**(Example)** G31 X100. P5 F100 ;

Operation is skipped if skip signal 1 or 3 is input.

- (4) If skip signal command Pp is not specified, the skip condition specified by the G31 command works. If speed command Ff is not specified, the skip speed specified by the G31 command parameter works.

**Table 2 Relations between skip and multi-stage skip**

Skip specifications	x		O	
	Condition	Speed	Condition	Speed
G31 X100 ; (Without P and F)	Program error (P601)		Skip 1	Parameter
G31 X100 P5 ; (Without F)	Program error (P602)		Command value	Parameter
G31 X100 F100 ; (Without P)	Program error (P601)		Skip 1	Command value
G31 X100 P5 F100 ;	Program error (P602)		Command value	Command value

**(Note)** "Parameter" in the above table indicates that specified with a skip command (G31).

- (5) If skip specification is effective and P is specified as an axis address, skip signal command P is given priority and axis address P is ignored.

**(Example)** G31 P500. F100 ;

This is regarded as a skip signal command and program error (P35) results.

- (6) Those items other than (1) to (5) are the same with the ordinary skip function (G31 without P).

## 15. Measurement Support Functions

### 16.5 Programmable current limitation

#### 16.5 Programmable current limitation

[E68]



#### Function and purpose

This function allows the current limit value of the servo axis to be changed to a desired value in the program, and is used for the workpiece stopper, etc.

The commanded current limit value is designated with a ratio of the limit current to the rated current.



#### Command format

**G10 L14 Xn ;**

(n is the skip signal 1 to 8)

L14 : Current limit value setting (+ side/- side)

X : Axis address

n : Current limit value (%) Setting value: 1 to 300



#### Precautions and restrictions

- (1) If the current limit value is reached when the current limit is valid, the current limit reached signal is output.
- (2) The following two modes can be used as the operation after the current limit is reached. The external signal determines which mode applies.
  - Normal mode  
The movement command is executed in the current state.  
During automatic operation, the movement command is executed to the end, and then the next block is moved to with the droops still accumulated.
  - Interlock mode  
The internal interlock state has been entered while accumulating the droops, and the next movement is not executed.  
During automatic operation, the operation stops at the corresponding block, and the next block is not moved to.  
During manual operation, the following same direction commands are ignored.
- (3) During the current limit, the position droop generated by the current limit can be canceled when the current limit changeover signal of external signals is canceled.  
(Note that the axis must not be moving.)
- (4) The setting range of the current limit value is 1% to 300%. Commands that exceed this range will cause a program error (P35).
- (5) If a decimal point is designated with the G10 command, only the integer will be valid.  
**(Example)** G10 L14 X10.123 ; The current limit value will be set to 10%.
- (6) For the axis name "C", the current limit value cannot be set from the program (G10 command). To set from the program, set the axis address with an incremental axis name, or set the axis name to one other than "C".

## Appendix 1. Program Parameter Input N No. Correspondence Table

### Appendix 1. Program Parameter Input N No. Correspondence Table

**(Note 1)** The units in the table indicate the minimum setting units for the parameter data.

**(Note 2)** The setting ranges given in the table are the setting ranges on the screen. Designate parameters related to the length by doubling the input setting unit.

**(Example 1)** If the input setting unit in the metric system is "B" (0.001mm), set 30mm in the parameter.

**(Example 2)** If the input setting unit in the inch system is "B" (0.0001inch), set 5inch in the parameter.

**(Note 3)** The binary type parameters must be converted into by-type data, and commanded with a decimal data after address D.

**(Example 1)** Binary data  
 $01010101_B = 55_H = 85_D \dots\dots\dots$  Command 85

**(Example 2)** ASCII code  
 "M" =  $01001101_B = 4D_H = 77_D \dots\dots\dots$  Command 77  
 (B indicates Binary, H indicates Hexadecimal, and D indicates Decimal.)

**[Machining parameters]**

**Parameter 1.**

#	Item	P	A	N	Data type	Setting range	(Unit)	Remarks
8007	OVERRIDE	11	-	1010	D	0 to 100	(%)	
8008	MAX ANGLE	11	-	756	L	0 to 180	(°)	
8009	DSC. ZONE	11	-	760	L	0 to 99999.999	(mm)	
8010	ABS. MAX.	11	-	776	L	0 to 99.999	(mm)	
8011	INC. MAX.	11	-	780	L	0 to 99.999	(mm)	
8013	G83 n	11	-	832	L	0 to 99999.999	(mm)	
8014	CDZ-VALE	11	-	1012	D	0 to 127	0.1 lead	
8015	CDZ-ANGLE	11	-	1011	D	0 to 89	(°)	
8016	G71 MINIMUM	11	-	788	L	0 to 99.999	(mm)	
8017	DELTA-D	11	-	792	L	0 to 99.999	(mm)	

## Appendix 1. Program Parameter Input N No. Correspondence Table

### [Machining parameters]

### Parameter 1.

#	Item	P	A	N	Data type	Setting range	(Unit)	Remarks
8051	G71 THICK	11	-	784	L		(mm)	
8052	PULL UP	11	-	796	L		(mm)	
8053	G73 U	11	-	800	L		(mm)	
8054	W	11	-	804	L		(mm)	
8055	R	11	-	808	L		(time)	
8056	G74 RETRACT	11	-	820	L		(mm)	
8057	G76 LAST-D	11	-	824	L		(mm)	
8058	TIMES	11	-	997	D	0 to 99	(time)	
8059	ANGLE	11	-	998	D	0 to 99	(°)	

### [Axis parameters]

### Parameter 1.

#	Item	P	A	N	Data type	Setting range	(Unit)	Remarks
8202	OT-CHECK OFF	2	No.	897	H2	0 to 1		
8204	OT-CHECK-N	2	No.	916	L	±99999.999	(mm)	
8205	OT-CHECK-P	2	No.	912	L	±99999.999	(mm)	

### [Barrier data]

### Parameter 1.

#	Item	P	A	N	Data type	Setting range	(Unit)	Remarks
8300	(P0) X	11	-	1128	L	±99999.999	(mm)	
8301	(P1) X	11	No.	1136	L	±99999.999	(mm)	
	Z	11	No.	1160	L	±99999.999	(mm)	
8302	(P2) X	11	No.	1140	L	±99999.999	(mm)	
	Z	11	No.	1164	L	±99999.999	(mm)	
8303	(P3) X	11	No.	1144	L	±99999.999	(mm)	
	Z	11	No.	1168	L	±99999.999	(mm)	
8304	(P4) X	11	No.	1148	L	±99999.999	(mm)	
	Z	11	No.	1172	L	±99999.999	(mm)	
8305	(P5) X	11	No.	1152	L	±99999.999	(mm)	
	Z	11	No.	1176	L	±99999.999	(mm)	
8306	(P6) X	11	No.	1156	L	±99999.999	(mm)	
	Z	11	No.	1180	L	±99999.999	(mm)	

## Appendix 1. Program Parameter Input N No. Correspondence Table

### [PLC data]

### Setup parameter 6.

#	Item	P	A	N	Data type	Setting range	(Unit)	Remarks
6301 to 6348	PLC constant	5	-	1 to 48	L	±99999999		

### [PLC timer]

### Setup parameter 6.

#	Item	P	A	N	Data type	Setting range	(Unit)	Remarks
6000 to 6015	10ms TIMER (T0 to T15)	6	-	0 to 15	S	0 to 32767	0.01s	
6016 to 6095	100ms TIMER (T56 to T135)	6	-	16 to 95	S	0 to 32767	0.1s	
6096 to 6103	100ms TIMER (T232 to T239) (INC.)	6	-	96 to 103	S	0 to 32767	0.1s	

### [PLC counter]

### Setup parameter 6.

#	Item	P	A	N	Data type	Setting range	(Unit)	Remarks
C0 to C23	PLC counter	7	-	0 to 23	L	0 to 32767		

### [Bit selection]

### Setup parameter 6.

#	Item	P	A	N	Data type	Setting range	(Unit)	Remarks
6401 to 6496	Bit selection	8	-	1 to 96	D or H0 to H7	0x00 to 0xFF (Note) 0 to 256 0 to 1	(Hex) (Dec)	

**(Note)** Convert the 0x00 to 0xFF data into a decimal before using.

## Appendix 2. Program Error

### Appendix 2. Program Error

(The bold characters are the message displayed in the screen.)

These alarms occur during automatic operation, and the causes of these alarms are mainly program errors which occur, for instance, when mistakes have been made in the preparation of the machining programs or when programs which conform to the specification have not been prepared.

Error No.	Details	Remedy
<b>P 10</b>	<b>EXCS. AXIS No.</b> The number of axis addresses commanded in the same block exceeds the specifications.	<ul style="list-style-type: none"> <li>• Divide the alarm block command into two.</li> <li>• Check the specifications.</li> </ul>
<b>P 11</b>	<b>AXIS ADR. ERROR</b> The axis address commanded by the program and the axis address set by the parameter do not match.	<ul style="list-style-type: none"> <li>• Revise the axis names in the program.</li> </ul>
<b>P 20</b>	<b>DIVISION ERROR</b> An axis command which cannot be divided by the command unit has been issued.	<ul style="list-style-type: none"> <li>• Check the program.</li> </ul>
<b>P 29</b>	<b>NOT ACCEPT CMND</b> The normal line control command (G40.1, G41.1, G42.1) has been issued during the modal in which the normal line control is not acceptable.	<ul style="list-style-type: none"> <li>• Check the program.</li> </ul>
<b>P 30</b>	<b>PARITY H</b> The number of holes per character on the paper tape is even for EIA code and odd for ISO code.	<ul style="list-style-type: none"> <li>• Check the paper tape.</li> <li>• Check the tape puncher and tape reader.</li> </ul>
<b>P 31</b>	<b>PARITY V</b> The number of characters per block on the paper tape is odd.	<ul style="list-style-type: none"> <li>• Make the number of characters per block on the paper tape even.</li> <li>• Set the parameter parity V selection OFF.</li> </ul>
<b>P 32</b>	<b>ADDRESS ERROR</b> An address not listed in the specifications has been used.	<ul style="list-style-type: none"> <li>• Check and revise the program address.</li> <li>• Check and correct the parameters values.</li> <li>• Check the specifications.</li> </ul>
<b>P 33</b>	<b>FORMAT ERROR</b> The command format in the program is not correct.	<ul style="list-style-type: none"> <li>• Check the program.</li> </ul>
<b>P 34</b>	<b>G-CODE ERROR</b> A G code not listed in the specifications has been used. An illegal G code was commanded during the coordinate rotation command (G68).	<ul style="list-style-type: none"> <li>• Check and correct the G code address in the program.</li> </ul>

## Appendix 2. Program Error

Error No.	Details	Remedy
<b>P 35</b>	<b>CMD-VALUE OVER</b> The setting range for the addresses has been exceeded.	<ul style="list-style-type: none"> <li>• Check the program.</li> </ul>
<b>P 36</b>	<b>PROGRAM END ERR</b> "EOR" has been read during tape and memory operation.	<ul style="list-style-type: none"> <li>• Enter the M02 and M30 command at the end of the program.</li> <li>• Enter the M99 command at the end of the subprogram.</li> </ul>
<b>P 37</b>	<b>PROG. NO. ZERO</b> A zero has been specified for program and sequence numbers.	<ul style="list-style-type: none"> <li>• The program numbers are designated across a range from 1 to 99999999.</li> <li>• The sequence numbers are designated across a range from 1 to 99999.</li> </ul>
<b>P 39</b>	<b>NO SPEC ERR</b> <ul style="list-style-type: none"> <li>• A non-specified G code was specified.</li> <li>• The high-speed program server operation specifications are not provided.</li> </ul>	<ul style="list-style-type: none"> <li>• Check the specifications.</li> </ul>
<b>P 40</b>	<b>PREREAD BL. ERR</b> When tool radius compensation is executed, there is an error in the pre-read block and so the interference check is disabled.	<ul style="list-style-type: none"> <li>• Reconsider the program.</li> </ul>
<b>P 60</b>	<b>OVER CMP. LENG.</b> The commanded movement distance is excessive. (Over $2^{31}$ )	<ul style="list-style-type: none"> <li>• Reconsider the axis address command.</li> </ul>
<b>P 62</b>	<b>F-CMD. NOTHING</b> <ul style="list-style-type: none"> <li>• No feed rate command has been issued.</li> <li>• There is no F command in the cylindrical interpolation or pole coordinate interpolation immediately after the G95 mode is commanded.</li> </ul>	<ul style="list-style-type: none"> <li>• The default movement modal command at power ON is G01. This causes the machine to move without a G01 command if a movement command is issued in the program, and an alarm results. Use an F command to specify the feedrate.</li> <li>• Specify F with a thread lead command.</li> </ul>
<b>P 65</b>	<b>NO G05P3 SPEC</b>	<ul style="list-style-type: none"> <li>• Check the high-speed mode III specifications.</li> </ul>
<b>P 70</b>	<b>ARC ERROR</b> <ul style="list-style-type: none"> <li>• There is an error in the arc start and end points as well as in the arc center.</li> <li>• The difference of the involute curve through the start point and the end point is large.</li> <li>• When arc was commanded, one of the two axes configuring the arc plane was a scaling valid axis.</li> </ul>	<ul style="list-style-type: none"> <li>• Check the numerical values of the addresses that specify the start and end points, arc center as well as the radius in the program.</li> <li>• Check the "+" and "-" directions of the address numerical values.</li> </ul>



## Appendix 2. Program Error

Error No.	Details	Remedy
<b>P 71</b>	<b>ARC CENTER</b> <ul style="list-style-type: none"> <li>The arc center is not sought during R-specified circular interpolation.</li> <li>The curvature center of the involute curve cannot be obtained.</li> </ul>	<ul style="list-style-type: none"> <li>Check the numerical values of the addresses in the program.</li> <li>Check whether the start point or end point is on the inner side of the base circle for involute interpolation. When carrying out tool radius compensation, check that the start point and end point after compensation are not on the inner side of the base circle for involute interpolation.</li> <li>Check whether the start point and end point are at an even distance from the center of the base circle for involute interpolation.</li> </ul>
<b>P 72</b>	<b>NO HELICAL SPEC</b> A helical command has been issued though it is not included in the specifications.	<ul style="list-style-type: none"> <li>Check the helical specifications.</li> <li>An Axis 3 command was issued by the circular interpolation command. If there is no helical specification, the linear axis is moved to the next block.</li> </ul>
<b>P 90</b>	<b>NO THREAD SPEC</b> A thread cutting command has been issued though it is not included in the specifications.	<ul style="list-style-type: none"> <li>Check the specifications.</li> </ul>
<b>P 93</b>	<b>SCREW PITCH ERR</b> The screw pitch has not been set correctly when the thread cutting command is issued.	<ul style="list-style-type: none"> <li>Issue the thread cutting command and then set the screw pitch command properly.</li> </ul>
<b>P100</b>	<b>NO CYLIND SPEC</b> Cylindrical interpolation was commanded when the cylindrical interpolation specifications were not provided.	<ul style="list-style-type: none"> <li>Check the specifications.</li> </ul>
<b>P111</b>	<b>PLANE CHG (CR)</b> Plane selection commands (G17, G18, and G19) were issued when a coordinate rotation command (G68) was issued.	<ul style="list-style-type: none"> <li>Before issuing the plane selection commands, issue G68 and then G69 (coordinate rotation cancel).</li> </ul>
<b>P112</b>	<b>PLANE CHG (CC)</b> <ul style="list-style-type: none"> <li>A plane selection command (G17, G18, G19) has been issued when the tool radius compensation command (G41, G42) or nose R compensation command (G41, G42, G46) is issued.</li> <li>The plane selection command was issued when nose R compensation is completed, there is no axial movement command after the G40 command, and the compensation has not been canceled.</li> </ul>	<ul style="list-style-type: none"> <li>Issue the plane selection command after the tool radius compensation command or nose R compensation command has been canceled (issue axial movement command after the G40 cancel command).</li> </ul>

## Appendix 2. Program Error

Error No.	Details	Remedy
<b>P113</b>	<b>ILLEGAL PLANE</b> The arc command axis is not on the selected plane.	<ul style="list-style-type: none"> <li>• Issue arc command on the correctly selected plane.</li> </ul>
<b>P122</b>	<b>NO AUTO C-OVR</b> An automatic corner override command (G62) has been issued though it is not included in the specifications.	<ul style="list-style-type: none"> <li>• Check the specifications.</li> <li>• Delete the G62 command from the program.</li> </ul>
<b>P126</b>	<b>ILL. CMD(H.A.)</b> An illegal command was issued during the high-accuracy control mode. <ul style="list-style-type: none"> <li>• A G code group 13 command was issued during the high-accuracy control mode.</li> <li>• Milling, cylindrical interpolation or pole coordinate interpolation was commanded during the high-accuracy control mode.</li> </ul>	<ul style="list-style-type: none"> <li>• Reconsider the program.</li> </ul>
<b>P130</b>	<b>2nd AUX. ADDR</b> The 2nd miscellaneous function address specified in the program does not match that set by the parameter.	<ul style="list-style-type: none"> <li>• Check and correct the 2nd miscellaneous function address in the program.</li> </ul>
<b>P131</b>	<b>NO G96 SPEC</b> (No constant peripheral speed) The constant peripheral speed command (G96) was issued despite the fact that such a command does not exist in the specifications.	<ul style="list-style-type: none"> <li>• Check the specifications.</li> <li>• Change from the constant peripheral speed command (G96) to the rotation speed command (G97).</li> </ul>
<b>P132</b>	<b>SPINDLE S = 0</b> No spindle speed command has been specified.	<ul style="list-style-type: none"> <li>• Reconsider the program.</li> </ul>
<b>P133</b>	<b>G96 P-No. ERR</b> An invalid constant peripheral speed control axis has been specified.	<ul style="list-style-type: none"> <li>• Reconsider the parameter specified for the constant peripheral speed control axis.</li> </ul>
<b>P140</b>	<b>NO T-POS OFST</b> The position compensation command (G45 to G48) specifications are not available.	<ul style="list-style-type: none"> <li>• Check the specifications.</li> </ul>
<b>P141</b>	<b>PAT-ROT ERROR</b> Position compensation was commanded during the figure rotation or coordinate rotation command.	<ul style="list-style-type: none"> <li>• Reconsider the program.</li> </ul>
<b>P142</b>	<b>T-OFFS G2 ERR</b> A position compensation invalid arc command was commanded.	<ul style="list-style-type: none"> <li>• Reconsider the program.</li> </ul>

## Appendix 2. Program Error

Error No.	Details	Remedy
<b>P150</b>	<b>NO C-CMP SPEC</b> <ul style="list-style-type: none"> <li>• Even though there were no tool radius compensation specifications, tool radius compensation commands (G41 and G42) were issued.</li> <li>• Even though there were no nose R compensation specifications, nose R compensation commands (G41, G42, and G46) were issued.</li> </ul>	<ul style="list-style-type: none"> <li>• Check the specifications.</li> </ul>
<b>P151</b>	<b>G2, 3 CMP. ERR</b> A compensation command (G40, G41, G42, G43, G44, G46) has been issued in the arc mode (G02, G03).	<ul style="list-style-type: none"> <li>• Issue the linear command (G01) or rapid traverse command (G00) in the compensation command block or cancel block. (Set the modal to linear interpolation.)</li> </ul>
<b>P152</b>	<b>I.S.P NOTHING</b> In interference block processing during execution of a tool radius compensation (G41 or G42) or nose R compensation (G41, G42, or G46) command, the intersection point after one block is skipped cannot be determined.	<ul style="list-style-type: none"> <li>• Reconsider the program.</li> </ul>
<b>P153</b>	<b>I.F ERROR</b> An interference error has arisen while the tool radius compensation command (G41, G42) or nose R compensation command (G41, G42, G46) was being executed.	<ul style="list-style-type: none"> <li>• Reconsider the program.</li> </ul>
<b>P155</b>	<b>F-CYC ERR (CC)</b> A fixed cycle command has been issued in the radius compensation mode.	<ul style="list-style-type: none"> <li>• The radius compensation mode is established when a fixed cycle command is executed and so the radius compensation cancel command (G40) should be issued.</li> </ul>
<b>P156</b>	<b>BOUND DIRECT</b> At the start of G46 nose R compensation, the compensation direction is undefined if this shift vector is used.	<ul style="list-style-type: none"> <li>• Change the vector to that with which the compensation direction is defined.</li> <li>• Exchange with a tool having a different tip point number.</li> </ul>
<b>P157</b>	<b>SIDE REVERSED</b> During G46 nose R compensation, the compensation direction is inverted.	<ul style="list-style-type: none"> <li>• Change the G command to that which allows inversion of the compensation direction (G00, G28, G30, G33, or G53).</li> <li>• Exchange with a tool having a different tip point number.</li> <li>• Turn ON the #8106 G46 inversion error avoidance parameter.</li> </ul>
<b>P158</b>	<b>ILLEGAL TIP P.</b> During G46 nose R compensation, the tip point is illegal (other than 1 to 8).	<ul style="list-style-type: none"> <li>• Change the tip point number to a legal one.</li> </ul>

## Appendix 2. Program Error

Error No.	Details	Remedy						
<b>P170</b>	<b>NO CORR. No.</b> The compensation number (DOO, TOO, HOO) command was not given when the radius compensation (G41, G42, G43, G46) command was issued. Alternatively, the compensation number is larger than the number of sets in the specifications.	<ul style="list-style-type: none"> <li>• Add the compensation number command to the compensation command block.</li> <li>• Check the number of compensation number sets a correct it to a compensation number command within the permitted number of compensation sets.</li> </ul>						
<b>P172</b>	<b>G10 L-No. ERR</b> (G10 L-number error) The L address command is not correct when the G10 command is issued.	<ul style="list-style-type: none"> <li>• Check the address L-Number of the G10 command and correct the number.</li> </ul>						
<b>P173</b>	<b>G10 P-No. ERR</b> (G10 compensation error) When the G10 command is issued, a compensation number outside the permitted number of sets in the specifications has been commanded for the compensation number command.	<ul style="list-style-type: none"> <li>• First check the number of compensation sets and then set the address P designation to within the permitted number of sets.</li> </ul>						
<b>P177</b>	<b>LIFE COUNT ACT</b> Registration of tool life management data with G10 was attempted when the used data count valid signal was ON.	<ul style="list-style-type: none"> <li>• The tool life management data cannot be registered when counting the used data. Turn the used data count valid signal OFF.</li> </ul>						
<b>P178</b>	<b>LIFE DATA OVER</b> The No. of registration groups, total No. of registered tools or the No. of registrations per group exceeded the specifications range.	<ul style="list-style-type: none"> <li>• Review the No. of registrations. The maximum No. of registrations is shown below.</li> </ul> <table border="1" data-bbox="863 1285 1222 1435"> <tbody> <tr> <td>No. of groups</td> <td>80</td> </tr> <tr> <td>No. of tools</td> <td>80</td> </tr> <tr> <td>Per group</td> <td>16</td> </tr> </tbody> </table>	No. of groups	80	No. of tools	80	Per group	16
No. of groups	80							
No. of tools	80							
Per group	16							
<b>P179</b>	<b>GROUP No. ILL.</b> <ul style="list-style-type: none"> <li>• When registering the tool life management data with G10, the group No. was commanded in duplicate.</li> <li>• A group No. that was not registered was designated during the T□□□□99 command.</li> <li>• An M code command must be issued as a single command but coexists in the same block as that of another M code command.</li> <li>• The M code commands set in the same group exist in the same block.</li> </ul>	<ul style="list-style-type: none"> <li>• The group No. cannot be commanded in duplicate. When registering the group data, register it in group units.</li> <li>• Correct to the correct group No.</li> </ul>						

## Appendix 2. Program Error

Error No.	Details	Remedy
<b>P180</b>	<b>NO BORING CYC.</b> A fixed cycle command was issued though there are not fixed cycle (G72 - G89) specifications.	<ul style="list-style-type: none"> <li>• Check the specifications.</li> <li>• Correct the program.</li> </ul>
<b>P181</b>	<b>NO S-CMD (TAP)</b> The spindle rotation speed command has not been issued when the hole drilling fixed cycle command is given.	<ul style="list-style-type: none"> <li>• Issue the spindle rotation speed command (S) when the hole drilling fixed cycle command G84, G74 (G84, G88) is given.</li> </ul>
<b>P182</b>	<b>SYN TAP ERROR</b> Connection to the main spindle unit was not established.	<ul style="list-style-type: none"> <li>• Check connection to the main spindle.</li> <li>• Check that the main spindle encoder exists.</li> </ul>
<b>P183</b>	<b>PTC/THD, No.</b> The pitch or thread number command has not been issued in the tap cycle of a hole drilling fixed cycle command.	<ul style="list-style-type: none"> <li>• Specify the pitch data and the number of threads by F or E command.</li> </ul>
<b>P184</b>	<b>NO PTC/THD CMD</b> <ul style="list-style-type: none"> <li>• The pitch or the number of threads per inch is illegal in the tap cycle of the hole drilling fixed cycle command.</li> <li>• The pitch is too small for the spindle rotation speed.</li> <li>• The thread number is too large for the spindle rotation speed.</li> </ul>	<ul style="list-style-type: none"> <li>• Check the pitch or the number of threads per inch.</li> </ul>
<b>P190</b>	<b>NO CUTTING CYC</b> A lathe cutting cycle command was input although the lathe cutting cycle was undefined in the specification.	<ul style="list-style-type: none"> <li>• Check the specification.</li> <li>• Delete the lathe cutting cycle command.</li> </ul>
<b>P191</b>	<b>TAPER LENG ERR</b> In the lathe cutting cycle, the specified length of taper section is illegal.	<ul style="list-style-type: none"> <li>• The radius set value in the lathe cycle command must be smaller than the axis shift amount.</li> </ul>
<b>P192</b>	<b>CHAMFERING ERR</b> Chamfering in the thread cutting cycle is illegal.	<ul style="list-style-type: none"> <li>• Set a chamfering amount not exceeding the cycle.</li> </ul>
<b>P200</b>	<b>NO MRC CYC SPC</b> The fixed cycle for compound lathe I (G70 to G73) was commanded when the fixed cycle for compound lathe I specifications were not provided.	<ul style="list-style-type: none"> <li>• Check the specification.</li> </ul>

## Appendix 2. Program Error

Error No.	Details	Remedy
<b>P201</b>	<p><b>PROG. ERR (MRC)</b></p> <ul style="list-style-type: none"> <li>When called with a fixed cycle for compound lathe I command, the subprogram contained at least one of the following commands: <ul style="list-style-type: none"> <li>Reference point return command (G27, G28, G29, G30)</li> <li>Thread cutting (G33, G34)</li> <li>Fixed cycle skip-function (G31)</li> </ul> </li> <li>The first move block of the finish shape program in fixed cycle for compound lathe I contains an arc command.</li> </ul>	<ul style="list-style-type: none"> <li>Delete the following G codes from this subprogram that is called with the fixed cycle for compound lathe I commands (G70 to G73): G27, G28, G29, G30, G31, G33, G34, fixed cycle G-code.</li> <li>Remove G2 and G3 from the first move block of the finish shape program in fixed cycle for compound lathe I.</li> </ul>
<b>P202</b>	<p><b>BLOCK OVR (MRC)</b></p> <p>The number of blocks in the shape program of the fixed cycle for compound lathe I is over 50 or 200 (this differs according to the model).</p>	<ul style="list-style-type: none"> <li>Specify 50 or a less value. The number of blocks in the shape program called by the fixed cycle for compound lathe I commands (G70 to G73) must be decreased below 50 or 200 (this differs according to the model).</li> </ul>
<b>P203</b>	<p><b>CONF. ERR (MRC)</b></p> <p>The fixed cycle for compound lathe I (G70 to G73) shape program could not cut the work normally because it defined an abnormal shape.</p>	<ul style="list-style-type: none"> <li>Check the fixed cycle for compound lathe I (G70 to G73) shape program.</li> </ul>
<b>P204</b>	<p><b>VALUE ERR (MRC)</b></p> <p>A command value of the fixed cycle for compound lathe (G70 to G76) is illegal.</p>	<ul style="list-style-type: none"> <li>Check the fixed cycle for compound lathe (G70 to G76) command value.</li> </ul>
<b>P210</b>	<p><b>NO PAT CYC SPC</b></p> <p>A fixed cycle for compound lathe II (G74 to G76) command was input although it was undefined in the specification.</p>	<ul style="list-style-type: none"> <li>Check the specification.</li> </ul>
<b>P220</b>	<p><b>NO SPECIAL CYC</b></p> <p>No special fixed cycle specifications are available.</p>	<ul style="list-style-type: none"> <li>Check the specifications.</li> </ul>
<b>P221</b>	<p><b>NO HOLE (S_CYC)</b></p> <p>A 0 has been specified for the number of holes in special fixed cycle mode.</p>	<ul style="list-style-type: none"> <li>Reconsider the program.</li> </ul>
<b>P222</b>	<p><b>G36 ANGLE ERR</b></p> <p>A G36 command specifies 0 for angle intervals.</p>	<ul style="list-style-type: none"> <li>Reconsider the program.</li> </ul>
<b>P223</b>	<p><b>G12 G13 R ERR</b></p> <p>The radius value specified with a G12 or G13 command is below the compensation amount.</p>	<ul style="list-style-type: none"> <li>Reconsider the program.</li> </ul>
<b>P224</b>	<p><b>NO G12, G13 SPC</b></p> <p>There are no circular cutting specifications.</p>	<ul style="list-style-type: none"> <li>Check the specifications.</li> </ul>

## Appendix 2. Program Error

Error No.	Details	Remedy
<b>P230</b>	<b>NESTING OVER</b> <ul style="list-style-type: none"> <li>• A subprogram has been called 8 or more times in succession from the subprogram.</li> <li>• The program in the IC card contains the M198 command.</li> <li>• The program in the IC card has been called more than once (the program in the IC card can be called only once at a time).</li> </ul>	<ul style="list-style-type: none"> <li>• Check the number of subprogram calls and correct the program so that it does not exceed 8 times.</li> <li>• When using the IC card, the IC card and the number of IC card program calls.</li> </ul>
<b>P231</b>	<b>NO N-NUMBER</b> At subprogram call time, the sequence number set at return from the subprogram or specified by GOTO, was not set.	<ul style="list-style-type: none"> <li>• Specify the sequence numbers in the call block of the subprogram.</li> <li>• When using an IC card, check the program and its No. in the IC card.</li> </ul>
<b>P232</b>	<b>NO PROGRAM No.</b> <ul style="list-style-type: none"> <li>• The specified subprogram is not registered in the memory or IC card.</li> <li>• The program file name registered in the IC card and O No. do not match.</li> </ul>	<ul style="list-style-type: none"> <li>• Enter the subprogram.</li> <li>• Check the program number in the IC card.</li> </ul>
<b>P241</b>	<b>NO VARI NUMBER</b> The variable number commanded is higher than the numbers in the specifications.	<ul style="list-style-type: none"> <li>• Check the specifications.</li> <li>• Check the program variable number.</li> </ul>
<b>P242</b>	<b>EQL. SYM. MSSG.</b> The "=" sign has not been commanded when a variable is defined.	<ul style="list-style-type: none"> <li>• Designate the "=" sign in the variable definition of the program.</li> </ul>
<b>P243</b>	<b>VARIABLE ERR.</b> An invalid variable has been specified in the left or right side of an operation expression.	<ul style="list-style-type: none"> <li>• Correct the program.</li> </ul>
<b>P252</b>	<b>PAT.&amp;COORD-ROT</b> A coordinate rotation related command (G68, G69) was issued during figure rotation.	<ul style="list-style-type: none"> <li>• Reconsider the program.</li> </ul>
<b>P260</b>	<b>NO COOD-RT SPC</b> Even though there were no coordinate rotation specifications, a coordinate rotation command was issued.	<ul style="list-style-type: none"> <li>• Check the specifications.</li> </ul>
<b>P270</b>	<b>NO MACRO SPEC</b> A macro specification was commanded though there are no such command specifications.	<ul style="list-style-type: none"> <li>• Check the specifications.</li> </ul>

## Appendix 2. Program Error

Error No.	Details	Remedy
<b>P271</b>	<b>NO MACRO INT.</b> A macro interrupt command has been issued though it is not included in the specifications.	<ul style="list-style-type: none"> <li>• Check the specifications.</li> </ul>
<b>P272</b>	<b>MACRO ILL.</b> A statement and a macro statement exist together in the same block.	<ul style="list-style-type: none"> <li>• Reconsider the program and place the executable statement and macro statement in separate blocks.</li> </ul>
<b>P273</b>	<b>MACRO OVERCALL</b> The number of macro call nests exceeded the specifications.	<ul style="list-style-type: none"> <li>• Reconsider the program and correct it so that the macro calls do not exceed the limit imposed by the specification.</li> </ul>
<b>P275</b>	<b>MACRO ARG. EX.</b> The number of macro call argument type II sets has exceeded the limit.	<ul style="list-style-type: none"> <li>• Reconsider the program.</li> </ul>
<b>P276</b>	<b>CALL CANCEL</b> A G67 command was issued though it was not during the G66 command modal.	<ul style="list-style-type: none"> <li>• Reconsider the program.</li> <li>• The G67 command is the call cancel command and so the G66 command must be designated first before it is issued.</li> </ul>
<b>P277</b>	<b>MACRO ALM MESG</b> An alarm command has been issued in #3000.	<ul style="list-style-type: none"> <li>• Refer to the operator messages on the DIAG screen.</li> <li>• Refer to the instruction manual issued by the machine manufacturer.</li> </ul>
<b>P280</b>	<b>EXC. [ ,</b> The number of parentheses "[" or "]" which can be commanded in a single block has exceeded five.	<ul style="list-style-type: none"> <li>• Reconsider the program and correct it so the number of "[" or "]" does not exceed five.</li> </ul>
<b>P281</b>	<b>[ , ] ILLEGAL</b> The number of "[" and "]" parentheses commanded in a single block does not match.	<ul style="list-style-type: none"> <li>• Reconsider the program and correct it so that "[" and "]" parentheses are paired up properly.</li> </ul>
<b>P282</b>	<b>CALC. IMPOSS.</b> The arithmetic formula is incorrect.	<ul style="list-style-type: none"> <li>• Reconsider the program and correct the formula.</li> </ul>
<b>P283</b>	<b>DIVIDE BY ZERO</b> The denominator of the division is zero.	<ul style="list-style-type: none"> <li>• Reconsider the program and correct it so that the denominator for division in the formula is not zero.</li> </ul>
<b>P290</b>	<b>IF SNT. ERROR</b> There is an error in the IF conditional GOTO□ statement.	<ul style="list-style-type: none"> <li>• Reconsider the program.</li> </ul>
<b>P291</b>	<b>WHILE SNT. ERR</b> There is an error in the WHILE conditional DO□-END□ statement.	<ul style="list-style-type: none"> <li>• Reconsider the program.</li> </ul>
<b>P292</b>	<b>SETVN SNT. ERR</b> There is an error in the SETVN□ statement when the variable name setting was made.	<ul style="list-style-type: none"> <li>• Reconsider the program.</li> <li>• The number of characters in the variable name of the SETVN statement must be 7 or less.</li> </ul>



## Appendix 2. Program Error

Error No.	Details	Remedy
<b>P293</b>	<b>DO-END EXCESS</b> The number of □'s for DO-END□ in the WHILE conditional DO□ – END□ statement has exceed 27.	<ul style="list-style-type: none"> <li>• Reconsider the program and correct it so that the number of 's in the DO – END statement does not exceed 27.</li> </ul>
<b>P294</b>	<b>DO-END MMC.</b> The DO's and END's are not paired off properly.	<ul style="list-style-type: none"> <li>• Reconsider the program and correct it so that the DO's and END's are paired off properly.</li> </ul>
<b>P295</b>	<b>WHILE/GOTO TPE</b> There is a WHILE or GOTO statement on the tape during tape operation.	<ul style="list-style-type: none"> <li>• During tape operation, a program which includes a WHILE or GOTO statement cannot be executed and so the memory operation mode is established instead.</li> </ul>
<b>P296</b>	<b>NO ADR (MACRO)</b> A required address has not been specified in the user macro.	<ul style="list-style-type: none"> <li>• Review the program.</li> </ul>
<b>P297</b>	<b>ADR-A ERR.</b> The user macro does not use address A as a variable.	<ul style="list-style-type: none"> <li>• Review the program.</li> </ul>
<b>P298</b>	<b>PTR OP (MACRO)</b> User macro G200, G201, or G202 was specified during tape or MDI operation.	<ul style="list-style-type: none"> <li>• Review the program.</li> </ul>
<b>P300</b>	<b>VER. NAME ERROR</b> The variable names have not been commanded properly.	<ul style="list-style-type: none"> <li>• Reconsider the variable names in the program and correct them.</li> </ul>
<b>P301</b>	<b>VAR NAME DUPLI</b> The name of the variable has been duplicated.	<ul style="list-style-type: none"> <li>• Correct the program so that the name is not duplicated.</li> </ul>
<b>P360</b>	<b>NO PROG.MIRR.</b> A mirror image (G50.1 or G51.1) command has been issued though the programmable mirror image specifications are not provided.	<ul style="list-style-type: none"> <li>• Check the specifications.</li> </ul>
<b>P370</b>	<b>NO OPOS MR SPC</b> The facing turret mirror image specifications are not provided.	<ul style="list-style-type: none"> <li>• Check the specifications.</li> </ul>
<b>P371</b>	<b>MIRR ILLEGAL</b> Facing turret mirror image was commanded to an axis for which external mirror image or parameter mirror image is valid. Facing turret mirror image validating mirror image for a rotary axis was commanded.	<ul style="list-style-type: none"> <li>• Check the program.</li> <li>• Check the parameters.</li> </ul>

## Appendix 2. Program Error

Error No.	Details	Remedy
<b>P380</b>	<b>NO CORNER R/C</b> A command was issued for corner chamfering/corner rounding though there are no such specifications.	<ul style="list-style-type: none"> <li>• Check the specifications.</li> <li>• Remove the corner chamfering/corner rounding command from the program.</li> </ul>
<b>P381</b>	<b>NO ARC R/C SPC</b> Corner chamfering/corner rounding was specified in the arc interpolation block although corner chamfering/corner rounding II is unsupported.	<ul style="list-style-type: none"> <li>• Check the specifications.</li> </ul>
<b>P382</b>	<b>CORNER NO MOVE</b> The block next to corner chamfering/corner rounding is not a movement command.	<ul style="list-style-type: none"> <li>• Replace the block succeeding the corner chamfering/corner rounding command by G01 command.</li> </ul>
<b>P383</b>	<b>CORNER SHORT</b> In the corner chamfering/corner rounding command, the movement distance was shorter than the value in the corner chamfering/corner rounding command.	<ul style="list-style-type: none"> <li>• Make the corner chamfering/corner rounding less than the movement distance since this distance is shorter than the corner chamfering/corner rounding.</li> </ul>
<b>P384</b>	<b>CORNER SHORT</b> When the corner chamfering/corner rounding command was input, the movement distance in the following block was shorter than the length of the corner chamfering/corner rounding.	<ul style="list-style-type: none"> <li>• Make the corner chamfering/corner rounding less than the movement distance since this distance in the following block is shorter than the corner chamfering/corner rounding.</li> </ul>
<b>P385</b>	<b>G0 G33 IN CONR</b> A block with corner chamfering/corner rounding was given during G00 or G33 modal.	<ul style="list-style-type: none"> <li>• Recheck the program.</li> </ul>
<b>P390</b>	<b>NO GEOMETRIC</b> A geometric command was issued though there are no geometric specifications.	<ul style="list-style-type: none"> <li>• Check the specifications.</li> </ul>
<b>P391</b>	<b>NO GEOMETRIC 2</b> There are no geometric IB specifications.	<ul style="list-style-type: none"> <li>• Check the specifications.</li> </ul>
<b>P392</b>	<b>LES AGL (GEOMT)</b> The angular difference between the geometric line and line is 1° or less.	<ul style="list-style-type: none"> <li>• Correct the geometric angle.</li> </ul>
<b>P393</b>	<b>INC ERR (GEOMT)</b> The second geometric block was specified by an incremental value.	<ul style="list-style-type: none"> <li>• Specify this block by an absolute value.</li> </ul>
<b>P394</b>	<b>NO G01 (GEOMT)</b> The second geometric block contains no linear command.	<ul style="list-style-type: none"> <li>• Specify the G01 command.</li> </ul>
<b>P395</b>	<b>NO ADRS (GEOMT)</b> The geometric format is invalid.	<ul style="list-style-type: none"> <li>• Recheck the program.</li> </ul>

## Appendix 2. Program Error

Error No.	Details	Remedy
<b>P396</b>	<b>PL CHG. (GEOMT)</b> A plane switching command was executed during geometric command processing.	<ul style="list-style-type: none"> <li>Execute the plane switching command before geometric command processing.</li> </ul>
<b>P397</b>	<b>ARC ERR (GEOMT)</b> In geometric IB, the circular arc end point does not contact or cross the next block start point.	<ul style="list-style-type: none"> <li>Recheck the geometric circular arc command and the preceding and following commands.</li> </ul>
<b>P398</b>	<b>NO GEOMETRIC 1B</b> Although the geometric IB specifications are not included, a geometric command is given.	<ul style="list-style-type: none"> <li>Check the specifications.</li> </ul>
<b>P421</b>	<b>PRAM. IN ERROR</b> <ul style="list-style-type: none"> <li>The specified parameter number or set data is illegal.</li> <li>An illegal G command address was input in parameter input mode.</li> <li>A parameter input command was input during fixed cycle modal or nose R compensation.</li> </ul>	<ul style="list-style-type: none"> <li>Check the program.</li> </ul>
<b>P430</b>	<b>AXIS NOT RET.</b> <ul style="list-style-type: none"> <li>A command was issued to move an axis, which has not returned to the reference point, away from that reference point.</li> <li>A command was issued to an axis removal axis.</li> </ul>	<ul style="list-style-type: none"> <li>Execute reference point return manually.</li> <li>The command was issued to an axis for which axis removal is validated so invalidate axis removal.</li> </ul>
<b>P431</b>	<b>NO 2nd REF. SPC</b> A command for second, third or fourth reference point return was issued though there are no such command specifications.	<ul style="list-style-type: none"> <li>Check the specifications.</li> </ul>
<b>P434</b>	<b>COLLATION ERR</b> One of the axes did not return to the start position when the origin point collate command (G27) was executed.	<ul style="list-style-type: none"> <li>Check the program.</li> </ul>
<b>P435</b>	<b>G27/M ERROR</b> An M command was issued simultaneously in the G27 command block.	<ul style="list-style-type: none"> <li>An M code command cannot be issued in a G27 command block and so the G27 command and M code command must be placed in separate blocks.</li> </ul>
<b>P436</b>	<b>G29/M ERROR</b> An M command was issued simultaneously in the G29 command block.	<ul style="list-style-type: none"> <li>An M code command cannot be issued in a G29 command block and so the G29 command and M code command must be placed in separate blocks.</li> </ul>

## Appendix 2. Program Error

Error No.	Details	Remedy
<b>P438</b>	<b>NOT USE (G52)</b> A local coordinate system command was issued during execution of the G54.1 command.	<ul style="list-style-type: none"> <li>Review the program.</li> </ul>
<b>P450</b>	<b>NO CHUCK BARR.</b> The chuck barrier on command (G22) was specified although the chuck barrier was undefined in the specification.	<ul style="list-style-type: none"> <li>Check the specification.</li> </ul>
<b>P460</b>	<b>TAPE I/O ERROR</b> <ul style="list-style-type: none"> <li>An error has arisen in the tape reader or, alternatively, in the printer during macro printing.</li> <li>Restart search was attempted in the tape mode (tape/IC card operation) without a search operation after a reset upon a restart search.</li> </ul>	<ul style="list-style-type: none"> <li>Check the power and cable of the connected devices.</li> <li>Check the I/O device parameters.</li> <li>Insert the IC card.</li> <li>After reset, always perform a search operation to have the head of program indexed prior to a restart search.</li> </ul>
<b>P461</b>	<b>FILE I/O ERROR</b> A file of the machining program cannot be read.	<ul style="list-style-type: none"> <li>In memory mode, the programs stored in memory may have been destroyed. Output all of the programs and tool data once and format them.</li> <li>Ensure that the external device that contains the file is mounted.</li> </ul>
<b>P462</b>	<b>DNC ERROR</b> A communication error occurred during the BTR operation.	<ul style="list-style-type: none"> <li>L01 DNC ERROR is displayed simultaneously, so remedy the problem according to the error No.</li> </ul>
<b>P480</b>	<b>NO MILL SPEC</b> <ul style="list-style-type: none"> <li>Milling was commanded when the milling specifications were not provided.</li> <li>Pole coordinate interpolation was commanded when the pole coordinate interpolation specifications were not provided.</li> </ul>	<ul style="list-style-type: none"> <li>Check the specification.</li> </ul>
<b>P481</b>	<b>MILL ILL. G</b> <ul style="list-style-type: none"> <li>An illegal G code was used during the milling mode.</li> <li>An illegal G code was used during cylindrical interpolation or pole coordinate interpolation.</li> <li>The G07.1 command was issued during the tool radius compensation.</li> </ul>	<ul style="list-style-type: none"> <li>Check the program.</li> </ul>

## Appendix 2. Program Error

Error No.	Details	Remedy
<b>P482</b>	<b>MILL ILL. AXIS</b> <ul style="list-style-type: none"> <li>• A rotary axis was commanded during the milling mode.</li> <li>• Milling was executed even though an illegal value was set for the milling axis No.</li> <li>• Cylindrical interpolation or pole coordinate interpolation was commanded during mirror image.</li> <li>• Cylindrical interpolation or pole coordinate interpolation was commanded before the tool compensation was completed after the T command.</li> <li>• G07.1 was commanded when cylindrical interpolation was not possible (there is no rotary axis, or external mirror image is ON).</li> <li>• G12.1 was commanded when polar coordinate interpolation was not possible.</li> <li>• An axis other than a cylindrical coordinate system axis was commanded during cylindrical interpolation.</li> </ul>	<ul style="list-style-type: none"> <li>• Check the machining program, parameters and PLC I/F signal.</li> </ul>
<b>P484</b>	<b>MILL AXIS RET.</b> <ul style="list-style-type: none"> <li>• Movement was commanded to an axis that had not completed reference point return during the milling mode.</li> <li>• Movement was commanded to an axis that had not completed reference point return during cylindrical interpolation or pole coordinate interpolation.</li> </ul>	<ul style="list-style-type: none"> <li>• Carry out manual reference point return.</li> </ul>

## Appendix 2. Program Error

Error No.	Details	Remedy
<b>P485</b>	<b>MILL ILL. MODAL</b> <ul style="list-style-type: none"> <li>• The milling mode was turned ON during nose R compensation or constant surface speed control.</li> <li>• A T command was issued during the milling mode.</li> <li>• The mode was switched from milling to cutting during tool compensation.</li> <li>• Cylindrical interpolation or pole coordinate interpolation was commanded during the constant surface speed control mode (G96).</li> <li>• The command unacceptable in the cylindrical interpolation was issued.</li> <li>• A T command was issued during the cylindrical interpolation or pole coordinate interpolation mode.</li> <li>• A movement command was issued when the plane was not selected just before or after the G07.1 command.</li> <li>• A plane selection command was issued during the pole coordinate interpolation mode.</li> <li>• Cylindrical interpolation or pole coordinate interpolation was commanded during tool radius compensation.</li> <li>• The G16 plane in which the radius value of a cylinder is 0 was specified.</li> <li>• A cylindrical interpolation or pole coordinate interpolation command was issued during program coordinate rotation (G68).</li> </ul>	<ul style="list-style-type: none"> <li>• Check the program.</li> <li>• Before issuing G12.1, issue G40 or G97.</li> <li>• Before issuing G12.1, issue a T command.</li> <li>• Before issuing G13.1, issue G40.</li> <li>• Specify the radius value of a cylinder other than 0, or specify the X axis's current value other than 0 before issuing G12.1/G16.</li> </ul>
<b>P486</b>	<b>MILLING ERROR</b> <ul style="list-style-type: none"> <li>• The milling command was issued during the mirror image (when parameter or external input is turned ON).</li> <li>• Pole coordinate interpolation, cylindrical interpolation or milling interpolation was commanded during facing turret mirror image.</li> <li>• The start command of the cylindrical interpolation or polar coordinate interpolation was issued during the normal line control.</li> </ul>	<ul style="list-style-type: none"> <li>• Check the program.</li> </ul>

## Appendix 2. Program Error

Error No.	Details	Remedy
<b>P511</b>	<b>SYNC CODE ERR</b> <ul style="list-style-type: none"> <li>Two or more synchronization M codes were commanded in the same block.</li> <li>The synchronization M code and "!" code were commanded in the same block.</li> </ul>	<ul style="list-style-type: none"> <li>Check the program.</li> </ul>
<b>P600</b>	<b>NO AUTO TLM.</b> An automatic tool length measurement command (G37) was execute though there are no such command specifications.	<ul style="list-style-type: none"> <li>Check the specifications.</li> </ul>
<b>P601</b>	<b>NO SKIP SPEC.</b> A skip command (G31) was issued though there are no such command specifications.	<ul style="list-style-type: none"> <li>Check the specifications.</li> </ul>
<b>P602</b>	<b>NO MULTI SKIP</b> A multiple skipping command (G31.1, G31.2 or G31.3) was issued though there are no such command specifications.	<ul style="list-style-type: none"> <li>Check the specifications.</li> </ul>
<b>P603</b>	<b>SKIP SPEED 0</b> The skip speed is 0.	<ul style="list-style-type: none"> <li>Specify the skip speed.</li> </ul>
<b>P604</b>	<b>TLM ILL. AXIS</b> No axis or more than one axis was specified in the automatic tool length measurement block.	<ul style="list-style-type: none"> <li>Specify only one axis.</li> </ul>
<b>P605</b>	<b>T-CMD IN BLOCK</b> The T code is in the same block as the automatic tool length measurement block.	<ul style="list-style-type: none"> <li>Specify this T code before the block.</li> </ul>
<b>P606</b>	<b>NO T-CMD BEFOR</b> The T code was not yet specified in automatic tool length measurement.	<ul style="list-style-type: none"> <li>Specify this T code before the block.</li> </ul>
<b>P607</b>	<b>TLM ILL. SIGNAL</b> Before the area specified by the D command or decelerating area parameter d, the measurement position arrival signal went ON. The signal remains OFF to the end.	<ul style="list-style-type: none"> <li>Check the program.</li> </ul>
<b>P608</b>	<b>SKIP ERROR (CC)</b> A skip command was specified during radius compensation processing.	<ul style="list-style-type: none"> <li>Specify a radius compensation cancel (G40) command' or remove the skip command.</li> </ul>

## Appendix 2. Program Error

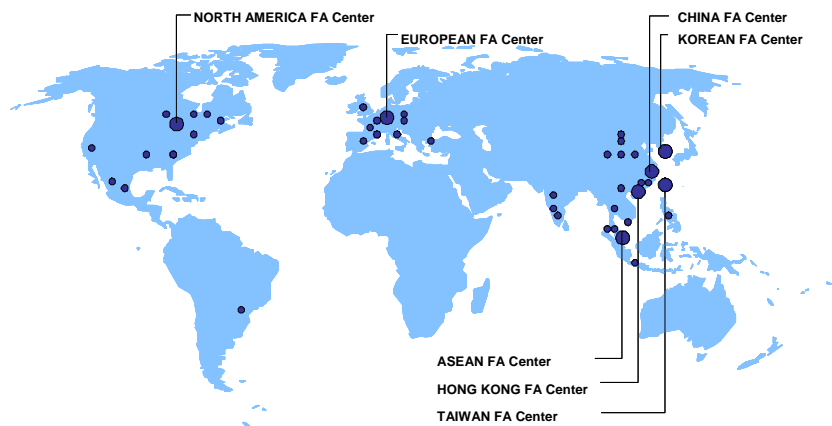
Error No.	Details	Remedy
<b>P610</b>	<b>ILLEGAL PARA.</b> • The parameter setting is not correct.	<ul style="list-style-type: none"> <li>• Check the program.</li> <li>• Check the parameter.</li> </ul>
<b>P612</b>	<b>EXP. ERROR</b> A movement command for exponential function interpolation was issued during facing turret mirror image.	<ul style="list-style-type: none"> <li>• Check the program.</li> </ul>
<b>P700</b>	<b>CMD-VALUE ILL.</b> Spindle synchronization was commanded to a spindle that is not connected serially.	<ul style="list-style-type: none"> <li>• Check the program.</li> <li>• Check the parameter.</li> </ul>
<b>P900</b>	<b>NO TANZ. SPEC</b> A normal line control command (G40.1, G41.1, G42.1) was issued when the normal line control specifications were not provided.	<ul style="list-style-type: none"> <li>• Check the specifications.</li> </ul>
<b>P901</b>	<b>TAN. AXIS G92</b> A coordinate system preset command (G92) was issued to a normal line control axis during normal line control.	<ul style="list-style-type: none"> <li>• Check the program.</li> </ul>
<b>P902</b>	<b>TAN. AXIS LINE</b> <ul style="list-style-type: none"> <li>• The normal line control axis was set to a linear axis.</li> <li>• The normal line control axis was set to the linear type rotary axis II axis.</li> <li>• The normal line control axis has not been set.</li> <li>• The normal line control axis was the same as the plane selection axis.</li> </ul>	<ul style="list-style-type: none"> <li>• Correct the normal line control axis.</li> </ul>
<b>P903</b>	<b>PLANE CHG (TAN)</b> The plane selection command (G17, G18, G19) was issued during normal line control.	<ul style="list-style-type: none"> <li>• Delete the plane selection command (G17, G18, G19) from the program for normal line control.</li> </ul>
<b>P990</b>	<b>PREPRO S/W ERR</b> Combining commands that required pre-reading (nose R offset, corner chamfering/corner rounding, geometric I, geometric IB, and fixed cycle for compound lathe) resulted in eight or more pre-read blocks.	<ul style="list-style-type: none"> <li>• Reduce the number of commands that require pre-reading or delete such commands.</li> </ul>



## Revision History

Date of revision	Manual No.	Revision details
Mar. 2006	IB(NA)1500173-A	First edition created.

# Global service network



## North America FA Center (MITSUBISHI ELECTRIC AUTOMATION INC.)

**Illinois CNC Service Center**  
500 CORPORATE WOODS PARKWAY, VERNON HILLS, IL. 60061, U.S.A.  
TEL: +1-847-478-2500 (Se) FAX: +1-847-478-2650 (Se)

**California CNC Service Center**  
5665 PLAZA DRIVE, CYPRESS, CA. 90630, U.S.A.  
TEL: +1-714-220-4796 FAX: +1-714-229-3818

**Georgia CNC Service Center**  
2810 PREMIERE PARKWAY SUITE 400, DULUTH, GA., 30097, U.S.A.  
TEL: +1-678-258-4500 FAX: +1-678-258-4519

**New Jersey CNC Service Center**  
200 COTTONTAIL LANE SOMERSET, NJ. 08873, U.S.A.  
TEL: +1-732-560-4500 FAX: +1-732-560-4531

**Michigan CNC Service Satellite**  
2545 38TH STREET, ALLEGAN, MI., 49010, U.S.A.  
TEL: +1-847-478-2500 FAX: +1-269-673-4092

**Ohio CNC Service Satellite**  
62 W. 500 S., ANDERSON, IN., 46013, U.S.A.  
TEL: +1-847-478-2608 FAX: +1-847-478-2690

**Texas CNC Service Satellite**  
1000, NOLEN DRIVE SUITE 200, GRAPEVINE, TX. 76051, U.S.A.  
TEL: +1-817-251-7468 FAX: +1-817-416-1439

**Canada CNC Service Center**  
4299 14TH AVENUE MARKHAM, ON. L3R 0J2, CANADA  
TEL: +1-905-475-7728 FAX: +1-905-475-7935

**Mexico CNC Service Center**  
MARIANO ESCOBEDO 69 TLALNEPANTLA, 54030 EDO. DE MEXICO  
TEL: +52-55-9171-7662 FAX: +52-55-9171-7698

**Monterrey CNC Service Satellite**  
ARGENTINA 3900, FRACC. LAS TORRES, MONTERREY, N.L., 64720, MEXICO  
TEL: +52-81-8365-4171 FAX: +52-81-8365-4171

**Brazil MITSUBISHI CNC Agent Service Center**  
(AUTOMOTION IND. COM. IMP. E EXP. LTDA.)  
ACESSO JOSE SARTORELLI, KM 2.1 18550-000 BOITUVA – SP, BRAZIL  
TEL: +55-15-3363-9900 FAX: +55-15-3363-9911

## European FA Center (MITSUBISHI ELECTRIC EUROPE B.V.)

**Germany CNC Service Center**  
GOTHAER STRASSE 8, 40880 RATINGEN, GERMANY  
TEL: +49-2102-486-0 FAX: +49-2102486-591

**South Germany CNC Service Center**  
KURZE STRASSE, 40, 70794 FILDERTSTADT-BONNLANDEN, GERMANY  
TEL: +49-711-3270-010 FAX: +49-711-3270-0141

**France CNC Service Center**  
25, BOULEVARD DES BOUVETS, 92741 NANTERRE CEDEX FRANCE  
TEL: +33-1-41-02-83-13 FAX: +33-1-49-01-07-25

**Lyon CNC Service Satellite**

**U.K CNC Service Center**  
TRAVELLERS LANE, HATFIELD, HERTFORDSHIRE, AL10 8XB, U.K.  
TEL: +44-1707-282-846 FAX: +44-1707-278-992

**Italy CNC Service Center**  
ZONA INDUSTRIALE VIA ARCHIMEDE 35 20041 AGRATE BRIANZA, MILANO ITALY  
TEL: +39-039-60531-342 FAX: +39-039-6053-206

**Spain CNC Service Satellite**  
CTRA. DE RUBI, 76-80 -APDO.420 08190 SAINT CUGAT DEL VALLES, BARCELONA SPAIN  
TEL: +34-935-65-2236 FAX:

**Turkey MITSUBISHI CNC Agent Service Center**  
(GENEL TEKNİK SİSTEMLER LTD. STI.)  
DARULACEZE CAD. FAMES İS MERKEZİ A BLOK NO.43 KAT:2 80270 OKMEYDANI İSTANBUL, TURKEY  
TEL: +90-212-320-1640 FAX: +90-212-320-1649

**Poland MITSUBISHI CNC Agent Service Center (MPL Technology Sp. z. o. o)**  
UL SŁICZNA 34, 31-444 KRAKOW, POLAND  
TEL: +48-12-632-28-85 FAX:

**Wroclaw MITSUBISHI CNC Agent Service Satellite (MPL Technology Sp. z. o. o)**  
UL KOBIERZYCKA 23, 52-315 WROCLAW, POLAND  
TEL: +48-71-333-77-53 FAX: +48-71-333-77-53

**Czech MITSUBISHI CNC Agent Service Center**  
(AUTOCONT CONTROL SYSTEM S.R.O. )  
NEMOCNICNI 12, 702 00 OSTRAVA 2 CZECH REPUBLIC  
TEL: +420-596-152-426 FAX: +420-596-152-112

## ASEAN FA Center (MITSUBISHI ELECTRIC ASIA PTE. LTD.)

**Singapore CNC Service Center**  
307 ALEXANDRA ROAD #05-01/02 MITSUBISHI ELECTRIC BUILDING SINGAPORE 159943  
TEL: +65-6473-2308 FAX: +65-6476-7439

**Thailand MITSUBISHI CNC Agent Service Center (F. A. TECH CO., LTD)**  
898/19,20,21,22 S.V. CITY BUILDING OFFICE TOWER 1 FLOOR 12,14 RAMA III RD BANGPONGPANG, YANNAWA, BANGKOK 10120, THAILAND  
TEL: +66-2-682-6522 FAX: +66-2-682-6020

**Malaysia MITSUBISHI CNC Agent Service Center**  
(FLEXIBLE AUTOMATION SYSTEM SDN. BHD.)  
60, JALAN USJ 10/1B 47620 UEP SUBANG JAYA SELANGOR DARUL EHSAN MALAYSIA  
TEL: +60-3-5631-7605 FAX: +60-3-5631-7636

**Johor MITSUBISHI CNC Agent Service Satellite**  
(FLEXIBLE AUTOMATION SYSTEM SDN. BHD.)  
NO. 16, JALAN SHAHBANDAR 1, TAMAN LINGKU TUN AMINAH, 81300 SKUDAI, JOHOR MALAYSIA  
TEL: +60-7-557-8218 FAX: +60-7-557-3404

**Indonesia MITSUBISHI CNC Agent Service Center**  
(PT. AUTOTEKNINDO SUMBER MAKMUR)  
WISMA NUSANTARA 14TH FLOOR JL. M.H. THAMRIN 59, JAKARTA 10350 INDONESIA  
TEL: +62-21-3917-144 FAX: +62-21-3917-164

**India MITSUBISHI CNC Agent Service Center (MESSUNG SALES & SERVICES PVT. LTD.)**  
B-36FF, PAVANA INDUSTRIAL PREMISES M.I.D.C., BHOASRI PUNE 411026, INDIA  
TEL: +91-20-2711-9484 FAX: +91-20-2712-8115

**BANGALORE MITSUBISHI CNC Agent Service Satellite**  
(MESSUNG SALES & SERVICES PVT. LTD.)  
S 615, 6TH FLOOR, MANIPAL CENTER, BANGALORE 560001, INDIA  
TEL: +91-80-509-2119 FAX: +91-80-532-0480

**Delhi MITSUBISHI CNC Agent Parts Center (MESSUNG SALES & SERVICES PVT. LTD.)**  
1197, SECTOR 15 PART-2, OFF DELHI-JAIPUR HIGHWAY BEHIND 32ND MILESTONE GURGAON 122001, INDIA  
TEL: +91-98-1024-8895 FAX:

**Philippines MITSUBISHI CNC Agent Service Center**  
(FLEXIBLE AUTOMATION SYSTEM CORPORATION)  
UNIT No.411, ALABANG CORPORATE CENTER KM 25. WEST SERVICE ROAD SOUTH SUPERHIGHWAY, ALABANG MUNTINLUPA METRO MANILA, PHILIPPINES 1771  
TEL: +63-2-807-2416 FAX: +63-2-807-2417

**Vietnam MITSUBISHI CNC Agent Service Center (SA GIANG TECHNO CO., LTD)**  
47-49 HOANG SA ST. DAKAO WARD, DIST.1 HO CHI MINH CITY, VIETNAM  
TEL: +84-8-910-4763 FAX: +84-8-910-2593

## China FA Center (MITSUBISHI ELECTRIC AUTOMATION (SHANGHAI) LTD.)

**China CNC Service Center**  
2/F., BLOCK 5 BLDG. AUTOMATION INSTRUMENTATION PLAZA, 103 CAOBAO RD. SHANGHAI 200233, CHINA  
TEL: +86-21-6120-0808 FAX: +86-21-6494-0178

**Shenyang CNC Service Center**  
TEL: +86-24-2397-0184 FAX: +86-24-2397-0185

**Beijing CNC Service Satellite**  
9/F, OFFICE TOWER1, HENDERSON CENTER, 18 JIANGUOMENNEI DAJIE, DONGCHENG DISTRICT, BEIJING 100005, CHINA  
TEL: +86-10-6518-8830 FAX: +86-10-6518-8030

**China MITSUBISHI CNC Agent Service Center**  
(BEIJING JIAYOU HIGHTECH TECHNOLOGY DEVELOPMENT CO.)  
RM 709, HIGH TECHNOLOGY BUILDING NO.229 NORTH SI HUAN ZHONG ROAD, HAIDIAN DISTRICT, BEIJING 100083, CHINA  
TEL: +86-10-8288-3030 FAX: +86-10-6518-8030

**Tianjin CNC Service Satellite**  
RM909, TAIHONG TOWER, NO220 SHIZILIN STREET, HEBEI DISTRICT, TIANJIN, CHINA 300143  
TEL: +86-22-2653-9090 FAX: +86-22-2635-9050

**Shenzhen CNC Service Satellite**  
RM02, UNIT A, 13/F, TIANAN NATIONAL TOWER, RENMING SOUTH ROAD, SHENZHEN, CHINA 518005  
TEL: +86-755-2515-6691 FAX: +86-755-8218-4776

**Changchun Service Satellite**  
TEL: +86-431-50214546 FAX: +86-431-5021690

**Hong Kong CNC Service Center**  
UNIT A, 25/F RYODEN INDUSTRIAL CENTRE, 26-38 TA CHUEN PING STREET, KWAI CHUNG, NEW TERRITORIES, HONG KONG  
TEL: +852-2619-8588 FAX: +852-2784-1323

## Taiwan FA Center (MITSUBISHI ELECTRIC TAIWAN CO., LTD.)

**Taichung CNC Service Center**  
NO.8-1, GONG YEH 16TH RD., TAICHUNG INDUSTRIAL PARK TAICHUNG CITY, TAIWAN R.O.C.  
TEL: +886-4-2359-0688 FAX: +886-4-2359-0689

**Taipei CNC Service Satellite**  
TEL: +886-4-2359-0688 FAX: +886-4-2359-0689

**Tainan CNC Service Satellite**  
TEL: +886-4-2359-0688 FAX: +886-4-2359-0689

## Korean FA Center (MITSUBISHI ELECTRIC AUTOMATION KOREA CO., LTD.)

**Korea CNC Service Center**  
DONGSEO GAME CHANNEL BLDG. 2F. 660-11, DEUNGCHON-DONG KANGSEO-KU SEOUL, 157-030 KOREA  
TEL: +82-2-3660-9607 FAX: +82-2-3663-0475

## **Notice**

Every effort has been made to keep up with software and hardware revisions in the contents described in this manual. However, please understand that in some unavoidable cases simultaneous revision is not possible.

Please contact your Mitsubishi Electric dealer with any questions or comments regarding the use of this product.

## **Duplication Prohibited**

This manual may not be reproduced in any form, in part or in whole, without written permission from Mitsubishi Electric Corporation.

© 2006 MITSUBISHI ELECTRIC CORPORATION  
ALL RIGHTS RESERVED.



MODEL	EZMotion-NC E60/E68 Series
MODEL CODE	008—399
Manual No.	IB-1500173(ENG)-A

Specifications subject to change without notice.  
Printed in Japan on recycled paper.