OSP5020M
OSP500M-G

PROGRAMMING (4th Edition)

Pub. No. 3336-E-R3 (ME33-002-R4)

OKUMA

SAFETY PRECAUTIONS

The machine is equipped with a variety of safety devices designed to protect prevent accidents and injuries from occurring. Still, operators must use the machine carefully and learn not to rely on these safety devices.

This section describes the general safety precautions required for machine operation. Every operator must read this section carefully and observe the guidelines it describes.

1. Before Turning on the Power

(1) Make sure that doors to the operation panel and the electric control cabinet are closed.

(2) Do not put any obstacles on or around the machine.

(3) Turn on the main power switch before turning on the **CONTROL ON** switch on the operation panel.

(4) Make sure that there is not dangerous elements around the machine.

2. In Manual or Continuous Operation

(1) Always follow the operation manual.

(2) Never operate the machine without closing the front shield.

(3) In initial grinding, check for operating conditions and interference according to the steps from no-load run to single block grinding to continuous operation.

(4) Ensure the safety before rotating or moving the spindle.

(5) Check the tool offset values.

(6) Check the zero offset values.

(7) Ensure that the spindle speed and feedrate override switches are set at 100 %. Operate the machine carefully and stop it immediately in case of abnormality. If necessary, set the feedrate override switch to 5, 10, and 50 % etc.

(8) Never attempt to touch the spindle or tool in rotation.

(9) Do not touch the spindle or the tool when indexing or changing speed. The spindle may turn.

(10) Never apply a rotating torque to the spindle by operations such as retightening of milling chuck before the spindle is completely indexed. The spindle may turn.

(11) Do not turn the spindle with the tool or other parts loosely fixed.

(12) Be sure to stop the spindle tool and use a brush or broom when removing chips adhering to the spindle.

(13) Cutting operation causes scatter of chips, coolant, and in some cases fragments of the tool.

Be sure to close the cover and stay away from the spindle.

(14) Load a workpiece securely on the table. After loading, ensure that there are no tightening tools left on the table.

(15) Do not move the table when an operator is on it.

(16) When two or more persons operate the machine, each operator should make sure of the other operators' safety.

(17) Movements of ATC, APC and AAC are large and dangerous. Operators must keep away from these units while in operation.

(18) If the ATC, APC or AAC stops in the middle of operation for some reasons and you inspect the machine without turning off the power, always secure an emergency escape zone and never touch the problem unit directly by your hands. The unit may start moving suddenly.

3. At the End of the Day

(1) Clean the area around the machine.

(2) Return the APC, ATC and AAC to the preset retracting position.

(3) Make sure that all the power switches are turned off before leaving your workplace.

(4) Turn off the **CONTROL ON** switch on the operation panel before turning off the main power switch.

4. In Maintenance Inspection or Abnormality Treatment

(1) Press the **EMERGENCY STOP** switch in case of emergency.

(2) The person responsible for maintenance should grasp the points of inspection or repair before maintenance work.

(3) When two or more persons conducts maintenance work, they should cooperate each other by exchanging signs and information.

(4) Use fuses or other parts of the specified rating for replacement.

(5) Turn off the power before replacing parts or checking wiring connections.

(6) After removing parts for inspection or repair, do not forget to remount and securely tighten them with the screws.

(7) Use calibrated measuring instruments in measuring inspections such as voltage check.

(8) Use great care in handling the following high-voltage units:

- Main breaker
- Servo drive unit
- VAC drive unit
- Power cables

(9) Do not leave any flammables or metallic parts inside the operation panel or the terminal box.

For the safety precautions in machine operation, refer to the Instruction Manual for the NC Machining Center.

5. Symbols

This manual uses the following symbols to emphasize the items which must be strictly observed:

REMARKS : Precaution in machine operations.

Default of this item affects the machining accuracy and smooth operation.

CAUTION : Default of this item may cause faulty machine operation, machine damage, or operator injury.

More care must be exercised to this item than the items indicated under "REMARKS".

WARNING :

Default of this item may cause not only machine damage but also serious accidents that affect human life.

Sufficient care must be taken to this item.

TABLE OF CONTENTS

PAGE

PAGE

PAGE

PAGE

PAGE

SECTION 1 INTRODUCTION

1. Features

The OSP5020M/OSP500M-G series consists of high performance numerically-controlled units (CNC) capable of handling a variety of software. Based on a design philosophy not found in conventional NC systems, the OSP5020M/OSP500M-G contains a number of functions and features not present even in our OSP2300M/3000M/5000M/5000M-G series which have also been pace-setters in the world of numerical controllers.

(1) Bubble Memory Storage for Variable CNC Software

The use of bubble memories for storage makes it possible to change software. NC control and system software functions may be added to the NC unit, even after delivery, whenever they may be desired.

(2) Across the Board Enrichment of Die Machining Functions

(a) Storage of large capacity tapes, several thousand meters in length, is now possible. In addition, machining programs may be entered on a floppy disk (3.5 inches), thus doing away with the bother of paper tape operations. (Optional)

(b) Restarting operations have been improved by enhancing the sequence restart with a return search function (internally executed) and a mid-point start-up function (sequence number search).

(c) A variety of locking functions, such as an STM lock, axis command cancel, and a machine lock have been provided.

(d) A pulse handle for up to three axes and a pulse handle intervention function allow the machining talents of the individual operator full scope. (Optional)

(e) The availability of manual interruption improves operation. A three-dimensional tool offset function has also been provided.

(3) Multi-task Processing

An adopted multi-task system permits the reading, punching, and editing of programs even during NC program execution, which greatly increases operation efficiency.

(4) Greatly Perfected Automatic and Untended Operations (Optional)

(a) The use of schedule programs makes the machining of several types of workpieces possible.

(b) Monitoring various stages of production is possible.

(c) Also, the following functions are provided: adaptive control function, tool control functions (tool life management, tool breakage detection, automatic spare tool selection, etc.), overload detection function, automatic work gauging function, automatic tool length compensation function, automatic zero offset function.

(5) Simplified Programming

The use of mnemonics, such as GOTO and IF, simplifies the programming and checking of programs. Subprograms and common/local variables permit more efficient generation of similar programs. In addition, automatic programming functions, such as coordinate computation, area machining, geometry enlargement/reduction, and coordinate shift and rotation, permit programs to be simplified and shortened. (Optional)

(6) Improved System Expandability

A file management function, the ability to serve as an FMS cell controller, and DNC communications have also been provided. The system has been designed to allow the addition of hardware and software updates so that the system will never lag behind changes in technology.
In addition, a user-interactive function and a built-in automatic programming system have been provided.

2. Explanation

This programming manual contains instructions for programming the OSP5020M/OSP500M-G and precautions that should be observed when programming. A careful reading of the manual will be of great assistance in obtaining the full benefit of all the superior functions the machine has to offer. For the most complete understanding, this manual should be read in conjunction with the "Operation Manual for OSP5020M/OSP500M-G", as the two manuals are very closely related.

Machine improvements that have been made since this manual was written may occasionally result in slight differences between the manual and the machine.

SECTION 2 PROGRAM CONFIGURATIONS

1. Program Types and Extensions

Four types of programs are used: schedule programs, main programs, subprograms, and library programs.

(1) Schedule Program (Optional)

The schedule program specifies how many times the program is to be executed and the execution order of the main programs, when more than one type of workpiece is being machined using a pallet changer or other automatic loading and unloading equipment. This feature allows the realization of untended machine operation.
A program name may not be used.
The END code must be specified at the end of the schedule program.
For details, refer to SECTION 12, 2. "Schedule Program".

(2) Main Program

The main program contains a series of commands to machine one type of workpiece. Subprograms can be called from the main program to simplify programming.
The program name, which must start with an "O", is required at the beginning of the main program.
The M02 or M30 command must be specified at the end of the main program.

(3) Subprogram

A subprogram can be called from the main program or another subprogram.
There are two types of subprograms: those written and supplied by Okuma (maker subprogram), and those written by the customer (user subprogram) .
The program name, which must start with "O", is required at the beginning of the subprogram.
The RTS command must be specified at the end of the subprogram.
For details, refer to SECTION 10, "SUBPROGRAM FUNCTIONS".

(4) Library Program

Subprograms and G code macros which are used frequently may be registered as library programs. Library programs are stored in the operation buffer when the power is turned on and can be accessed at any time.
For details, refer to SECTION 12, 3. "Library Program".

(5) Extensions

An extension is added to the file name when the program is stored to assist the OSP5020M/OSP500M-G in file management. The format is shown below:

<File Name Format>

Main file name*	Extension
(The file name consists of a character string of up to 16 characters which begins with an alphabetic character.)	

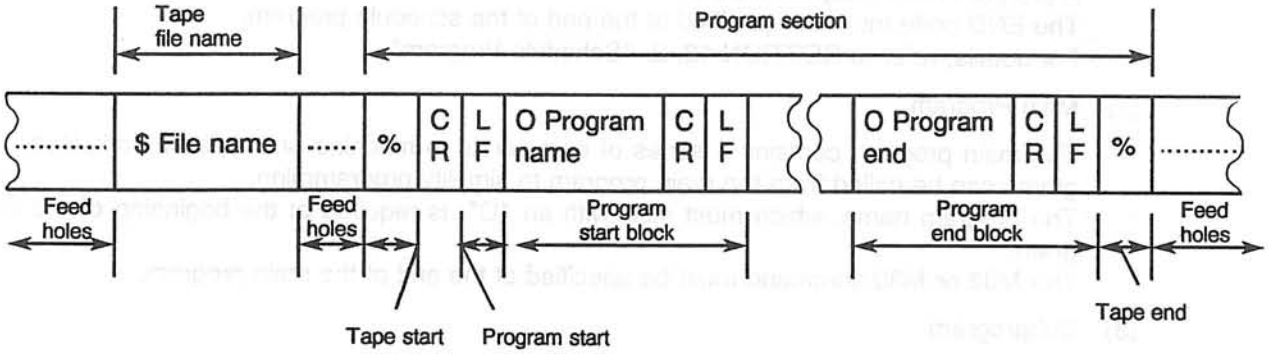
List of extension names:

1) SDF schedule program file
2) MIN main program file
3) MSB maker's subprogram file
4) SSB system subprogram file
5) SUB user's subprogram file
6) LIB library subprogram file

For details on file management, see SECTION 12, 4. "File Management".

2. Program Tape Format

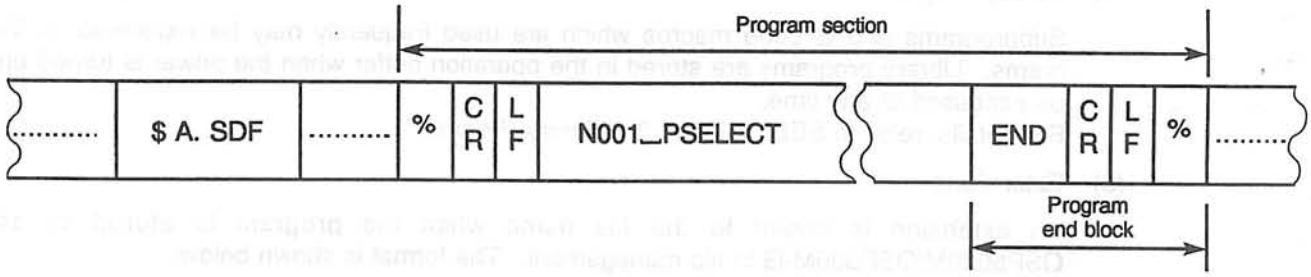
The general format of an OSP5020M/OSP500M-G machining program is shown below:



(1) In the EIA code, one EOB (CR) character may be used instead of CR and LF used in the ISO code. In the ISO code, CR may be omitted.

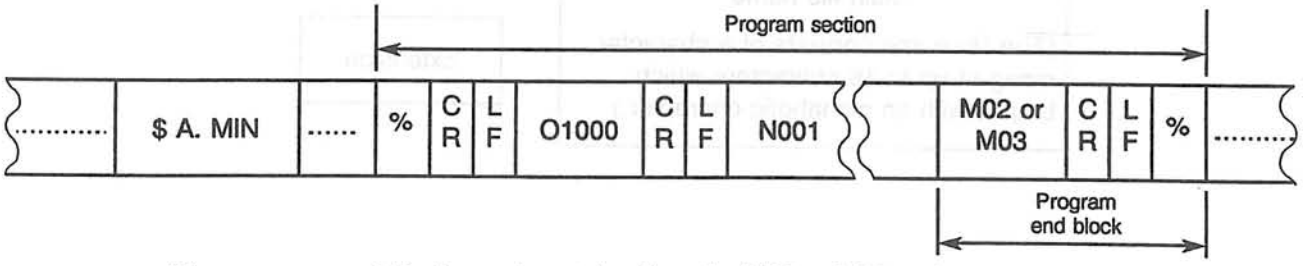
(2) The program section must begin and end with a percent sign "%".
Three types of programs are handled by the OSP5020M/OSP500M-G. Their tape formats are shown below:

(a) Schedule Programs



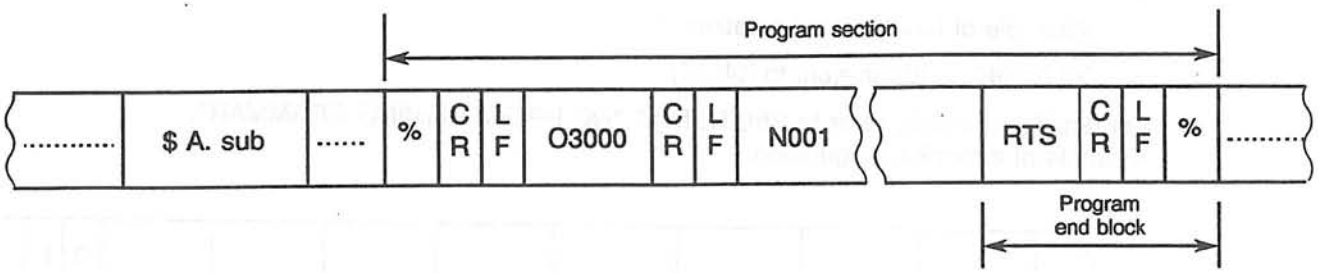
No program name may be punched in a schedule program.
The program end block must contain the "END" code.

(b) Main Programs



The program end block must contain either the M02 or M30 code.

(c) Subprograms



The RTS command must be specified at the end of the subprogram.

3. Block Configuration

A program is composed of several commands, one unit of which is referred to as a block. An end of block (EOB) code is placed as a delimiter between blocks.

(1) The end of block code is different depending on the coding system selected, ISO or EIA:

ISO ..... "LF"
EIA ..... "CR"

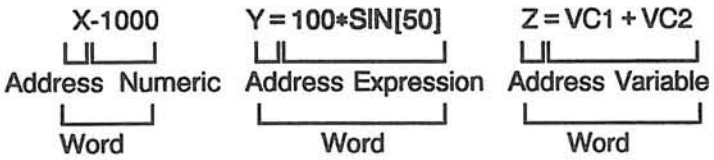
(2) A block is comprised of several words.

(3) Up to 158 characters are allowed in one block.

4. Word Configuration

A word is defined as an address character followed by a group of numeric values, an expression, or a variable name. If the word consists of an expression or a variable, the address character must be followed by an equal sign "=".

Examples:



(1) An address character may consist of one of the alphabetic characters A through Z to specify respective contents. An extended address character, consisting of two alphabetic characters may also be used.

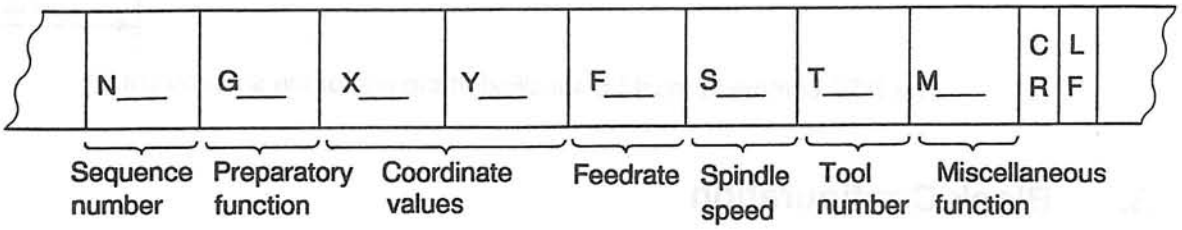
(2) Refer to Section 11, 1-2. "Variable Function" for more information on variables.

(3) Hexadecimal numeric values may also be used.

Example of hexadecimal notation:

X#1000H (Equivalent to X4096)

For details on words, refer to SECTION 13 "NC PROGRAMMING GRAMMAR".
Example of a block configuration:



5. Numeric Value (Inch/Metric Convertible)

The unit system includes millimeters (inches), degrees, and seconds, along with the decimal specification.

(1) Notes on Decimal Point Specification

(a) The address characters O, N, G, and M cannot use numeric values with decimal points.

(b) If no decimal point is included in the numeric value, it is assumed that the value is followed by a decimal point.

(c) Real number values assigned to the integer address characters S, T, H, D, and Q are rounded down to the lower integer.

(d) Real number values assigned to address characters other than S, T, H, D, and Q are subjected to standard rounding.

(2) Setting the Unit System

The unit system is determined by a bit combination of the NC optional parameters (bit) No. 3 bit 0 through bit 7 and No. 4, bit 0. The following table lists the possible combinations.

Parameter No.	Bit No.	Contents	Set At 1	Set At 0	Factory Set Initial Value
3	0	Selects "inches" OR "mm" for length unit system.	Inch	mm	0
	1	Selects 1 mm, 1 inch, 1 degree, and 1 second for the input unit system.	Yes	Defers to settings at bits 2,3,4,5,7.	1
	2	Selects 0.01 mm OR 0.001 mm for length unit system.	0.01 mm	0.001 mm	0
	3	Selects either 0.1 mm/min and 0.01 in/min OR 1 mm/min and 0.1 in/min for the speed input system.	0.1 mm/min 0.01 in.min	1 mm/min 0.1 in/min	0
	4	Selects either 0.001 mm/rev and 0.0001 in/rev OR 0.01 mm/rev and 0.001 in/rev for the speed input system.	0.001 mm/rev 0.0001 in/rev	0.01 mm/rev 0.001 in/rev	0
	5	Selects 0.01 sec OR 0.1 sec for the time input unit.	0.01 sec	0.1 sec	0
	6	Selects 1 mm, 1 inch, 1 degree and 1 second for the unit value with decimal point.	Yes	Defers to settings at bits 1,2,3,4,5,7.	0
	7	Selects 0.001 sec OR 0.1 sec for the time input system. (See note below.)	0.001 sec	0.1 sec	0
4	0	Selects 0.001 degree OR 0.0001 degree for the angle unit system of rotation around an axis.	0.0001 deg	0.001 deg	0

Note 1: The unit of input time is always 0.01 sec when the setting of bit 5 is "1".

Note 2: Changing the setting of bit 0 is only effective on machines with the optional system conversion function.

(3) Examples of Unit Systems

(a) Metric System

No.4	Parameter Bit No.3								Length	Angle	Feedrate	Feedrate	Time Period
0	7	6	5	4	3	2	1	0	(mm)	(deg)	(mm/min)	(mm/rev)	(sec)
*	*	*	*	*	*	*	1	0	1	1	1	1	1
0	0	*	0	0	0	0	0	0	0.001	0.001	1	0.01	0.1
0	0	*	0	0	0	1	0	0	0.01	0.001	1	0.01	0.1
0	0	*	0	0	1	0	0	0	0.001	0.001	0.1	0.01	0.1
1	0	*	0	1	0	0	0	0	0.001	0.0001	1	0.001	0.1
1	*	*	1	0	0	0	0	0	0.001	0.0001	1	0.01	0.01
1	1	*	0	0	0	0	0	0	0.001	0.0001	1	0.01	0.001

(b) Inch System (Optional)

No.4	Parameter Bit No.3								Length	Angle	Feedrate	Feedrate	Time Period
0	7	6	5	4	3	2	1	0	(mm)	(deg)	(mm/min)	(mm/rev)	(sec)
*	0	*	*	*	*	*	1	1	1	1	1	1	1
0	0	*	0	0	0	*	0	1	0.0001	0.001	0.1	0.001	0.1
0	0	*	0	0	1	*	0	1	0.0001	0.001	0.01	0.001	0.1
0	0	*	0	1	0	*	0	1	0.0001	0.001	0.1	0.0001	0.1
1	*	*	1	0	0	*	0	1	0.0001	0.0001	0.1	0.001	0.01
1	1	*	0	0	0	*	0	1	0.0001	0.0001	0.1	0.001	0.001

Note: In the tables above, the asterisk "" is used to denote either settings of "0" or "1".*

*μm/mm Unit System

The μm/mm unit system is defined as a unit system in which numerical values expressed in real numbers are handled in units of mm (inches) and those expressed in integer numbers are handled in units of microns (1/10000 inches). This unit system can be selected by setting NC optional parameter (bit) No. 3, bit 6 to "1".

The unit is determined depending on whether or not a decimal point has been specified. When a decimal is specified on a numerical value, the value is handled in units of mm (inches). When a decimal point is not included in the value, it is handled in units of microns (1/10000 inches).

Example :

X100. → 100 mm
X100 → 100 μm

If a numerical value is assigned to a variable or expression, the value is handled as a real number in mm (inch), regardless of the decimal point.

Example : Local variables

PX = 100
X = PX → 100 mm
(100 is not handled as 100 μm)

PX = 100.
X = PX → 100 mm

If a numerical value is assigned to a variable, the value is handled as a real number having significant digits.

(4) The table below shows examples of how numerical values are handled. (Unit: mm/μm)

Command	Numerical Value	Metrik System Factor
X 100	100 μm	—
X = 100	100 μm	—
X 100.	100 mm	decimal point
X = 100.	100 mm	decimal point
X 100 + 100	200 mm	expression
X 100. + 100	200 mm	expression
X = 100 + 100.	200 mm	expression
X 100 + 100 * 2	300 mm	expression
X = 100 + 100 * 2.	300 mm	expression
X = 100 + 100 * 2.5	350 mm	expression
PX = 100 X = 100 + PX	200 mm	variable
PX = 100. X = 200 − PX	100 mm	variable
X = 200 − 100	100 mm	expression
X −100	−100 μm	—
X −100.	−100 mm	decimal point
X +100	100 μm	—
X +100.	100 mm	decimal point
X = ROUND [100]	100 mm	expression
X = FIX [100.]	100 mm	expression
X = FUP [−100]	−100 mm	expression
LA1 = 4 F = FIX [LA1]	4 mm/min	variable

(5) Programmable Range of Address Characters

The programmable range of numerical values of individual address characters are shown in the following table:

Address	Function	Programmable Range		Remarks
		Metric	Inch	
O	Program number	0000 - 9999	Same as metric	Alphabetic characters available
N	Sequence number	00000 - 99999	Same as metric	Alphabetic characters available
G	Preparatory function	0 - 399	Same as metric	Mnemonics available
X,Y,Z, U,V,W	Coordinate values (linear axis)	±99999.999 mm	±9999.9999 inch	
I,J,K	Coordinate values of center of arc	±99999.999 mm	±9999.9999 inch	
R	Radius of arc	±99999.999 mm	±9999.9999 inch	
A,B,C	Coordinate values of rotary axis	±360.0000 deg.	Same as metric	Multi-rotary specification ±9999.9999 deg.
F	Feedrate per minute	0.1 - 24000.0 mm/min	0.01 - 2400.0 inch/min	
	Feedrate per revolution	0.001 - 500.000 mm/rev	0.0001 - 50.0000 inch/rev	
	Dwell time period	0.001 - 99999.999 sec	Same as metric	
S	Spindle speed	0 - 9999	Same as metric	
T	Tool number	1 through maximum tool data number	Same as metric	
M	Miscellaneous function	0 - 399	Same as metric	
H	Tool length offset number	1 through maximum tool data number	Same as metric	
D	Cutter radius compensation number	1 through maximum tool data number	Same as metric	
P	Dwell time period (during fixed cycle)	0.001 - 99999.999 sec	Same as metric	
Q	Second dwell time period (during fixed cycle)	0.001 - 99999.999 sec	Same as metric	
	Depth of cut (during fixed cycle)	0 - 99999.999 mm	0 - 9999.9999 inch	
	Repetition time (schedule program)	1 - 9999	Same as metric	
R	Cut starting level (during fixed cycle)	±99999.999 mm	±9999.9999 inch	

Note: An alarm is activated when any of the following addresses are specified more than once within a block: X, Y, Z, U, V, W, A, B, C, F.

6. Program Name

A program name or program number is assigned to each program. Operations can be carried out just by calling the program.

If the program name contains only alphabetic characters, it is called a program label. If it contains only numbers, it is called a program number. This manual refers to both as program names.

(1) Program Name Designation

(a) Input alphabets (A - Z) or numbers (0 - 9) following after "O"

(b) Up to four characters can be used

(c) Alphabetic characters may not come after a numeric character in the program name. The beginning of the program name must be an alphabetic character.

(d) All of the four characters may be numeric. Ex. O1234.
The program name O0*** may not be used, as it is assigned to programs for system operation, automating functions, etc.
(Note: " *** " is used to denote 3 numeric characters.)

(e) A block which contains a program name must not contain other commands. For details on block format, refer to the Operations Manual for OSP5020M/OSP500M-G, SECTION 2, "PROGRAM MAKING".

(f) The program name for both main and sub program must start with "O".

(g) The schedule program is not assigned a name.

(h) Program names are read in units of characters. Examples:

1) O0123 is different from O123

2) O00 is different from O0.

(i) The same program name should not be used for two different programs.

7. Sequence Name

A sequence name is defined as a name assigned to a block. Numeric or alphabetic characters following after "N" are designated for a sequence name.

A sequence name makes it possible to use a sequence search function, a sequence stop function, and a branching function in a program.

If a sequence name contains alphabetic characters, it is referred to as a sequence label. If it is composed of numerics, it is referred to as a sequence name. This manual uses the term 'sequence name' to refer to both.

(1) Sequence Name Designation

(a) Input alphabets (A - Z) or Numbers (0 - 9) following after address "N".

(b) Up to five characters can be used.

(c) Both alphabetic and numeric characters may be used.

(d) A sequence name must be placed at the top of a block. However, an optional block skip command may be placed preceding a sequence name.

(e) Sequence numbers may be specified in any order.

(f) Sequence names are read in units of characters.

Examples:

1) N0123 is different from N123

2) N00 is different from N0.

8. Mathematical Operation Symbols

Mathematical operation symbols are used to convey logical operations, arithmetic operations, and trigonometric functions. A table of the operation symbols is shown below.

Operation symbols can be used together with variables to control peripherals or to pass on the results of the operation.

The logical operation functions and the trigonometric functions are optional.

Table 2-1 Mathematical Operation Symbols

Category	Operation	Operation Symbol	Remarks
Logical operation	Exclusive OR	EOR	0110 = 1010_EOR_1100
	Logical OR	OR	1110 = 1010_OR_1100
	Logical AND	AND	1000 = 1010_AND_1100
	Negation	NOT	1010 = NOT_0101
Arithmetic operation	Addition	+	8 = 5 + 3
	Subtraction	−	2 = 5 − 3
	Multiplication	*	15 = 5 * 3
	Division	/ (slash)	3 = 15 / 5
Trigonometric functions, etc.	Sine	SIN	0.5 = SIN [30]
	Cosine	COS	0.5 = COS [60]
	Tangent	TAN	1 = TAN [45]
	Arctangent (1)	ATAN	45 = ATAN [1] (value range −90° to 90°)
	Arctangent (2)	ATAN2	30 = ATAN 2 [1, $\sqrt{3}$] (See Note 1.)
	Square root	SQRT	4 = SQRT [16]
	Absolute value	ABS	3 = ABS [−3]
	Decimal to binary conversion	BIN	25 = BIN [$25] ($ represents hexadecimal number)
	Binary to decimal conversion	BCD	$25 = BCD [25]
	Integer implementation (rounding)	ROUND	128 = ROUND [1.2763 x 10^2]
	Integer implementation (truncation)	FIX	127 = FIX [1.2763 x 10^2]
	Integer implementation (raising)	FUP	128 = FUP [1.2763 x 10^2]
	Unit integer implementation (rounding)	DROUND	13.265 = DROUND [13.26462] (See Note 2.)
	Unit integer implementation (truncation)	DFIX	13.264 = DFIX [13.26462] (See Note 2.)
	Unit integer implementation (raising)	DFUP	13.265 = DFUP [13.26462] (See Note 2.)
	Remainder	MOD	2 = MOD [17, 5]
Brackets	Opening bracket	[	Determines the priority of an operation. (Operations inside the bracket are performed first.)
	Closing bracket	]	

Note 1: The value of ATAN2 (b, a) is an argument (range −180° to 180°) of the point whose rectangular coordinate is (a, b).

Note 2: In this example, the setting unit is mm.

Note 3: *Blanks must be placed before and after the logical operation symbols (EOR, OR, AND, NOT).*

Note 4: *Numbers after function operation symbols (SIN, COS, TAN, etc.) must be enclosed in brackets "[ ]".*
("a", "b", and "c" are used to indicate the contents of the corresponding bits.)

(a) Exclusive OR (EOR) c = a␣EOR␣b

If the two corresponding values agree, EOR outputs 0.
If the two values do not agree, EOR outputs 1.

a	b	c
0	0	0
0	1	1
1	0	1
1	1	0

(b) Logical OR (OR) c = a␣OR␣b

If both corresponding values are 0, OR outputs 0.
If not, OR outputs 1.

a	b	c
0	0	0
0	1	1
1	0	1
1	1	1

(c) Logical AND (AND) c = a␣AND␣b

If both corresponding values are 1, AND outputs 1.
If not, AND outputs 0.

a	b	c
0	0	0
0	1	0
1	0	0
1	1	1

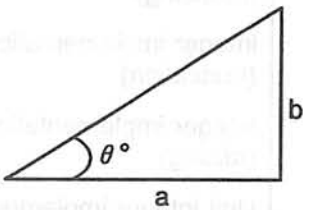
(d) Negation (NOT) b = NOT␣a

NOT inverts the value (from 0 to 1, and 1 to 0).

a	b
0	1
1	0

(e) Arc tangent (1) (ATAN)

θ = ATAN [b/a]



(f) Arc tangent (2) (ATAN2)

θ = ATAN2 [b, a]

(g) Rounding off to integer (ROUND)

ROUND rounds off a specified value into an integer.

(h) Truncation into integer (FIX)

FIX truncates (shortens) a specified value into an integer.

(i) Raising into integer (FUP)

FUP raises a specified value into an integer.

Note: *The units for ROUND, FIX, and FUP functions are microns.*

9. Block Skip Function

(1) Function

(a) Blocks preceded by "/n" are ignored during automatic mode operation if the BLOCK SKIP "n" switch, on the machine operation panel, has been pressed. If the switch is off (press again), the blocks are executed normally.
The operator thus has a method to execute or ignore blocks containing the "/n" code.

(b) When the block skip function is activated, the entire block will be ignored.

(2) Notes

(a) With this function, the number of block which can be skipped is one for standard.
Optionally, up to three blocks can be skipped and these blocks are distinguished from each other by placing the number after slash; "/1", "/2", and "/3". ("/n" is identified with "/1".)

(b) The slash "/n" must be placed at the start of the block. If it is placed in the middle of the block, an alarm is activated. A sequence name may precede the slash "/n".

(c) The slash "/n" may not be contained in the program name block.

(d) Blocks which contain a "/n" code are also subjected to TV and TH checks, regardless of the BLOCK SKIP switch position.

(e) Blocks which contain a "/n" code are also subjected to the sequence search function, regardless of the BLOCK SKIP switch position.

(f) The block skip function is not possible during SINGLE BLOCK mode. The succeeding block is executed, and then the operation stops.

10. Program Branch Function (Optional)

The program branch command specified in a part program can be executed by turning on the program branch switch on the machine operation panel.
Two program branch switches are provided: program branch switch 1 and program branch switch 2.
The following commands cause a branch.

(1) IF VPBR1 N*** A branch to the N*** block is executed when the program branch switch 1 is ON.

(2) IF VPBR2 N*** A branch to the N*** block is executed when the program branch switch 2 is ON.

Example:

```
          IF VPBR1 N100
          G00 X100 Z100
N100      G00  Y100
          IF VPBR2 N200
          G00 X200 Z200
N200      G00  Y200
          M02
```

Note 1: *In operation method B (large-capacity operation mode), use a sequence label name to specify the branch destination.*

Note 2: *The program branch function has the same restrictions as the branch function of User Task 1.*

11. Comment Function (Control Out/In)

A program may be made easier to understand by using comments in parentheses. The comment must be parenthesized to distinguish it from general operation information.

(1) All information placed in parentheses is regarded by the machine as comments.

(2) Comments are also subject to TV and TH checks and displayed in the normal character size.

Example:

N100 G00 X200 (FIRST STEP)
Comment

12. Message Function (Optional)

A message may be programmed to display during a conditional branching sequence.

(1) Format:

MSG (message statement)

(2) The display of the message statement on the CRT is twice the size of normal characters.

(3) If the MSG code is not followed by a message statement, the last comment statement from the previous block will be displayed.

(4) Up to 128 characters may be used in the message statement.

(5) The message function is possible only during machine operation mode.

(6) The following code can be used in the program to return the CRT screen to its original status after the message has been displayed:

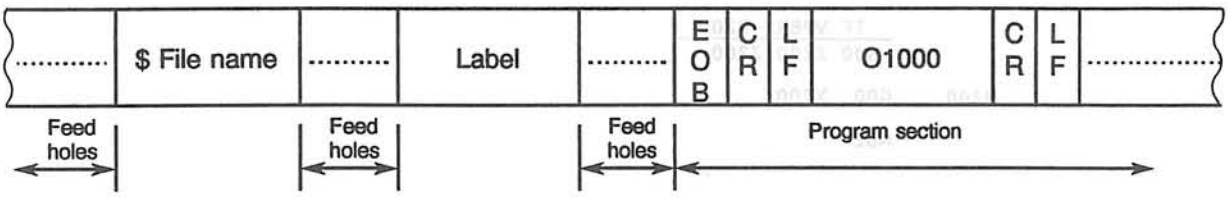
NMSG

13. Label Skip Function

Information, such as work name, parts code number, etc., may be added to the start of the program tape to assist in later identification by operators.
For this to be possible, a label skip function has been built into the machine. The label skip function allows the machine to ignore all information located at the start of the machining tape. This information is regarded as a label as is not read as programming commands.

(1) Label Location

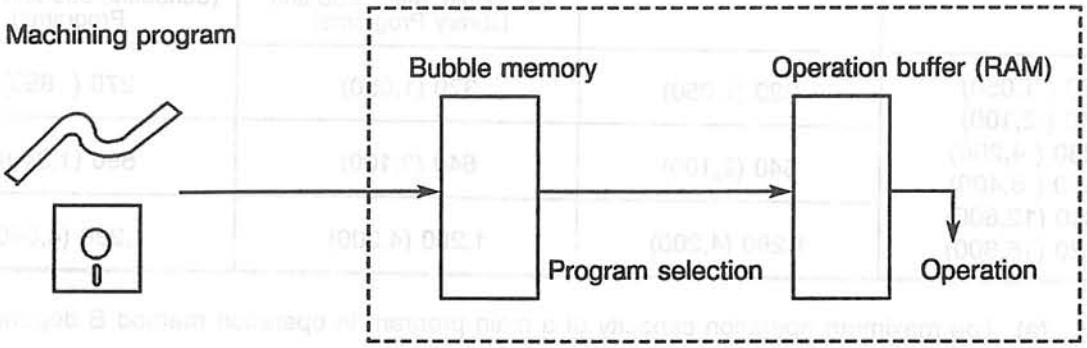
(a) The label should be located on the tape as shown:



(b) Information up to the EOB code on the program tape is regarded as a label and is ignored. If the tape starts with a "$" symbol, the accompanying file name is recognized and processed, and the information following the file name is taken as the label.

14. Operation Methods and Program Storage Memory Capacity

The OSP5020M/OSP500M-G uses bubble memory to store machining programs. The bubble memory capacity is selectable depending on the size of the user program. For execution, a program is transferred from the bubble memory to the operation buffer (RAM).
If the program size is larger than the operation buffer capacity, (for example, 160 meters), the program cannot be transferred from the bubble memory to the operation buffer in batch (at one time). A different operation method must be used in such cases.
The four operation methods possible with the OSP5020M are described in Section 14, 1. The programming restrictions which must be observed depending on the operation method selected are described in Section 14, 2.



(1) Operation Methods

The operation method is selected by the setting of the NC optional parameter (word) No. 11.

(a) When "0" is Set (Operation Method A)

The program to be executed is called to the operation buffer in batch. This method can be used when the program is smaller than the operation buffer capacity.

(b) When "1" is Set (Operation Method B)

The program to be executed is called to the operation buffer in several segments. This method is used when the program is larger than the operation buffer capacity.
Since schedule programs, subprograms, and library programs are generally called in batch, restrictions must be observed when making these programs.

(c) When "2" is Set (Operation Method S)

This operation method is used to execute a large volume program which does not use branch and subprogram call functions.

(d) When "3" is Set (Multi-volume Method M (optional))

This operation method is selected to execute a program which is stored on more than one floppy disk. When this method is selected, the program stored on different floppy disks is called to the operation buffer and executed sequentially.

(2) Programming Restrictions for the Operation Method

For various programming characteristics and their validity for three operation methods, refer to SECTION 12, 2-1. "PSELECT Block". These restrictions must be observed when making a program so that it conforms to the operation method format.

(3) Operation Buffer Expansion Specification (Optional)

The operation buffer capacity can be expanded to 320 m (1,050 ft.), 640 m (2,100 ft.), or 1,280 m (4,200 ft.) for a wider range of tape storage memory capacity.
The expanded memory capacity allows large volume programs to be handled by Operation Method A and Method B.

unit: m (ft.)

Tape Storage Capacity (Standard: 80 (263))	Operation Buffer Capacity (Standard: 160 (525))	Operation Method A (Normal capacity)	Operation Method B (Large capacity)
		Max. Operation Capacity (Schedule, Main, Sub and Library Programs)	Max. Operation Capacity (Schedule, Sub and Library Programs)
320 (1,050) 640 (2,100) 1,280 (4,200) 2,560 (8,400) 3,840 (12,600) 5,120 (16,800)	320 (1,050)	320 (1,050)	270 (890)
	640 (2,100)	640 (2,100)	590 (1,940)
	1,280 (4,200)	1,280 (4,200)	1,230 (4,040)

(a) The maximum operation capacity of a main program in operation method B depends on tape storage capacity.

(b) The library program capacity is equivalent to the designated library program buffer size. This means that the library buffer size is always contained in the operation capacity even if a library program is not registered.

(c) The number of subprograms and library programs possible is independent of the operation buffer size. They are always 63 and 65, respectively.

SECTION 3 COORDINATES AND COORDINATE COMMANDS

1. Control Axes

(1) The following table lists the addresses necessary to control the axes.

	Address	Contents
Basic axis	X, Y, Z	Addresses corresponding to the three axes orthogonal to one another
Parallel axis	U, V, W	Addresses of three orthogonal axes parallel to the basic axes
Rotary axis	A, B, C	Addresses of rotary axes in a plane orthogonal to the basic axes
Arc interpolation parameters	I, J, K	Addresses specifying distances, parallel to individual axes, from a start point to the center of an arc
	R	Addresses specifying the radius of an arc

(2) To move an axis, an axis address, the sign indicating the direction of the axis movement, and a numeric value which describes the axis movement must be designated. Refer to 6. "Absolute and Incremental Commands" for the numeric value.

(3) In this manual, to simplify the explanation for axis designation, "Xp", "Yp", and "Zp" are used instead of the actual axis addresses. They represent the axes as follows:

Xp ..... X-axis and the axis parallel to X-axis (U-axis)

Yp ..... Y-axis and the axis parallel to Y-axis (V-axis)

Zp ..... Z-axis and the axis parallel to Z-axis (W-axis)

(4) The OSP5020M/OSP500M-G can control up to six axes.

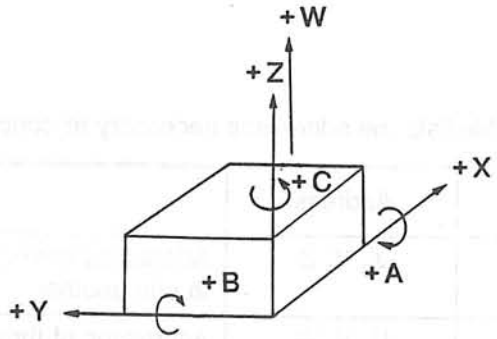
(5) The following table lists the number of simultaneously controllable axes for each of the axis movement modes.
Axis Movement Mode
Number of Simultaneously Controllable Axes

	"n" is the number of axes of the machine.
Positioning	n
Linear interpolation	n
Circular interpolation	2
Helical cutting	3
Manual feeding	1
Pulse handle feeding	1

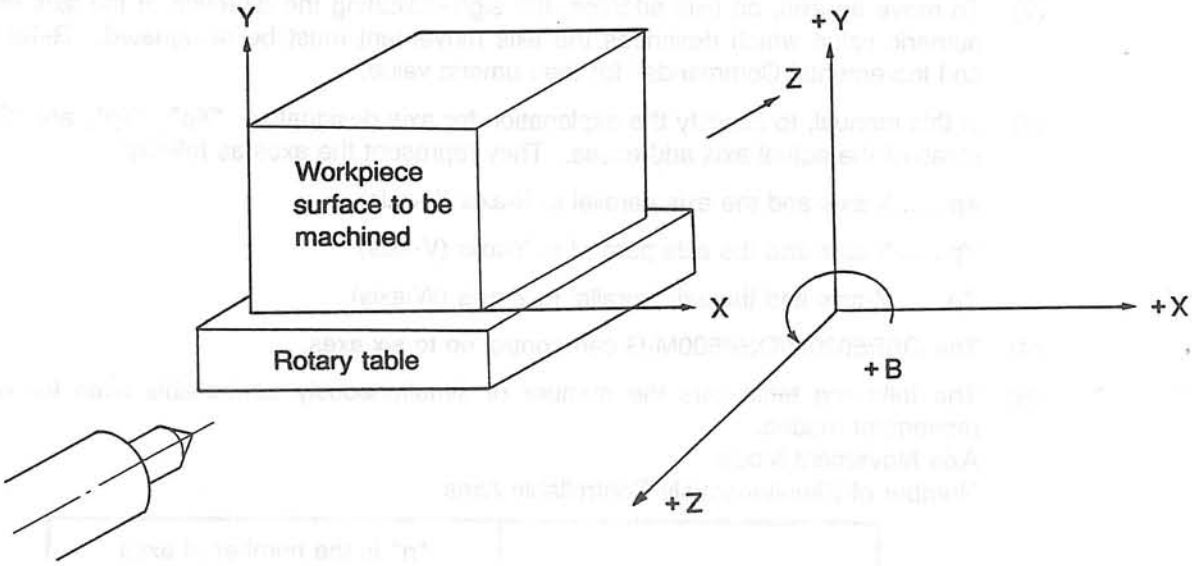
The pulse handle function is an option that can control 3 axes.

(6) The positive directions of the linear and rotary axes are defined as follows:

(a) Vertical



(b) Horizontal



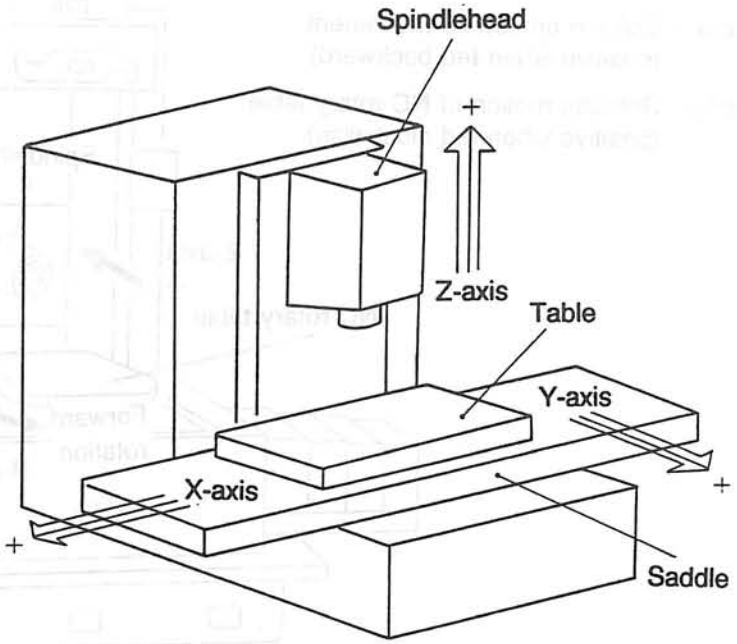
The definition of the coordinate axes and directions conforms to ISO (International Standards Organization) R841.

(c) Vertical Machining Center

X-axis : Table longitudinal movement (positive when fed to the left)

Y-axis : Saddle crosswise movement (positive when fed toward operator)

Z-axis : Spindlehead vertical movement (positive when fed upward)



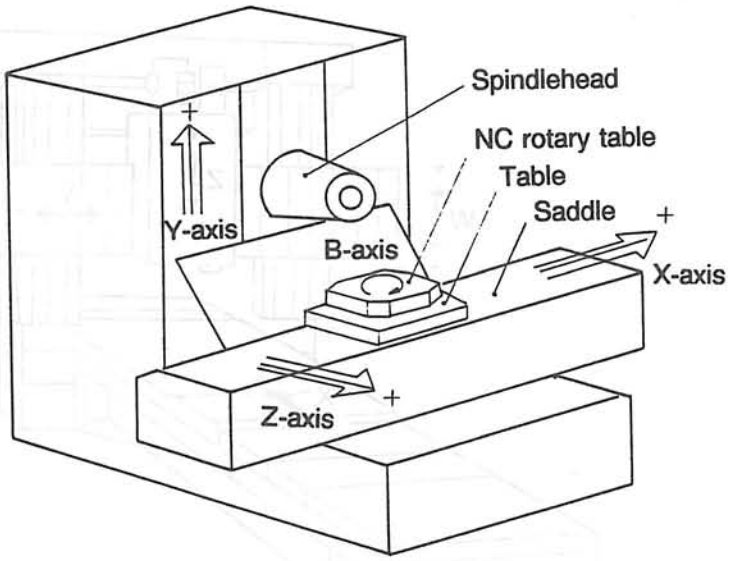
(d) Horizontal Machining Center

X-axis : Table longitudinal movement (positive when fed to the right)

Y-axis : Spindlehead vertical movement (positive when fed upward)

Z-axis : Saddle crosswise movement (positive when fed toward operator)

B-axis : Rotating motion of NC rotary table (positive when fed clockwise)



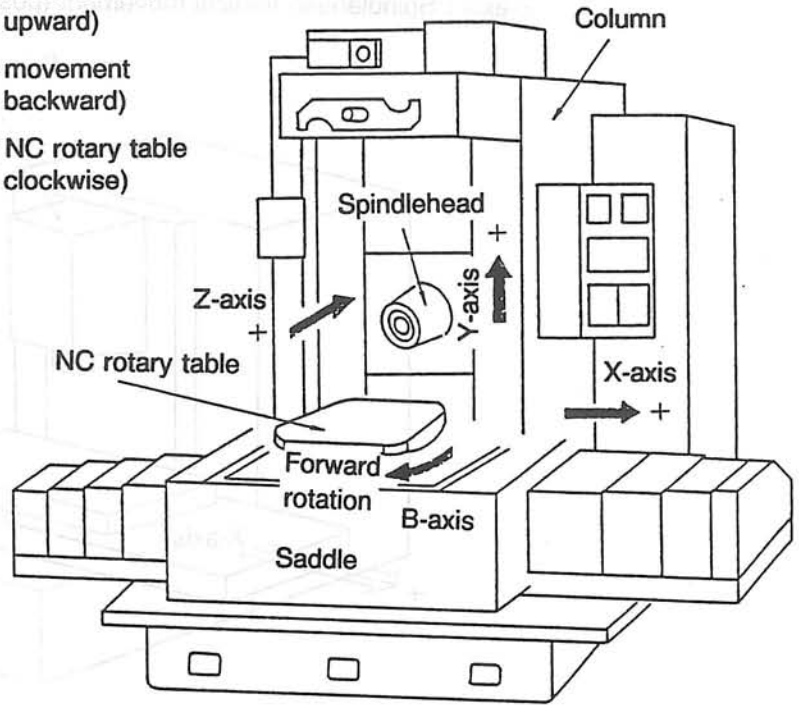
(e) Travelling Column Horizontal Machining Center

X-axis : Table longitudinal movement
(positive when fed to the right)

Y-axis : Spindlehead vertical movement
(positive when fed upward)

Z-axis : Column crosswise movement
(positive when fed backward)

B-axis : Rotating motion of NC rotary table
(positive when fed clockwise)



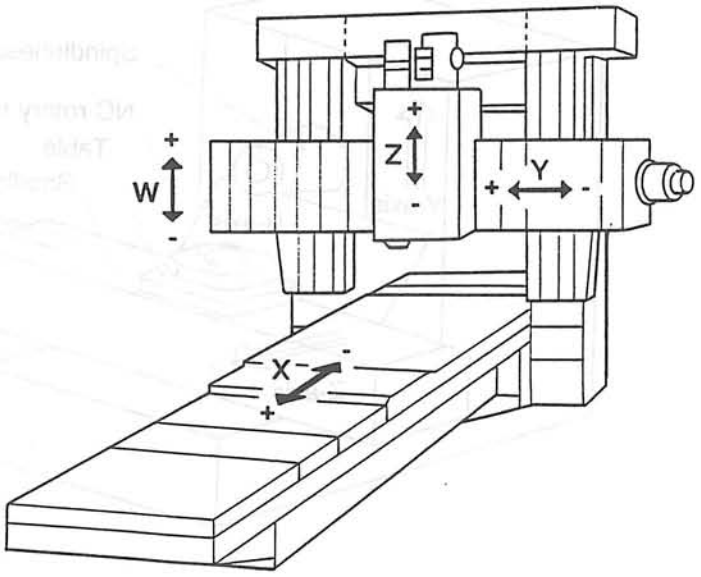
(f) Double Column Machining Center

X-axis : Table longitudinal movement (positive when fed in direction of advancing table)

Y-axis : Spindlehead crosswise movement (positive when fed to the left)

Z-axis : Quill vertical movement (positive when fed upward)

W-axis : Crossrail vertical movement (positive when fed upward)



2. Unit Systems

(1) Minimum Input Unit

The minimum input unit is the smallest unit which may be entered into the program.
For a linear axis, the minimum input unit is 0.001 mm or 0.0001 inch.
Selecting between metric and inch input is an optional function that can be done by setting bit 0 of the NC optional parameter (bit) No.3.
For a rotary axis, the minimum unit is 0.001 degree or 0.0001 degree regardless of which system (metric or inch) is selected. Either 0.001 degree or 0.0001 degree can be selected by changing the data at bit 0 of the NC optional parameter (bit) No. 3, bit 1, and No. 4, bit 0.

(2) Medium Input Unit

The input unit for the metric system can be changed to 0.01 mm be setting bit 1 and bit 2 of NC optional parameter (bit) No. 3 to "1". (All other units set remain unchanged.)

(3) Basic Input Unit

The input unit may be changed to the "basic" unit by setting bit 1 of NC optional parameter (bit) No. 3 to "1". The fundamental units are then 1 mm, 1 inch, 1 degree, and 1 second.

(4) Unit System for Data Setting

The unit system for program data entry and the unit system for data setting (such as zero position, tool data, and parameter data) are independent of each other.
The unit setting for data setting is set by setting bit 0, bit 1, and bit 6 of NC optional parameter (bit) No. 9.

3. Coordinate Systems and Values

(1) Description

In order to move the tool to a target position, a coordinate system must be established to specify the target position using coordinate values in the coordinate system.
Three different coordinate systems are used:

(a) Machine coordinate system. (Factory set.) The system may be changed from the factory setting, however the pitch error compensation data, travel limit values, etc. must also be changed if this is done.

(b) Work coordinate system. (Set by used.)

(c) Local coordinate system. (Temporarily set by NC program.)

The user can select the coordinate system to be used as needed.
The coordinate value is represented by components of the axes which make up the coordinate system.
A coordinate value is represented by a maximum of six axis components.

Example: "X__Y__Z__W__A__C__".

The number of programmable axes, that is, the number of axis components used to define a coordinate value varies depending on the machine specifications and models. This manual, therefore uses "P__" to designate a coordinate value.

(2) Machine Zero and Machine Coordinate System

The reference point on the machine is referred to as the machine zero and the coordinate system having the machine zero as the origin is referred to as the machine coordinate system.
The machine zero is set for each individual machine using system parameters. Since the travel end limits and the home positions are set on the machine coordinate system, the user should not change the location of the machine zero at their own discretion.

Note: that a cutting tool may not always be moved to the machine zero.

(3) Work Coordinate System

The coordinate system used to machine workpieces is referred to as the work coordinate system.

(a) Work coordinate systems are established and stored with work coordinate system numbers in the memory before starting operation. The desired work coordinate system is called from the machining stage.

(b) Work coordinate systems are set using offset values of the work zero from the machine zero.

(c) For details, see Section 5, 1-6. "Selection of Work Coordinate System" and Section 5, 1-7. "Change of Work Coordinate System".

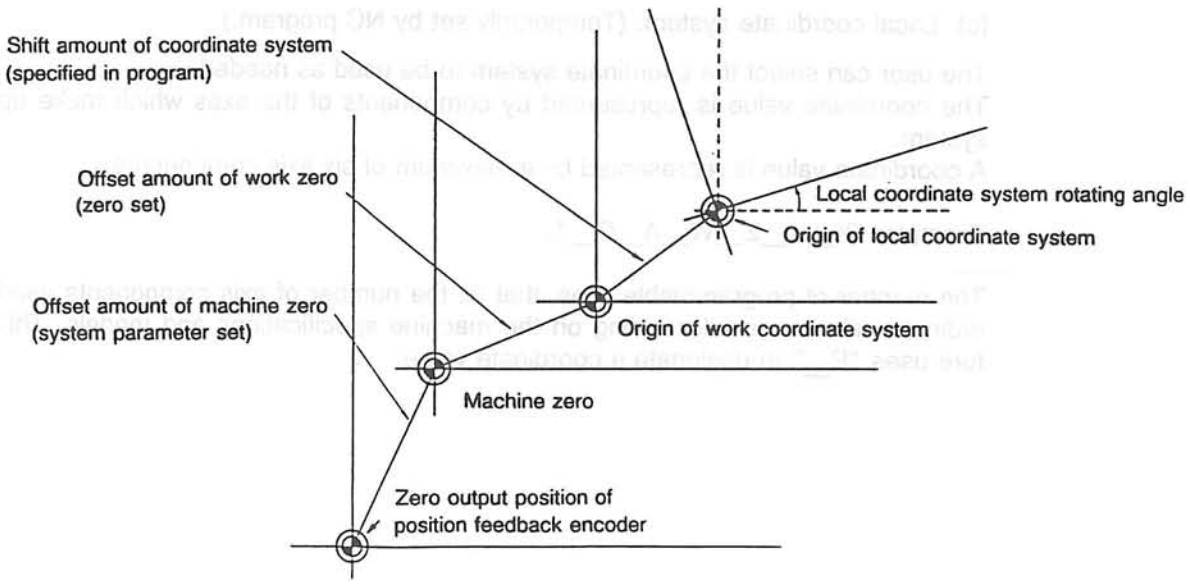
(4) Local Coordinate System

Programming the entire workpiece using only the work coordinate system may lead to difficulties on some portions of the workpiece. In such cases, a new coordinate system may permit easy programming.
The new coordinate system is referred to as a local coordinate system.

(a) The origin of the required local coordinate system is referenced to the origin of the presently selected work coordinate system. This coordinate value, along with the angle of rotation from the work coordinate system to the local coordinate system is programmed with the code G11 to establish the local coordinate system. Dimension words programmed after this setting has been made will all be executed on the newly-set local coordinate system.
To change the local coordinate system, the coordinate value and rotation angle of the new local coordinate system should be programmed with the code G11 just as the first local coordinate system was set.
The local coordinate system may be selected only by a program.

(b) To designate the coordinate value on the work coordinate system, cancel the local coordinate system with the code G10.

(c) For details, refer to Section 5, 3-1. "Parallel Shift and Rotation of Coordinates".



4. Travel Limit Commands

The OSP series is equipped with a type F OSP absolute position encoder, which is capable of setting the location of the travel limit, as an absolute value, using software instead of a hardware limit switch.

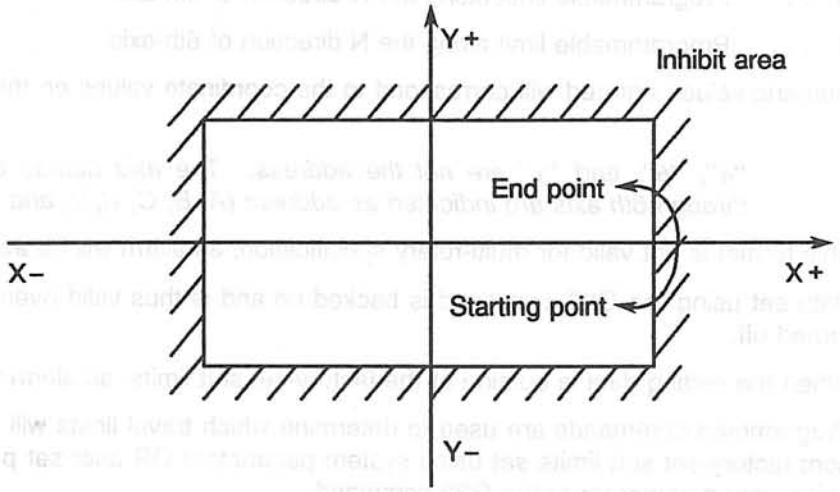
The travel limit location may be varied (user-set) by changing the set value, which is changed by a program. The user-set travel limit can therefore be referred to as a programmable limit.
The following two types of travel limit can be factory set or user set.

(1) Factory Set Travel Limit (Soft-Limit)

(a) The travel limit is set in accordance with the maximum travel distance from the machine zero of each axis. Setting the travel limits both in the positive (P) and negative (N) direction is carried out by the system parameters.

(b) The area inside of the set values (from the N direction travel limit to the P direction travel limit) is available for operation. The outside are is called the inhibit area and is not available for operation.

(c) The programmed path of an arc is prohibited from going into the inhibit area, even when the programmed end point lies inside the operation allow area.



(2) User-Set Travel Limit (Programmable Limit) (Optional)

The travel limit may be set by the user with user parameters and programs. Since both settings (user parameter and programmed command) establish the identical area and since the data is stored in the same area, the data entered last becomes the effective data, renewing the previously stored data. For example, the area-establishing data entered with user-parameter data is changed when area-setting data is generated by programmed commands. Both the data for the P and N directions must be given. The area between the P and N travel limits is available for operation, and the outside area, (including the set value), is the inhibit area.

(3) Setting Method

The setting method using programs is presented below.

Programming format:

【 G22 X__Y__Z__α__β__γ__I__J__K__P__Q__R__ 】

X . . . Programmable limit along the P direction of X-axis

Y . . . Programmable limit along the P direction of Y-axis

Z . . . Programmable limit along the P direction of Z-axis

α . . . Programmable limit along the P direction of 4th-axis

β . . . Programmable limit along the P direction of 5th-axis

γ . . . Programmable limit along the P direction of 6th-axis

I . . . . Programmable limit along the N direction of X-axis

J . . . Programmable limit along the N direction of Y-axis

K . . . Programmable limit along the N direction of Z-axis

P . . . Programmable limit along the N direction of 4th-axis

Q . . . Programmable limit along the N direction of 5th-axis

R . . . Programmable limit along the N direction of 6th-axis

The numeric values entered will correspond to the coordinate values on the work coordinate system.

Note: "α", "β", and "γ" are not the address. The axis names corresponding the 4th through 6th axis are indicated as address (A, B, C, U, V, and W).

(a) This format is not valid for multi-rotary specification; an alarm will be activated.

(b) Data set using the G22 command is backed up and is thus valid even after power has been turned off.

(c) When the setting data is outside of the factory-set soft limits, an alarm will be activated.

(d) Programmed commands are used to determine which travel limits will be available for operation: factory-set soft limits set using system parameters OR user-set programmable limits set using user parameters or the G22 command.

G22 . . . The user-set travel limits are used for checking data.

G23 . . . The G22 command is canceled and the factory-set travel limits are used for checking data.
If only the G22 command is specified, the user-set limits are used.

(e) For setting the soft limits with the user parameter, refer to the OSP5020M/OSP500M-G Operation Manual Section 3, 3-2. "User Parameters."

(f) The programmed path is checked for the inhibit area even when the programmed end point lies inside the operation allow area.

(g) Both G22 and G23 are options.

5. Home Position Command (G30)

The home position refers to a particular position preset on every machine. It is used as the tool and pallet changing position. The home position is defined using a coordinate value on the machine coordinate system, and can be set using a system parameter.
The home position command permits the axes to be immediately fed to that position.

Programming format: G30 P__

P : Home position number. Up to 8 home positions may be set.
On some models, up to 32 numbers may be set.
The operating sequence and the home position of each axis is specified by the manufacturer.
For details, refer to the OSP5020M/OSP500M-G Operation Manual, Section 3, 3-13. "Parameter Setting".

6. Absolute and Incremental Commands (G90, G91)

The absolute and incremental dimensioning commands are used to specify the mode of axis movement.

(1) Absolute Command, G90

The G90 command specifies the absolute dimensioning mode. In this mode, it is necessary to program the coordinate values, on the selected work coordinate system, of the point.

(2) Incremental Command, G91

The G91 command specifies the incremental dimensioning mode. In this mode, it is necessary to program the axis movement amount from the current position to the target position.

(3) Notes

(a) Commands G90 and G91 cannot be used simultaneously in one block.

(b) After power has been turned on, either G90 or G91 may be effective.

(c) The priority between the two commands may be specified using parameters (bit 4 of NC optional parameter (bit) No. 18). Setting the priority will determine which mode will be effective when power is turned on.

(d) When an incremental command needs to be designated right after the completion of a fixed cycle, axis commands in the fixed cycle must be designated using absolute values.

```
G81  X__Y__Z__R__F__
     X__Y__
        ⋮
     X__Y__
G80
G00  Z__
G91  X__Y__Z__
        ⋮
```

SECTION 4 MATH FUNCTIONS AND AXIS MOVEMENT COMMANDS

1. Positioning (G00)

Each axis moves independently from its current position to the target position at its own rapid feedrate.

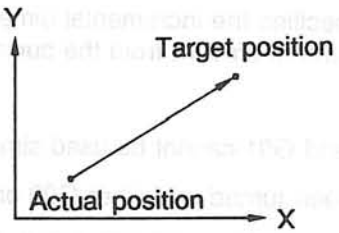
Programming format : G00 IP_

The G00 positioning command executes a position check and then moves the control on to the next block. The position check verifies that the machine stop position is somewhere in the required range by comparing the specified coordinate values against the actual position.

(1) The interpolation mode for positioning can be selected by setting bit 0 of the NC optional parameter (bit) No. 46.

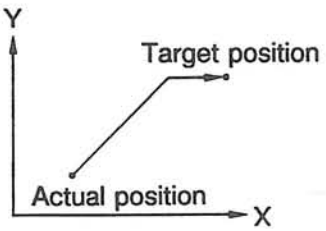
(a) Linear Interpolation Mode

The tool path is generated along a straight line from the actual position to the target position. Axis feedrates are selected close to the rapid feedrate of each axis so that positioning time can be minimized.



(b) Non-linear Interpolation Mode

The axes move independently of each other at a rapid feedrate. Therefore, the resultant tool path is not always a straight line.



Note 1: *The rapid feedrate of each axis can be set with a system parameter. It is not possible to set the feedrate using an F command.*

Note 2: *The "in-position" range can be set for each axis using a system parameter.*

2. Linear Interpolation (G01)

The G01 command specifies the axes to move directly from the current position to the specified co-ordinate values at the specified feedrate.

Programming format : G01 IP_ F_

IP : Target point (end point)
F : Feedrate.The command remains effective until updated by another value.

(1) Notes

(a) The feedrate becomes zero when the NC is reset, unless it has been saved with the F1 digit specification.

(b) The feedrate for each axis is indicated below. (Calculate the feedrate for X, Y, Z-axis in teh incremental value)

G01 XxYyZzFf

Calculation of feedrates:

X-axis feedrate: FX = (x/L)f
Y-axis feedrate: FY = (y/L)f
Z-axis feedrate: FZ = (z/L)f

where $L = \sqrt{x^2 + y^2 + z^2}$

Rotary axis feedrates are handled as follows:

1 mm/min = 1 deg/min
1 mm/min = 1 deg/min

The above formulas are also effective for linear interpolation in which a command for the rotary axis is designated.

Example:

G91 G01 X10 C20 F30.0

Metric system

X-axis feedrate $FX = 10/\sqrt{10^2 + 20^2} \times 30 \doteqdot 13.41$ mm/min
C-axis feedrate $FC = 20/\sqrt{10^2 + 20^2} \times 30 \doteqdot 26.83$ deg/min

Inch system

X-axis feedrate $FX = 10/\sqrt{10^2 + 20^2} \times 30 \doteqdot 13.41$ inch/min
C-axis feedrate $FC = 20/\sqrt{10^2 + 20^2} \times 30 \doteqdot 26.83$ deg/min

(c) In the inch system, it is possible to specify how the F (F1) value will be interpreted, as 1 deg/min or as 25.4 deg/min, by setting bit 7 of the NC optional parameter (bit) No. 15.

3. Circular Interpolation (G02, G03)

Circular interpolation can be used to generate a cutting path which follows an arc.
Programming Format

Arc on Xp-Yp plane

$$[\ G17\ \left\{\begin{matrix}G02\\G03\end{matrix}\right\}X_p\_Y_p\_\left\{\begin{matrix}R\_\\I\_J\_\end{matrix}\right\}F\_\ ]$$

Arc on Zp-Xp plane

$$[\ G18\ \left\{\begin{matrix}G02\\G03\end{matrix}\right\}Z_p\_X_p\_\left\{\begin{matrix}R\_\\K\_I\_\end{matrix}\right\}F\_\ ]$$

Arc on Yp-Zp plane

$$[\ G19\ \left\{\begin{matrix}G02\\G03\end{matrix}\right\}Y_p\_Z_p\_\left\{\begin{matrix}R\_\\J\_K\_\end{matrix}\right\}F\_\ ]$$

Xp = X-axis or U-axis

Yp = Y-axis or V-axis

Zp = Z-axis or W-axis

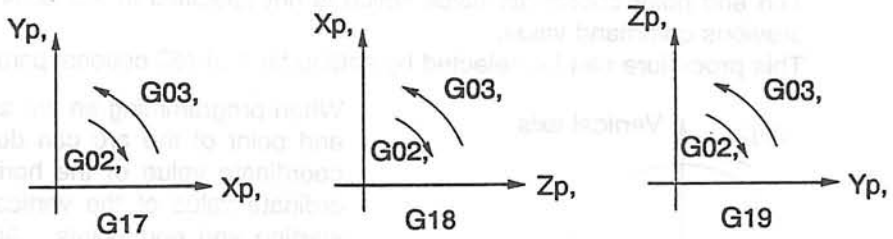
(1) G codes and addresses

The following table presents a summary of the G codes and addresses necessary for circular interpolation.

Item to be Designated		Command	Description
Plane selection		G17	Circular arc in the Xp-Yp plane
		G18	Circular arc in the Zp-Xp plane
		G19	Circular arc in the Yp-Zp plane
Rotary direction		G02	Clockwise
		G03	Counterclockwise
End point	G90	Specify 2 points [Xp, Yp], [Xp, Zp], or [Yp, Zp]	End point in the work coordinate system
	G91	Specify 2 points [Xp, Yp], [Xp, Zp], or [Yp, Zp]	End point referenced to the starting point. Values should include signs.
Center of arc referenced to starting point		Specify 2 values [I, J], [I, K], or [J, K]	Center of arc referenced to starting point including signs.
Arc radius		R	Radius of arc

(2) Rotary Directions

The two rotary directions, clockwise and counterclockwise, are defined when viewing the plane from the positive direction of the third axis on Xp-Yp (Zp-Xp, Yp-Zp) plane, as shown in the diagrams below.

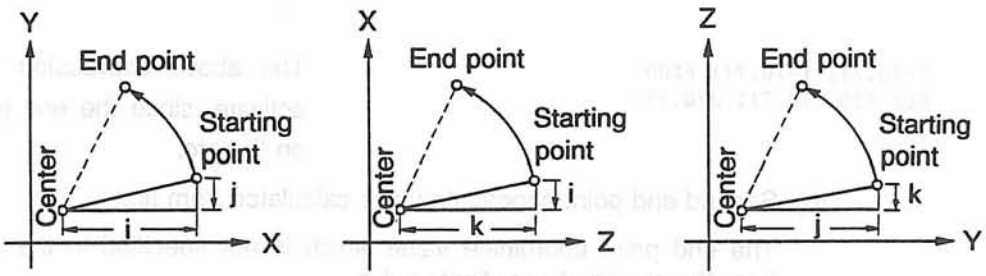


(3) End Points

The end point of an arc is reached depending on the G90/G91 selection.

(4) Center Points

The center point of an arc is determined by the I, J, and K values which correspond to Xp, Yp, and Zp, respectively. Their coordinate values are always incremental, regardless of G90 or G91.



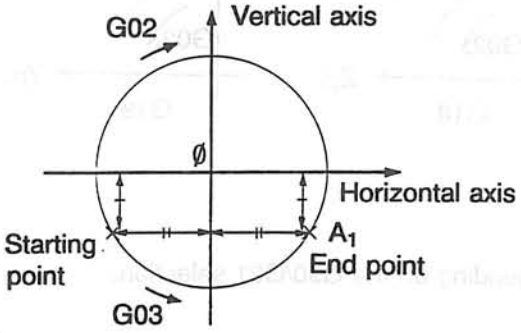
A minus should be used for the I, J, and K values when necessary.

(5) Arc end point designation,

The designation of the end point of an arc is possible by specifying the coordinate value on one of the two axes.
There are two arc-defining procedures:

(a) Previous command value used as end point coordinate value

The end point coordinate value which is not specified in the command can be taken as the previous command value.
This procedure can be selected by setting bit 1 of NC optional parameter (bit) No. 20 to "0".



When programming an arc as illustrated at the left, the end point of the arc can de designated with only the coordinate value of the horizontal axis, since the coordinate value of the vertical axis is the same at the starting and end points. An alarm occurs if the end point does not lie on an arc.

Example:

```
X-70.711 Y-70.711 F200
G02 X70.711 I70.711 J70.711
```

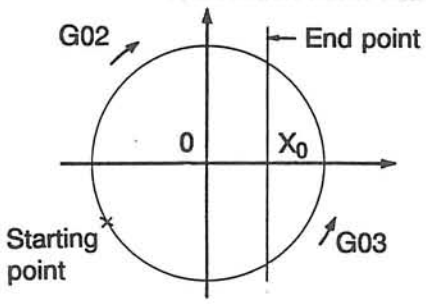
The above expression defines a clockwise arc whose radius is 100, center at (0, 0), starting point at (−70.711, −70.711) and end point at (70.711, −70.711).

```
X-70.711 Y-70.711 F200
G02 X10 I70.711 J70.711
```

The above expression will cause an alarm to activate, since the end point (10, −70.711) is not on the arc.

(b) Second end point coordinate value calculated from first

The end point coordinate value which is not specified in the command can be calculated from the specified coordinate value.
This procedure can be selected by setting bit 1 of NC optional parameter (bit) No. 20 to "1".



When programming an arc as illustrated at the left, the end point can be designated with only the horizontal axis coordinate value. The vertical axis coordinate value is calculated from the horizontal axis coordinate value.

If more than one end point is possible, the one which is reached first in the designated arc direction is selected.

Example:

```
X-70.711 Y-70.711 F200
G02 X10 I70.711 J70.711
```

The above expression defines a clockwise arc with radius 100, center at (0, 0), starting point at (−70.711, −70.711) and end point at (10, 99.499).

The operations explained in this section are possible for both the horizontal and the vertical axes.

(6) Types of Arcs

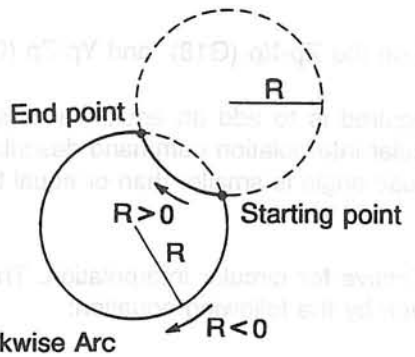
The center of an arc can be defined either by using values for I, J, and K, or by specifying the radius (R) of the arc. When the arc is specified by the radius, there are 4 possible different arcs.

(a) Clockwise arc (G02)

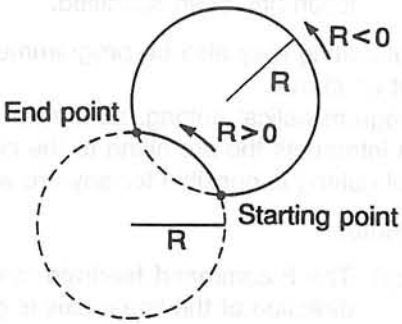
An arc whose central angle is smaller than or equal to 180 degrees : Radius R > 0
An arc whose central angle is greater than 180 degrees : Radius R < 0

(b) Counterclockwise arc (G03)

An arc whose central angle is smaller than or equal to 180 degrees : Radius R > 0
An arc whose central angle is greater than 180 degrees : Radius R < 0



i) Clockwise Arc



ii) Counterclockwise Arc

(7) Feedrates

The feedrate during circular interpolation is the feedrate component tangential to the arc.

(8) Notes

(a) To omit the values of I, J, or K, enter a zero "0".

(b) An arc with radius 0 cannot be specified.

(c) If the values for Xp, Yp, and Zp are omitted, there are two possible arcs:

① A 360-degree arc, with the center specified by I, J, and K values;

② A zero-degree arc, with the radius specified by the R value.

(d) It is not possible to specify values at R and I, J, K at the same time.

(e) It is not possible to specify any axis parallel to the axes which make up the selected plane. Designation of the W-axis is not allowed while the Z-X plane is selected.

(f) An alarm will be activated if the difference in radius between the starting point and the end point of an arc is greater than or equal to the value set on the NC optional parameter (long word) No. 3.

4. Helical Cutting (G02, G03) (Optional)

Helical cutting or helical interpolation may be performed by synchronizing circular interpolation with linear interpolation on an axis which in intersected at right angles with an arc plane.
Programming format:

XpYp plane

$$[\ G17 \left\{ \begin{matrix} G02 \\ G03 \end{matrix} \right\} X_p\_ Y_p\_ \left\{ \begin{matrix} R\_ \\ I\_ J\_ \end{matrix} \right\} F\_\ ]$$

α : the angle between the direction of helical movement and the plane on which circular interpolation has been specified.

Helical cutting may also be programmed on the Zp-Xp (G18) and Yp-Zp (G19) planes, using a similar format as above.
To program helical cutting, all that is required is to add an additional axis command (α) for the axis which intersects the arc plane to the circular interpolation command described in Section 3-3.
Helical cutting is possible for any arc whose angle is smaller than or equal to 360 degrees.

(1) Notes

(a) The F-command feedrate is effective for circular interpolation. Therefore, the feedrate in the direction of the linear axis is given by the following equation:

$$\frac{\text{Linear axis travel}}{\text{Arc length}} \times F$$

(b) Tool length offset is effective for the arc's right angle axis command.

(c) Cutter radius offset is effective only for circular interpolation commands.

5. Feed Functions (G94, G95)

(1) Rapid Feedrate

During the rapid feedrate mode, each of the axes is fed at the specified rapid feedrate independently of other simultaneously-fed axes. Consequently, the various axes arrive at the target point at different times.
The rapid feedrate may be modified by setting the override selector switch on the operation panel.

(2) Cutting Feed

(a) Feed per minute (G94)

1) How fast a tool should be fed is specified by the address "F" and the digits after it.

2) Depending on the setting of the NC optional parameter (bit) No. 3 setting, the unit system may be one of five choices: 1 mm/min, 0.1 mm/min, 1 inch/min, 0.1 inch/min or 0.01 inch/min.

3) The command value may range from 0.1 to 24000.0 mm/min or from 0.01 to 2400.00 inch/min.

4) When an axis has been programmed to be fed at a rate exceeding the maximum value, the feedrate "clamps" at a preset maximum value and an alarm message is displayed on the CRT:

W05 WARNING ! Over feedrate command limit (Replacing)

5) The clamp speed, which replaces the excessive programmed feedrate, is set by NC optional parameter (long word) No. 10, with a range from 0 to 20000. The initial setting is usually the maximum rapid feedrate of the machine.

6) The feedrate may be overridden in 24 steps which usually range between 0 and 200 percent, depending on the machine. The clamp speed function then applies to the actual speed, after the override has been performed.

(b) Feed per revolution (G95)

1) The G95 command is used to specify the speed of revolution of the spindle.

2) Depending on the setting of NC optional parameter (bit) No. 3, the unit system may be one of six choices: 1 mm/rev, 0.01 mm/rev, 0.001 mm/rev, 1 inch/rev, 0.001 inch/rev or 0.0001 inch/rev.

3) The command value mat range from 0.001 to 500.000 mm/rev or from 0.0001 to 50.000 inch/rev.

4) The clamp speed value is converted using the following expression:

$$fm = fr \times N$$

where,
N = spindle speed
fm = feedrate (mm/min)
fr = feedrate (mm/rev)

(c) F1 feed function (optional)

1) The F1 feed function is activated by feedrate switches or by using the parameters (F1 through F9) for controlling the feedrate and thus differs from the standard F command (F4). The F1 function activates four (or eight) 256-step rotary feedrate switches which are specified with a F command, F1 through F4 (or F1 through F8).

2) The F1 function can be distinguished from the standard feedrate command (F4) in the following ways:

a) If the numerical value following the address character F is an integer from 1 to 8, it is interpreted as an F1 function.

b) If a real number follows the F address, (for example, F1.), or the F address consists of a variable in its data section, it is interpreted as the standard feedrate command. The F1 function is not permitted to consist of variables.

Examples:

F1 . . . . . The feedrate is determined by the setting on rotary switch F1.
F200 . . . . Feedrate = 200 mm/min
F5. . . . . Feedrate = 5 mm/min

LA1 = 8
F = LA1
Feedrate = 8mm/min

3) A lamp on the operation panel indicates that the F1 function has been chosen.
The F1 function is not cleared when the control is reset, but it is cleared when a standard feedrate F command (F4) is given and when power is turned off.

4) Programming the "F0" command calls the axis rapid traverse function.

5) The setting of a F1 feed function rotary switch may be changed during the execution of programmed axis movements.

6) A feedrate override setting is ineffective while an F1 command is programmed.

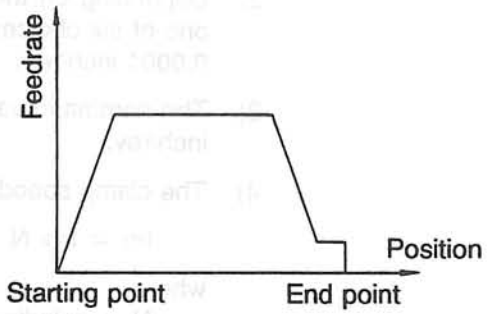
7) If an F1 command is given during the G95 mode (feed per revolution), an alarm occurs

6. Automatic Acceleration and Deceleration

Acceleration at the start of an axis movement and deceleration at the end of an axis movement is automatically incorporated into axis movements. The amount of acceleration is determined by a time constant which can be set with a system parameter or by programming a system variable.

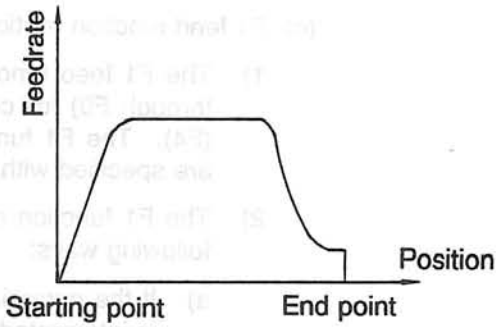
(1) Positioning Mode and Manual Feed Mode

During positioning mode and manual feed mode, automatic acceleration and deceleration is linear, as shown at right.



(2) Cutting Mode (G01, G02, G03)

During cutting mode, automatic acceleration and deceleration is competent, as shown at the right.



(3) Processing Between Blocks - Position Check

Type of Old Block / Type of New Block	Positioning	Cutting Feed	No Axis Movement
Positioning	○	○	○
Cutting Feed	○	×	○
No axis movement	○	○	○

○: Position check performed
×: No position check performed

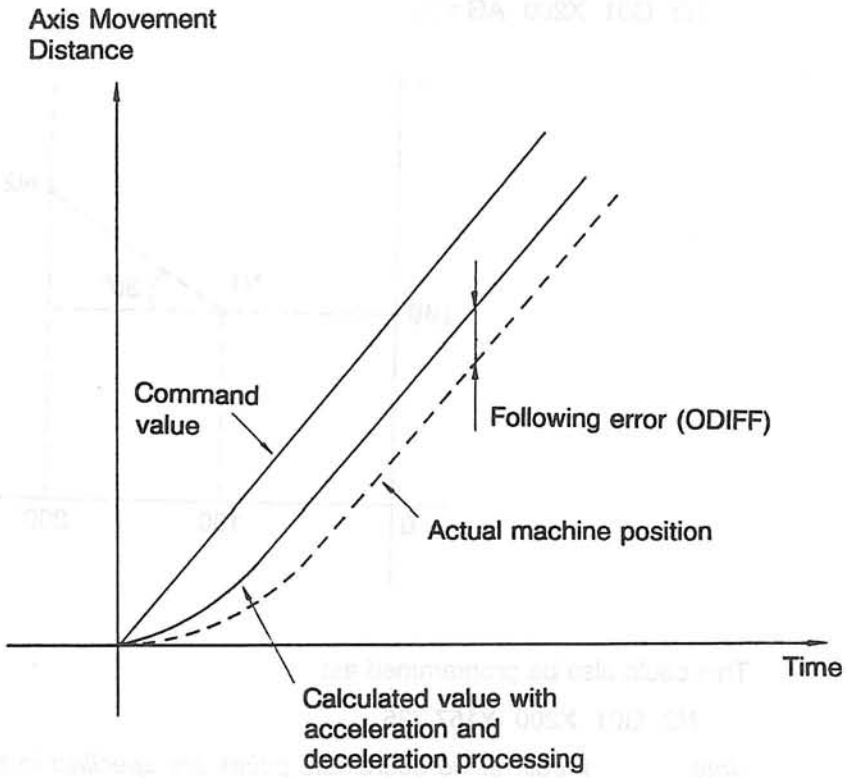
Usually, when control passes from one block to the next, a position check is performed to verify that the machine position is in the required range by comparing the specified coordinate values to the actual position. The "in-position" range can be set using a system parameter
The table above can be used to determine whether a position check will be performed when passing from one type of block to another type.

(4) Exact Stop (G61 and G09)

As shown in the table above, no position check is performed when changing from one cutting feed block to another cutting feed block. This may cause joints between blocks to be cut dull or rounded. This problem may be solved by using the exact stop check, G61, G09. For details, refer to Section 5, 1-3. "Exact Stop Check".

7. Following Error Check

Following error is defined as the difference between the command value from the NC and the detected position value.
A servo alarm will be activated if a following error (ODIFF) reaches a certain value during rapid or cutting feed of an axis.



8. Dwell Command (G04)

The dwell command can be programmed to be performed at the end of a block. During dwell, the control performs the following actions: checks the position, stands still for a specified period of time, and moves to the next block.
There are two possible formats for the dwell command.

(1) 【 G04 F___ 】

F: Dwell time
The unit system for the dwell time, specified with NC optional parameter (bit) No. 3, may be selected from four possibilities: 1, 0.1, 0.01 and 0.001 seconds.
The maximum possible dwell time is 99999.999 seconds.

(2) 【 G04 P___ 】

P: Dwell time
The unit system for the dwell time is same as for F above.

9. Angle Command Function

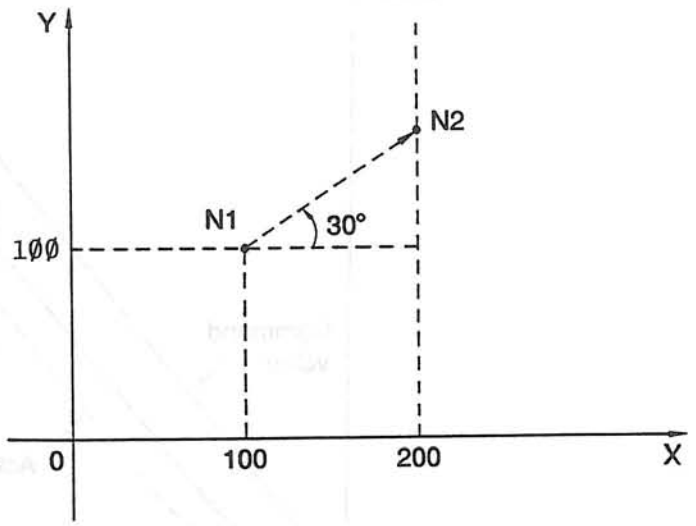
The angle command function can be used to move the axes to a desired point defined by one coordinate value and an angle. The axis coordinate value must be on the selected plane and the angle is measured from the horizontal axis to a line drawn from the present position to the desired position.

(1) The angle command is specified using "AG=" as an address character, followed by a numeric value.

(2) The angle may be specified in degrees or in some other unit by changing the NC optional parameter (bit) No.3 or No.4, bit 0. Negative values are permitted.

Example:
On the XY plane

```
N1  G00  X100  Y100
N2  G01  X200  AG=30
```



This could also be programmed as:

```
N2  G01  X200  Y157.735
```

Note: *If both or no coordinate points are specified in a line containing the AG command, an alarm is activated.*

10. Manual Shift Amount Cancel Command

This function can be used to cancel the total manual shift amount from pulse handle overlap or manual intervention operation. The function is performed by incorporating a command in the program; no manual axis operation is required.
The manual shift amount cancel function is not associated with actual axis motion, but only with changing the actual position data.

(1) Programming Format

MITCAN
There are two possible command formats. The block skip command may also be incorporated.

(a) Without a sequence number (label):

MITCAN

(b) With a sequence number (label):

N*** MITCAN

(2) Operations

(a) When the manual shift amount cancel command (MITCAN command) has been designated, the manual shift amount is canceled and actual position data is taken as the calculated value. It is not associated with axis motion.

Example 1:

1) Before designating the manual shift amount cancel command

100.0 + 500.0 = 600.0

(manual shift amount) (calculated value) (actual position data)

2) After designating the manual shift amount cancel command

0.0 + 600.0 = 600.0

(manual shift amount) (calculated value) (actual position data)

(b) The incremental command which is designated right after the manual shift amount has been canceled initiates incremental motion from the previously commanded value.

Example 2:
Assume that manual shift amounts in the X-, Y-, and Z-axis directions are 50, 50, and 0, respectively.

```
N100 G90 G0 X400 Y300 Z0
N101 MITCAN
N102 G91 X20
N103     Y10
```

1) Axis motion

```
N100/N101    (X, Y, Z) = (450, 350, 0)
N102         (X, Y, Z) = (420, 350, 0)
N103         (X, Y, Z) = (420, 310, 0)
```

2) Details of axis motion

N100 Axes are positioned at "X=450, Y=350, Z0" in the work coordinate system. Commanded values and calculated values are both "X=400, Y=300, Z0".

N101 "X=450, Y=350, Z0" are taken as calculated values.

N102 Axes are positioned at "X=420, Y=350, Z0" in the work coordinate system.
Commanded values are "X=420, Y=300, Z0".
Calculated values are "X=420, Y=350, Z0".

N103 Axes are positioned at "X=420, Y=310, Z0" in the work coordinate system.
Commanded values are "X=420, Y=310, Z0".
Calculated values are "X=420, Y=310, Z0".
Axes are positioned at a position which is obtained by adding the manual shift amount to the calculated value. (axes move from the position which has the previously calculated value to the position which is obtained by adding the manual shift amount to the commanded value.)

Example 3:

Assume that manual shift amounts in the X-, Y-, and Z-axis directions are 50, 50, and 0, respectively. And assume that the X-, Y-, and Z-coordinates of the tool change position are 700, 0, and 0, respectively.

```
N100 G90 G0 X400 Y300 Z0
N101M06
N102MITCAN
N103 G91 X20 Y10
```

1) Axis motion

```
N100          (X, Y, Z) = (450, 350, 0)
N101/N102     (X, Y, Z) = (700, 0, 0)
N103          (X, Y, Z) = (420, 310, 0)
```

2) Details of axis motion

N100 Axes are positioned at "X=450, Y=350, Z0" in the work coordinate system. Commanded values and calculated values are both "X=400, Y=300, Z0".

N101 Axes are positioned at the tool change position whose coordinate values are "X=700, Y=0, Z0".

Commanded values are "X=400, Y=300, Z0".

Calculated values are "X=650 Y=-50, Z0".

N102 The calculated value is "X=700 Y= 0, Z0".

N103 Axes are positioned at "X=420, Y=310, Z0" in the work coordinate system.

Commanded values are "X=420, Y=310, Z0".

Calculated values are "X=420, Y=310, Z0".

(3) Notes

(a) An alarm will be activated if the MITCAN command is executed during cutter radius compensation or 3-D offset mode.

(b) The MITCAN command may only be carried out before, and not during, a sequence restart operation.
The manual shift amount will not be canceled if the MITCAN command is given during the sequence restart operation or if the MITCAN command is designated in a block which contains the sequence restart command.

(c) No other commands may be designated in a block with the MITCAN command, except the block skip command.
Only a sequence number (label) and a block skip command may precede the MITCAN code.

Example: N 100 MITCAN X100 Y0
These commands are skipped.

SECTION 5 PREPARATORY FUNCTIONS

G codes are used to specify particular functions which are to be executed in individual blocks. Every G code consists of the address "G" plus a 2-digit number (00-99), although some G codes use mnemonics* (alphabetic commands).

*Mnemonics: A combination of characters A through Z, comprising up to eight characters.

(1) Effective G code ranges

One-shot : A one-shot G code is effective only in a specified block and is automatically canceled when it moves to the next block.

Modal : A modal G code is effective until it is changed to another G code in the same group.

(2) Special G codes

The mnemonic codes of subprogram calls (G101 through G110, for instance) and branch instructions are called special G codes. Every special G code must be specified at the beginning of a block, not halfway during a block. Note, however, that a "/" (optional block skip) and a sequence name may be placed before a special G code.

Note: *Refer to Section 14, 2 for the G codes list.*

1. Preparatatory Functions

1-1. Plane Selection (G17, G18, G19)

(1) Selecting a plane is necessary in order to perform the following functions:

Circular interpolation (Helical cutting)
Angle command (AG)
Cutter radius compensation
Coordinate rotation (Local coordinate system)
Fixed cycle
Coordinate calculation
Area machining

(2) Any one of the following planes may be selected

G17: Xp-Yp plane
G18: Zp-Xp plane
G19: Yp-Zp plane

Xp is the X- or U-axis;
Yp is the Y- or V-axis; and
Zp is the Z- or W-axis.

(3) Programming Format

【 G17 Xp__ Yp__ 】
【 G18 Zp__ Xp__ 】
【 G19 Yp__ Zp__ 】

(4) Either the basic axis system (X,Y,Z) or the parallel axis system (U,V,W) is selected by specifying the particular axis addresses in the block containing G17, G18 or G19.

Examples:

```
G17  X__Y__  XY plane
G17  U__Y__  UY plane
G18  Z__X__  ZX plane
G18  W__X__  WX plane
G19  Y__Z__  YZ plane
G19  Y__W__  YW plane
```

(5) A plane selection command block must include one of the codes G17, G18, or G19. Only specifying the axis address will not select a plane.

(6) The default of a missing axis address is the basic axis address.

Examples:

```
G17       XY plane
G17 X__   XY plane
G17 U__   UY plane
G18       ZX plane
G18 W__   WX plane
```

(7) If an axis movement command is specified on a plane which has not been selected, the programmed command will nonetheless be executed as programmed, and the selected plane will be temporarily ignored.

(8) Notes

(a) It is possible to select which plane will be automatically selected after power is turned on or after the NC has been reset with a parameter. Set bit 1 and bit 2 of NC optional parameter (bit) No. 18.

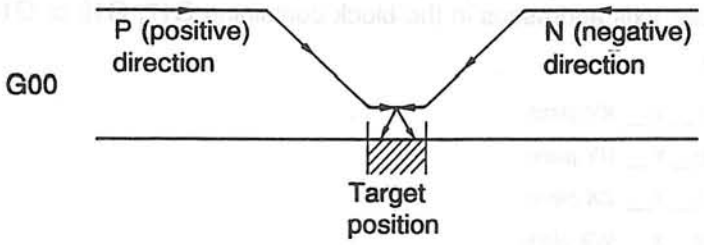
(b) An alarm is activated when both a basic axis and its parallel axis are specified within the same block selected on the plane.

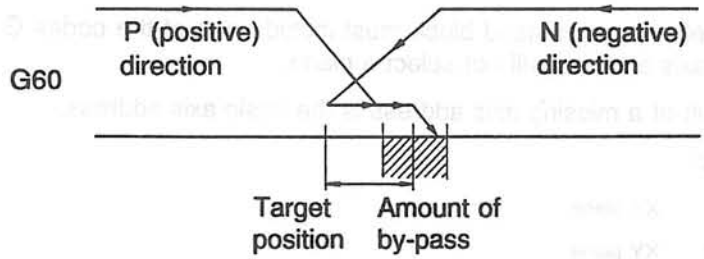
1-2. Unidirectional Positioning (G60) (Optional)

(1) The normal positioning command, G00, allows an amount of backlash to occur around the target position due to the difference between the two positioning directions, positive and negative. The backlash may be eliminated if the axis is always positioned from the same direction, thus assuring accuracy. This function, called unidirectional positioning, is specified by the G60 command.

(2) The positive (P) direction is specified by setting bits 0 through 5 of NC optional parameter (bit) No. 19.
Unidirectional positioning from the negative (N) direction is performed by passing the axis past the target point, and then approaching the target point again from the positive direction. The amount that the axis passes by the target point (by-pass amount) can be specified by user parameter No. 5.

Programming format: 【 G60 P_ 】

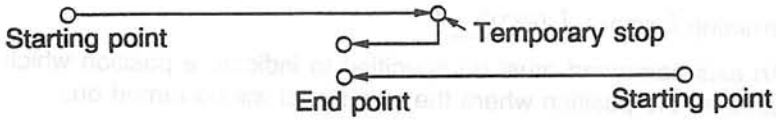




(3) The interpolation mode for positioning is selected according to the setting of the NC optional parameter (bit) No.46, bit 1. The interpolation modes available are linear and non-linear.

(4) Notes

(a) Positioning is carried out in the "positive" direction when the commanded positioning direction matches the positioning direction specified by the settings of bits 0 through 5 of NC optional parameter (bit) No.19.



(b) G60 is a modal command. (Remains in effect until a different positioning command is given.)

(c) Unidirectional positioning is not effective during a fixed cycle, in terms of cycle axis and hift movement.

(d) Unidirectional positioning is not effective on an axis for which no by-pass amount has been set.

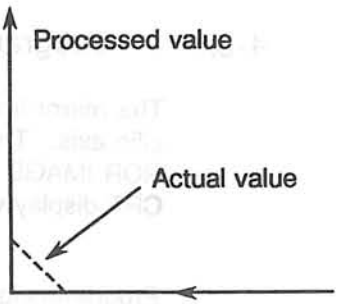
(e) No mirror image is effective in the positioning direction.

1-3. Exact Stop Check Function (G09, G61, G64) (Optional)

(1) During axis feed control, NC operation is capable of processing a block faster than the actual axis movements can be performed. If the next block is executed before the actual value reaches the first target point, axis motion will be forced towards the second target point too soon, resulting in rounded-off corners between joints.

(2) This problem may, however, be solved with the help of the exact stop check function. With this function, the next block does not start an operation until the actual value arrives at the target point (even if the NC operation has been completed earlier), so that the actual movement exactly follows the programmed path. When the actual value has reached the target point, it is said to be in position. The function allows for some margin, which may be specified by a system parameter.

(3) The exact stop check mode may be either one-shot (effective only for a programmed block) or modal, as explained in the following sections. Note that during a positioning mode (G00, G60), exact stop checks are carried out automatically independent of this command.



(a) The one-shot exact check stop is effective only for the one specified programmed block.

Programming format: [G09 P__]

(b) Once the modal exact stop check mode is specified, every block is executed with an exact stop check until a cutting mode (G64) is specified.

Programming format: [G61 P__]

(c) The G64 command is used to cancel the modal exact stop check mode (G61 command).

Programming format: [G64 P__]

The G64 command can also be added to blocks which contain a positioning command (G00, G60) or a one-shot exact stop check command (G09). This also applies to blocks involving noncontinuous cutting feed.

1-4. Skip Function (G31) (Optional)

The skip function can be used to interrupt axis movements and cause the control to "skip" to the next block. The skip command is a one-shot function which is effective only in the programmed block. The skip function is activated by an external skip signal given from a limit switch, proximity switch, or other device during axis movement. This function is useful for automatic work gauging, automatic tool gauging and other similar functions.

Programming format : [G31P__]

(1) An axis command must be specified to indicate a position which exceeds the target value, to be used as the position where the skip signal will be turned on.

(2) The coordinate values of the skip signal position are located either in the work or machine coordinate system, which is currently selected.
The skip signal position coordinate values are stored in system variables VSAP* and may be read. (The asterisk "*" is used to denote an address character representing an axis such as X, Y, or Z, and corresponds to the data of that axis. For details, refer to SECTION 11, 2-1. "System Variables".)

(3) After the skip signal has been turned on, the following operation is performed according to the dimensioning system selected in the next block.

(a) Incremental Command (G91)

The system operates incrementally, starting at the position where the axis movement was interrupted by the skip signal.

(b) Absolute Command (G90)

Only the specified axis moves to a specified position from the position where the axis movement was interrupted by the skip signal. Note that any axis not specified remains at the interrupted position.

(4) Notes

(a) The G31 command may not be used during cutter radius compensation mode.

(b) The G31 command may not be used during a fixed cycle.

(c) The override function is effective during execution of the G31 command.

1-5. Programmable Mirror Image (G62) (Optional)

The mirror image function creates a geometry which is formed in a symmetric position around a specific axis. The mirror image function may be activated by program commands or by setting the MIRROR IMAGE keys on the operation panel. An axis which is in mirror image mode is identified on the CRT display with a dash "-" which is added before the axis name on the ACTUAL POSITION page.

Programming format: [G62 P $\left\{\begin{matrix}0\\1\end{matrix}\right\}$]

Where the values of 0 and 1 which follow the axis address are interpreted as follows:

0: Normal (Mirror image off)

1: Mirror image

(1) The actual state of the mirror image function based on the specification of G62 and the MIRROR IMAGE switch setting is displayed in the table below.

G62 COMMAND	SWITCH SETTING	ACTUAL STATE
Normal	Normal	Normal
Normal	Mirror image	Mirror image
Mirror image	Normal	Mirror image
Mirror image	Mirror image	Normal

(2) Notes

(a) A block in which a G62 command is specified may not contain any other commands.

(b) The mirror image function is modal.

(c) Any axis which is not specified in the G62 block is assumed to be in normal mode.

(d) All axes are in normal mode when the power supply is turned on.

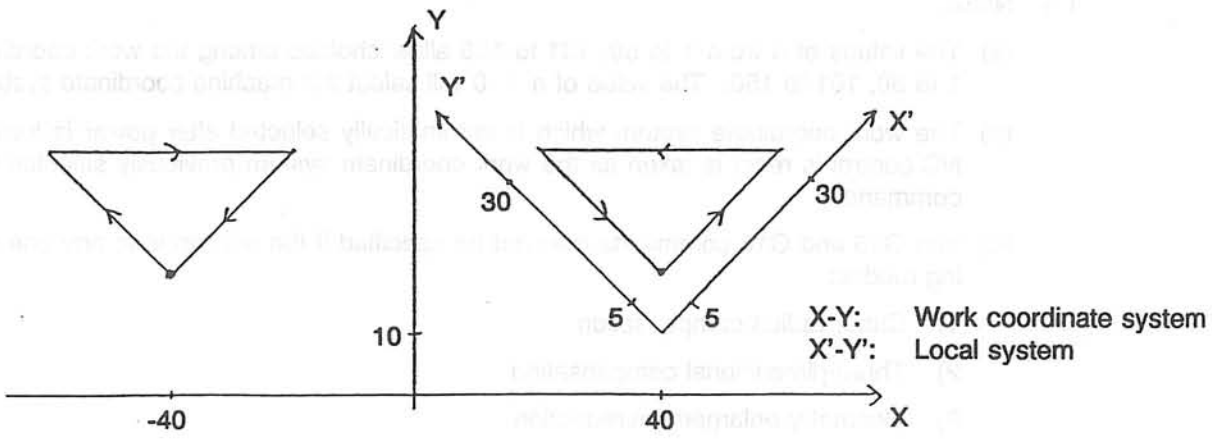
(e) The setting of bit 3 of NC optional parameter (bit) No. 17 determines whether all axes will be set in normal mode or not when the NC is reset.

(f) The setting of bit 2 of NC optional parameter (bit) No. 34 determines which coordinate system (local or work) the mirror image function will be active in.

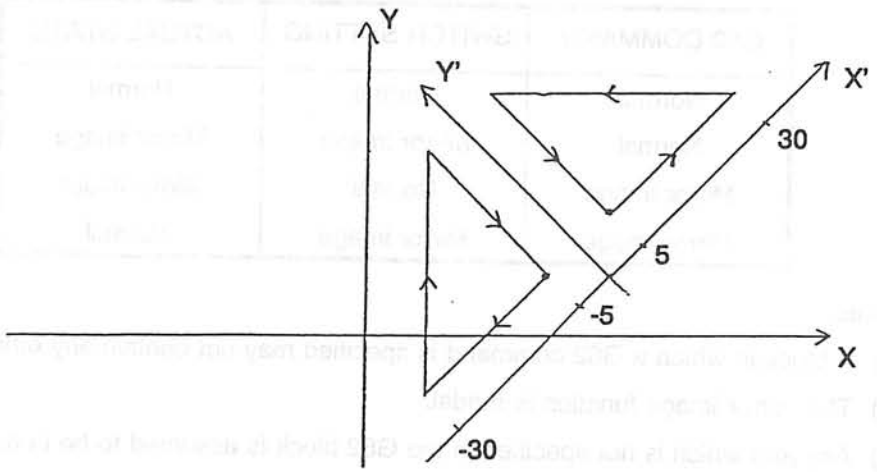
Example:

```
G11   X40  Y10   P45
G01   X5   Y5    S_ F_
      X30
      X5   Y30
      Y5
G62   X1
```

a) Setting "1" for O.P.B. No.34, bit 2 (work coordinate system)



b) Setting "0" for O.P.B. No.34, bit 2 (local coordinate system)



1-6. Selection of Work Coordinate System (G15, G16)

(1) Standard specifications provide for only one work coordinate system. Optionally, however, up to 20 (50 or 100) sets of work coordinate systems are selectable, both in the modal and one-shot mode.

(2) Programming Format:

- Programming format for modal selection:

  【 G15 Hn 】 (0 ≦ n ≦ 50、101≦ n≦ 150)

Coordinate values specified in this and the following blocks will correspond to the work coordinate system "n".

- Programming format for one-shot selection:

  【 G16 Hn 】 (0 ≦ n ≦ 50、101≦ n≦ 150)

Coordinate values specified in this block will correspond to the work coordinate system "n".

(3) Notes

(a) The values of n from 1 to 50, 101 to 150 allow choices among the work coordinate systems 1 to 50, 101 to 150. The value of n = 0 will select the machine coordinate system.

(b) The work coordinate system which is automatically selected after power is turned on or the NC control is reset is taken as the work coordinate system previously selected with the G15 command.

(c) The G15 and G16 commands may not be specified if the system is in any one of the following modes:

1) Cutter radius compensation
2) Three-dimensional compensation
3) Geometry enlargement/reduction
4) Coordinate system parallel shift/rotation

(d) The axis feed command after the G15 command shoud be specified with the absolute mode.

1-7. Change of Work Coordinate System (G92)

(1) The work coordinate system may be changed using the G92 command.

(2) Programming format: [G92 P__]

The G92 command automatically changes the work zero offset value of the work coordinate system to the coordinate values specified after G92.

(3) Notes:

(a) The G92 command changes the work coordinate system that is selected at that particular point of time, but does not affect any other work coordinate systems.

(b) In both the G90 (absolute) and G91 (incremental) modes, all coordinate values specified are absolute values.

(c) The work zero offset value does not change for an axis which is not specified in the G92 command.

(d) The G92 command may not be specified if the system is in any of the following modes:

1) Cutter radius compensation
2) Three-dimensional compensation
3) Geometry enlargement/reduction
4) Doordinate system parallel shift/rotation
5) With the machine coordinate system selected

1-8. Unit System Check (G20, G21) (Optional)

The G20 and G21 commands are used to check if the unit system selected is based on the metric or inch system, by examining bit 0 of NC optional parameter (bit) No. 3. If the wrong system is found, an alarm is activated.

(1) G20: Inch input confirmation

The G20 command seeks confirmation that the inch system has been selected. An alarm is activated if the unit system is based on the metric system.

(2) G21: Metric input confirmation

The G21 command seeks confirmation that the metric system has been selected. An alarm is activated if the unit system is based on the inch system.

2. Compensation Functions

2-1. Tool Length Offset Function (G53 - G59)

(1) The tool length offset function compensates for the position of a cutting tool so that the tip of the cutting is located at the programmed position.

Available G Codes

GCODE	FUNCTION
G53	Cancel tool length offset
G54	Tool length offset, X-axis
G55	Tool length offset, Y-axis
G56	Tool length offset, Z-axis
G57	Tool length offset, 4th-axis
G58	Tool length offset, 5th-axis
G59	Tool length offset, 6th-axis

(2) Programming format: [{G54 - G59} P__ H__]

P: Current position of tool tip after compensation

H: Offset register number
The standard H numbers are H00 through H50, and the optional H numbers are H00 through H100, H200, H300.
The offset amount of H00 is always zero.
This offset amount can be set in the tool data setting mode and may range from 0 to ±999.999 mm (0 to ±99.9999 inches).

(3) Notes

(a) The displayed active tool position value always includes the tool length offset amount.

(b) The tool length offset can not be applied to two or more axes at the same time or to the rotary axis.

(c) The tool length offset may be changed directly without having to cancel the previous command with the G53 command.

(d) After the NC control has been reset, H000 is automatically set.

2-2. Spindlehead Index Position Compensation

The MCM has a total of five spindlehead positions: V (vertical), HF (horizontal, front), HL (horizontal, left), HB (horizontal, back), and HR (horizontal, right). It is necessary to compensate the tool tip position for each of these spindle positions since the absolute position differs depending on the selected spindlehead position. This may be automatically implemented by the function called the spindlehead index position compensation.

(1) Reference Point

With a reference tool mounted at each position, the individual tool tip positions are V, HF, HL, HB, and HR, and the V position is the reference point for compensation.

(2) Compensation Amount

The relative position from the V reference point is the compensation amount of each tool tip position (HF, HL, HB, HR).

POSITION	COMPENSATION AMOUNT (mm)		
	X	Y	Z
HF (M73)	400.000 + α1	−150.000 + α5	−330.000 + α9
HL (M74)	−150.000 + α2	−400.000 + α6	−330.000 + α10
HB (M75)	−400.000 + α3	150.000 + α7	−330.000 + α11
HR (M76)	150.000 + α4	400.000 + α8	-330.000 + α12

Note: α_1 to α_{12} are used to denote the indexing error at each position.

(3) Setting Compensation Amount

The compensation amount is set using a system parameter.

(4) Operating Condition

The spindlehead index position compensation function is effective only in the AUTO and MDI modes.

(5) Compensation is effective only when the G54 or G55 (tool length offset X-axis/ Y-axis) command has been specified:

G54 .... Horizontal spindle front and back

G55 .... Horizontal spindle right and left

An alarm is activated when the spindle position and specified G code do not match.

2-3. Cutter Radius Compensation (G40, G41, G42)

2-3-1. Function

Programming the geometry of a workpiece as it is will not result in a correct final product, because the size (diameter) of the tool is not taken into consideration. It would, however, be extremely complicated and difficult to develop a program which takes the tool diameter into account. This problem may be solved by a function called cutter radius compensation which automatically compensates for the tool diameter. A series of commands to make a path according to the machining geometry will produce the path of the tool center which includes the necessary compensation.

Programming format:

[G17 G41 (G42) Xp__ Yp__ D__]

[G18 G41 (G42) Zp__ Xp__ D__]

[G19 G41 (G42) Yp__ Zp__ D__]

(1) G codes used for cutter radius compensation

(a) G40: Cancel cutter radius compensation

This mode is selected when power supply is turned on.

(b) G41: Cutting at left

Offset to the left side as seen from the tool advancing direction; downward cutting.

(c) G42: Cutting at right

Offset to the right side as seen from the tool advancing direction; upward cutting.
Either the G41 or G42 commands activate the cutter radius compensation mode, which may be canceled by a G40 command.

(d) G17: Xp-Yp plane selection
(e) G18: Zp-Xp plane selection
(f) G19: Yp-Zp plane selection

} The plane upon which cutter radius compensation should be activated is specified with a G17, G18, or G19 command.

(g) D: Cutter radius compensation number. (For details, see 2-3-6. "Notes on Cutter Radius Compensation" in this section.)

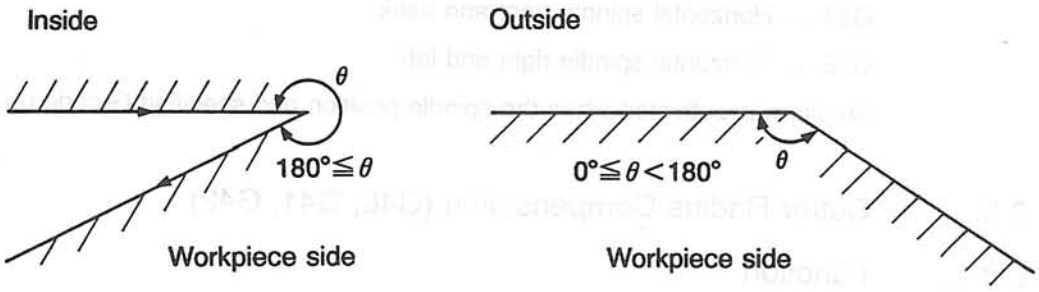
(2) Notes

(a) The G17 command (Xp-Yp plane) is automatically selected when power is turned on.

(b) Only the G00 and G01 modes may be used for controlling axis movements when entering the G41 or G42 cutter radius compensation mode. An alarm will be activated if other axis movement modes are specified.

(c) The mode is changed to the radius compensation mode after the block which contans the axis movement command.

(d) The terms "inside" and "outside" are defined as follows:
Inside cutting results if the angle between the move commands is at least 180 degrees as measured from the workpiece side, and outside cutting refers to any angle between 0 and 180 degrees.



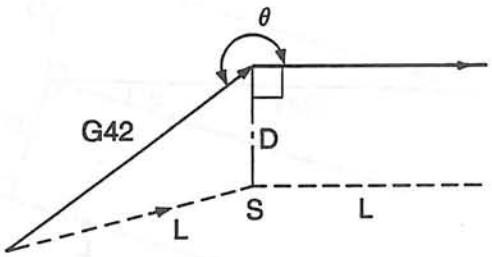
The following programming descriptions use symbols and lines which are defined here:

S : Single block stop point
L : Linear motion
C : Circular motion
T : Tangent to an arc
D : Cutter radius compensation amount
θ : Angle at the workpiece side
CP: Cross point made when a programmed path (or the tangent to an arc) is shifted by a compensation amount
——► : Programmed tool path
·····► : Tool center path
—·—·— : Auxiliary line

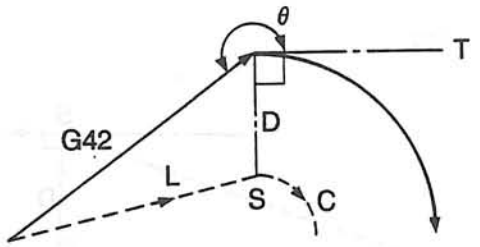
2-3-2. Tool Movement in Start-up

(1) Inside corner cutting ($\theta \geqq 180°$)

(a) Straight line - straight line

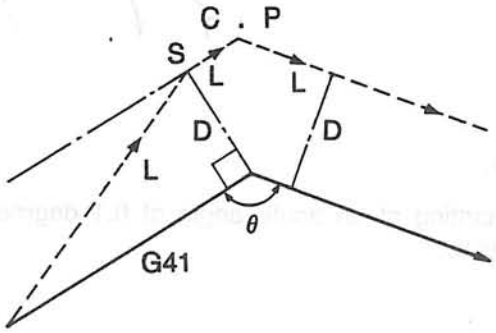


(b) Straight line - arc

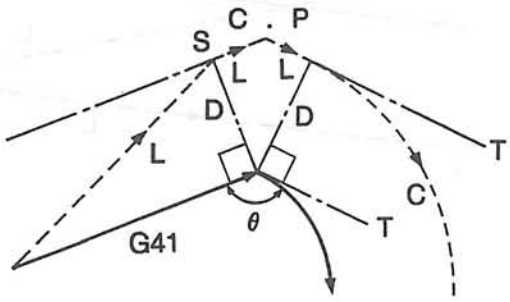


(2) Obtusely angled corner - outside cutting ($90° \theta \leqq 180°$)

(a) Straight line - straight line

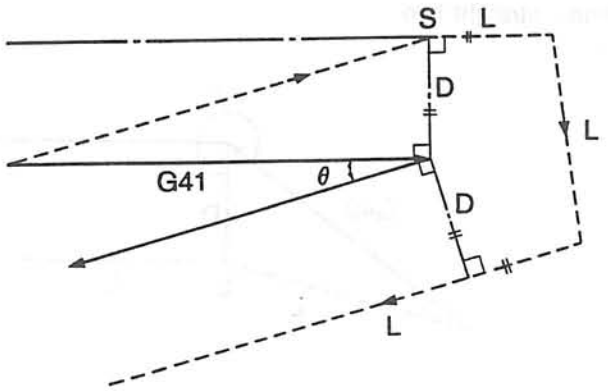


(b) Straight line - arc

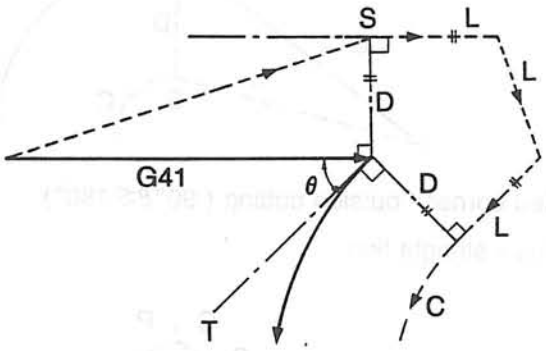


(3) Acutely angled corner - outside cutting ($\theta < 90°$)

(a) Straight line - straight line



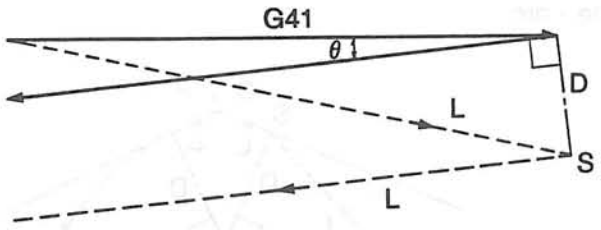
(b) Straight line - arc



(c) Exception

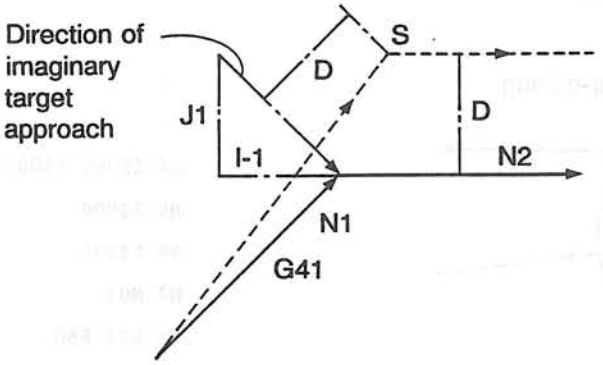
Outside cutting at an acute angle of 0.1 degrees or less is considered to be "inside" as shown below.

$\theta < 0.1°$

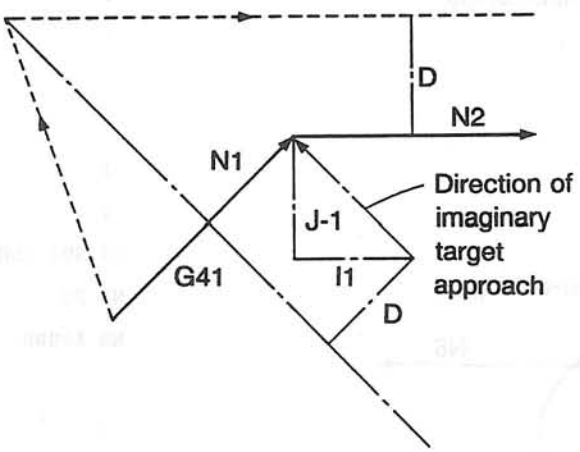


(4) Start-up with imaginary approach direction

If I__, J__, and K__ values belonging to the offset plane are specified in a block which starts the cutter radius compensation mode, a different tool center path will be chosen. In the G17 plane, values for I__ and J__ will create an imaginary (target) approach direction. The imaginary tool center path is located a distance D (compensation amount) away from the imaginary target approach direction. Point "S" is then defined as the stop point which would have resulted from the imaginary tool center path. The system moves to point S, always attempting to find a cross point, whether cutting is inside or outside.

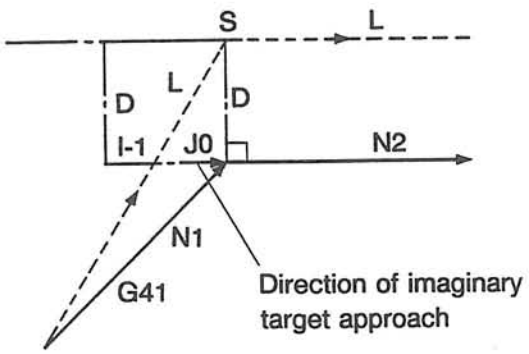



```
N1  G41   X5000 Y5000 I-1J1D1
N2        X100000
```




```
N1  G41   X5000 Y5000 I1J-1D1
N2        X100000
```

If no cross point exists, the command value (target point) of G41 is used to establish the position which is vertically shifted by the compensation amount, D.

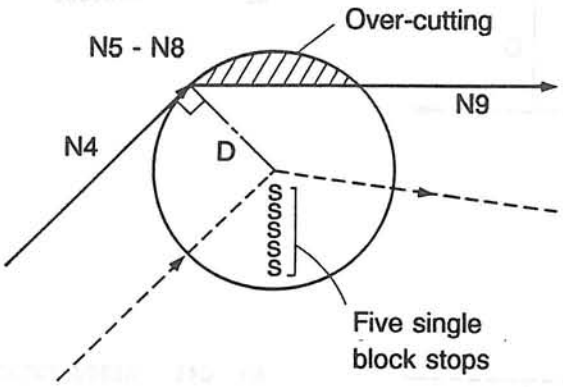



```
N1  G41   X5000 Y5000 I-1J0
N2        X100000
```

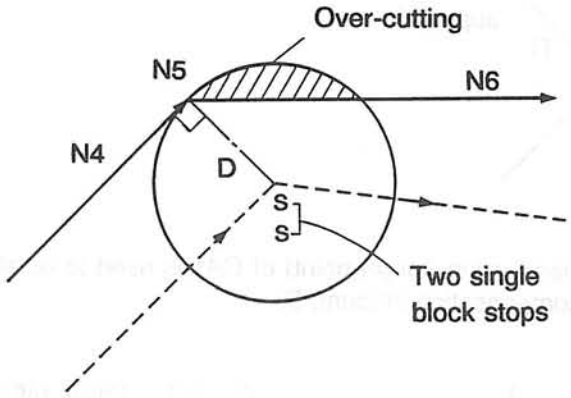
2-3-3. Tool Movement in Cutter Radius Compensation Mode

This section describes how the tool moves after the cutter radius compensation mode has been established until that mode is canceled. The cutter radius compensation is available in four different move modes: G00, G01, G02, and G03.

During cutter radius mode, up to three blocks which do not contain a command involving travel along the axis in the selected plane may be given successively. If such a command (no axis travel in the selected plane) occurs in four or more successive blocks, then the process below takes place, resulting in over- or under-cutting. Such commands, and even single blocks which cause no axis travel on the selected plane, should be avoided.



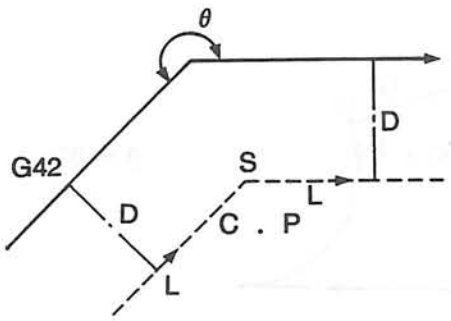

```
:
:
N4 X5000 Y5000
N5 Z5000
N6 F1000
N7 M01
N8 G04 F50
N9 X100000
:
:
```



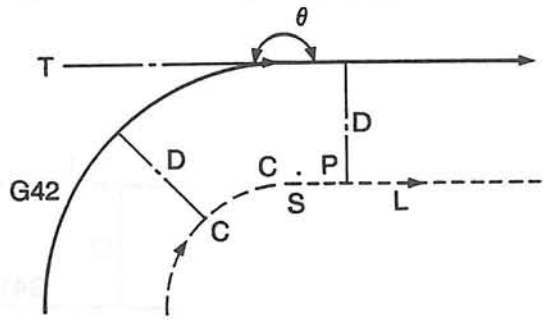

```
:
:
N4 G91 X5000 Y5000
N5 X0
N6 X5000
:
:
```

(1) Inside cutting (180° ≦ θ)

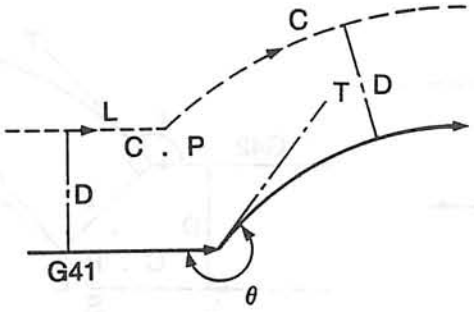
(a) Straight line - straight line



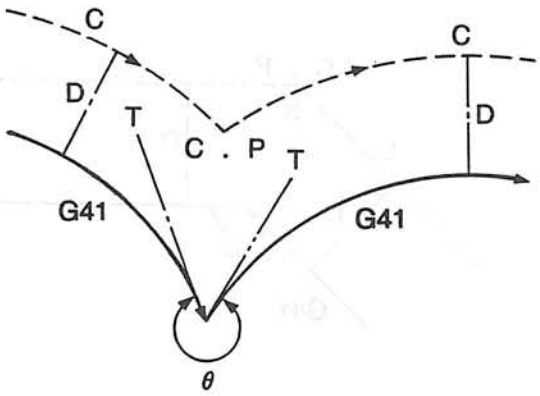
(b) Arc - straight line



(c) Straight line - arc

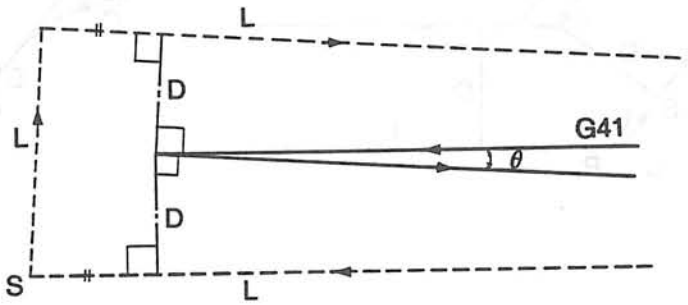


(d) Arc - arc



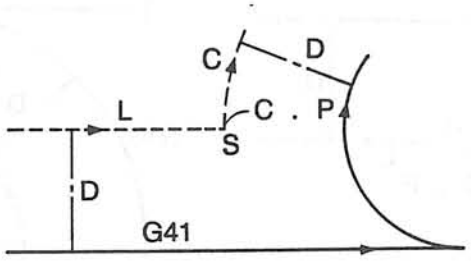
(e) Straight line - straight line (θ < 0.1°)

There is an exception in processing where inside cutting at 0.1 degrees or less for the straight line - straight line configuration is replaced by outside cutting (to be explained later) because the ordinary method of finding the cross point will significantly deviate from the command value.



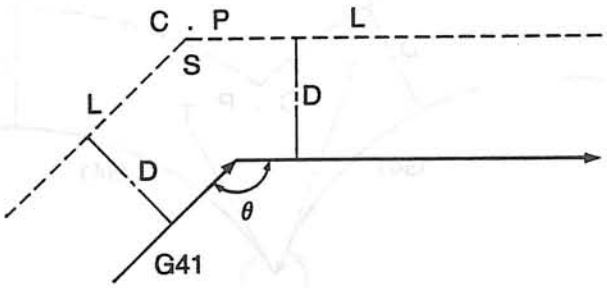
(f) Straight line - arc (θ = 0 °)

The processing shown in part e is limited to the straight line - straight line configuration. In other cases, such as the straight line - arc shown here, the ordinary method is used.

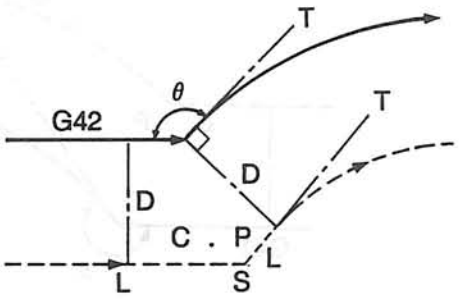


(2) Obtusely angled corner - outside cutting ($90° \leqq \theta < 180°$)

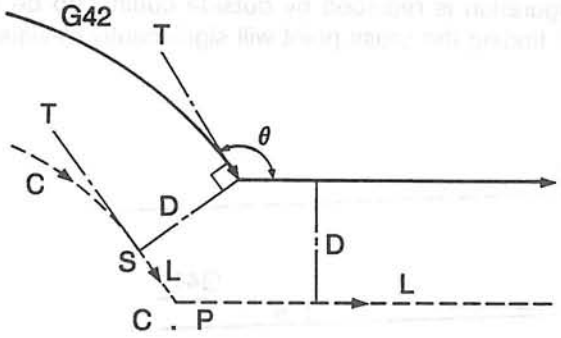
(a) Straight line - straight line



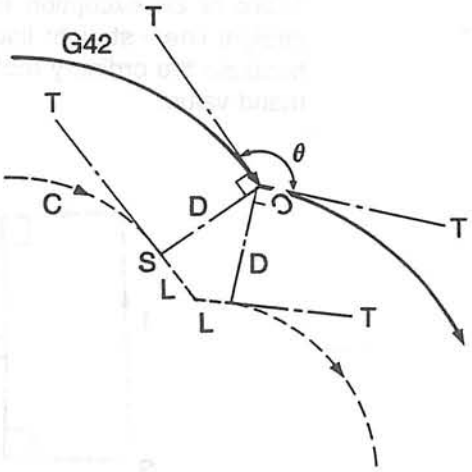
(b) Straight line - arc



(c) Arc - straight line

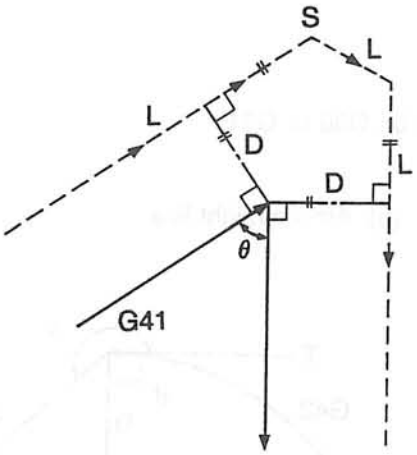


(d) Arc - arc

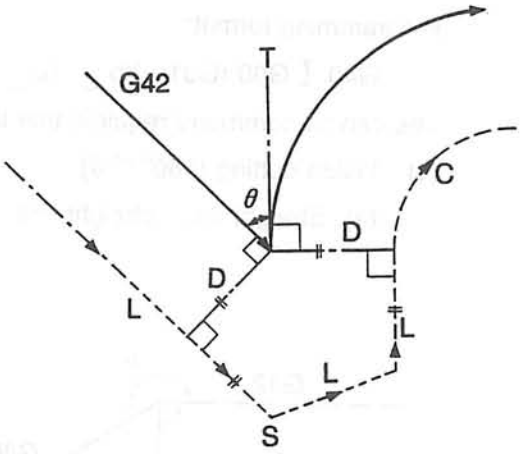


(3) Acutely angled corner - outside cutting ($\theta < 90°$)

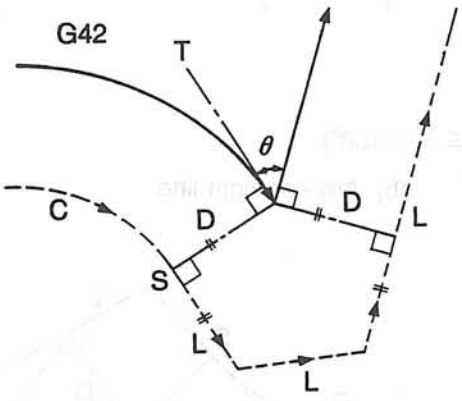
(a) Straight line - straight line



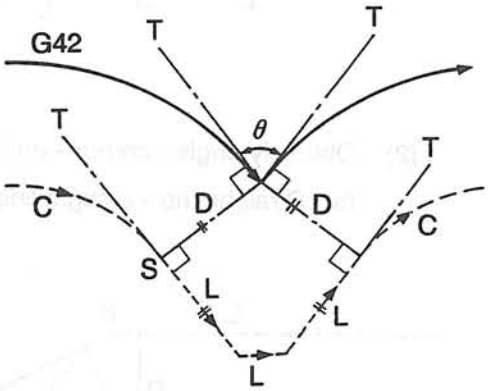
(b) Straight line - arc



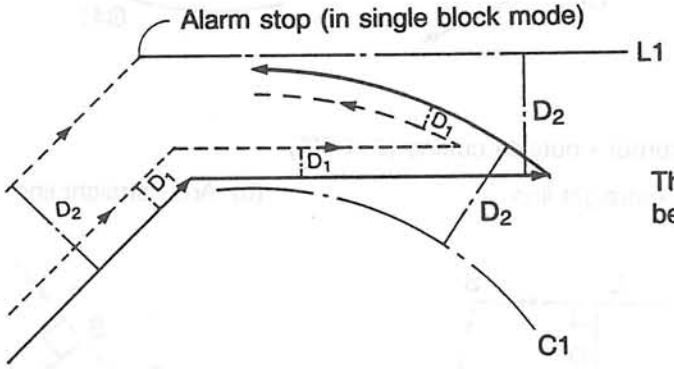
(c) Arc - straight line



(d) Arc - arc



(4) Inside cutting, with failure to find cross point



There is no cross point between L1 and C1.

As shown in the above figure, there may be situations in which a cross point exists with a small compensation amount (D1), but not with a large compensation amount (D2). In this case, an alarm is activated and operation stops.
In single block mode, the alarm is activated in the block which precedes the one which will cause an alarm state. In other modes, the alarm is activated several blocks before the block causing no cross point.

2-3-4. Tool Movement When Cutter Radius Compensation is Canceled

A G40 command given during the cutter radius compensation mode will cause the mode to end.

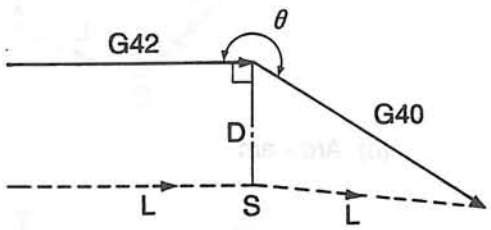
Programming format:

G40 【 G00 (G01) Xp__ Yp__ 】

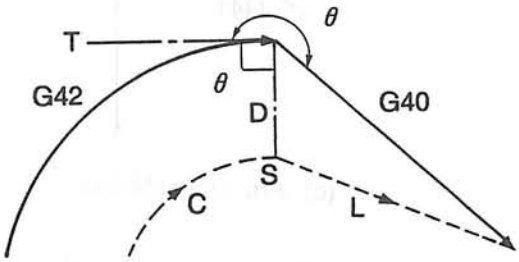
The cancel command requires that the transient mode be G00 or G01.

(1) Inside cutting ($180° \leqq \theta$)

(a) Straight line - straight line

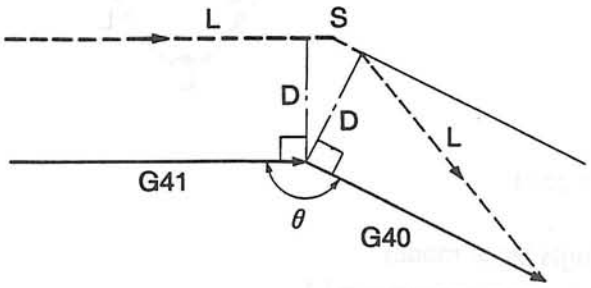


(b) Arc - straight line

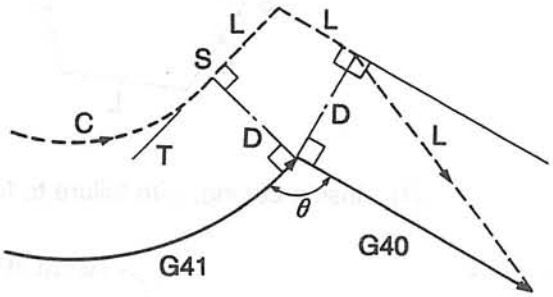


(2) Obtusely angled corner - outside cutting ($90° \leqq \theta < 180°$)

(a) Straight line - straight line

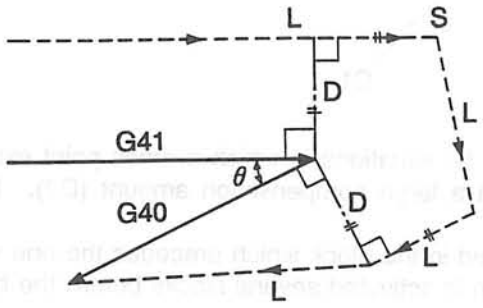


(b) Arc - straight line

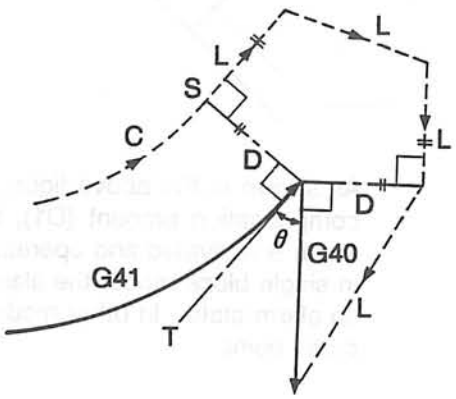


(3) Acutely angled corner - outside cutting ($\theta < 90°$)

(a) Straight line - straight line

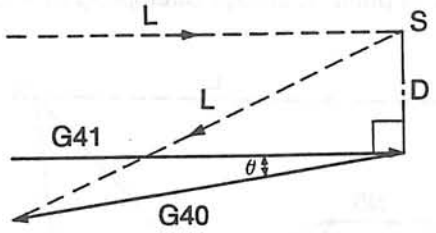


(b) Arc - straight line



(c) Exception

Outside cutting at an acute angle of 1 degree or less is considered to be "inside" as shown below.

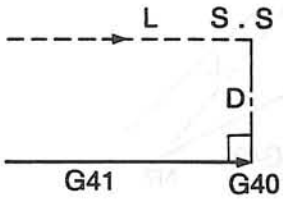


$\theta \leqq 1°$

(4) Independent G40 command

A G40 command given independently will position the axes at a point shifted in the vertical direction by an amount equivalent to the compensation amount (D) from the position given in the preceding block.

Straight line

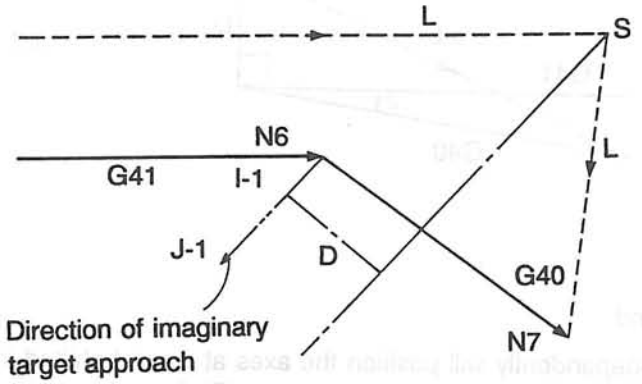


Two single block stops

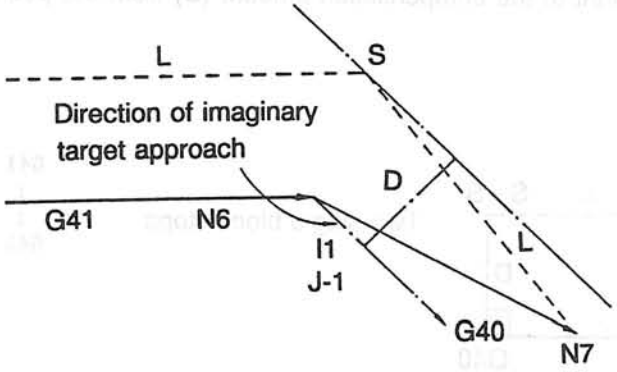
```
G41 X_ Y_
  :
  :
G40
```

(5) Cancel with imaginary approach direction

(a) If I__, J__, and K__ values belonging to the offset plane are specified in the same block as G40, a different tool center path will be chosen. In the G17 plane, values for I__ and J__ will create an imaginary (target) approach direction. The imaginary tool center path is located a distance D (compensation amount) away from the imaginary target approach direction. The system moves to point S, always attempting to find a cross point, whether cutting is inside or outside.

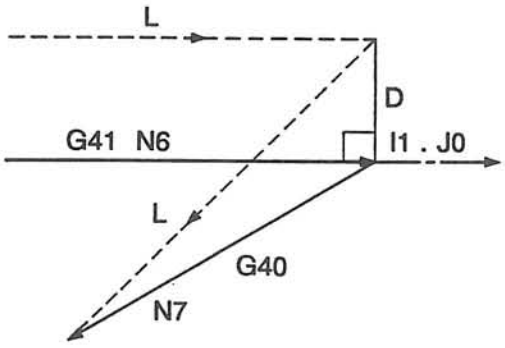



```
N6 G41 X10000
N7 G40 X20000 Y5000 I-1 J-1
```




```
N6  G41  X10000
N7  G40  X20000 Y5000 I1 J-1
```

(b) If no cross point exists, the system will position the axes at a point shifted in the vertical direction by an amount equivalent to the compensation amount (D) from the end point given in the block preceding the G40 block.




```
N6 G41  X10000
N7 G40  X5000 Y5000 I1 J0
```

2-3-5. Changing Compensation Direction During Cutter Radius Compensation Mode

The direction of compensation may be changed during the cutter radius compensation mode by issuing a G41 or G42 command or by reversing the sign (positive or negative) of the compensation amount.

Positive / negative sign / G CODE	+	−
G41	Offset to left (cutting left side)	Offset to right (cutting right side)
G42	Offset to right (cutting right side)	Offset to left (cutting left side)

Execution Conditions

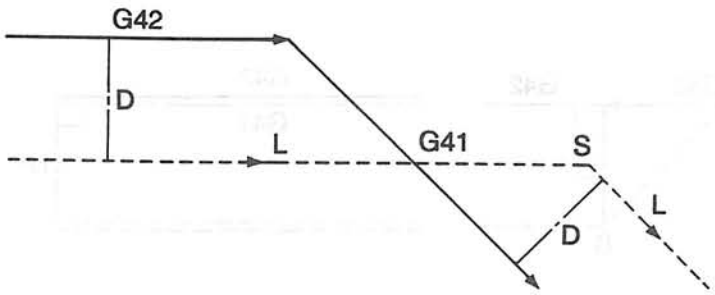
Mode	CommanD	Straight line - straight line	Straight line - arc	Arc - straight line	Arc - arc
G41	G41	Not effective (When the plus or minus sign of the offset amount is not changed.)			
G42	G42				
G41	G42	Executable		Alarm if no cross point exists	
G42	G41				

The change in the offset direction involves no distinction between inside and outside cutting, but may differ depending on the presence of a cross point.

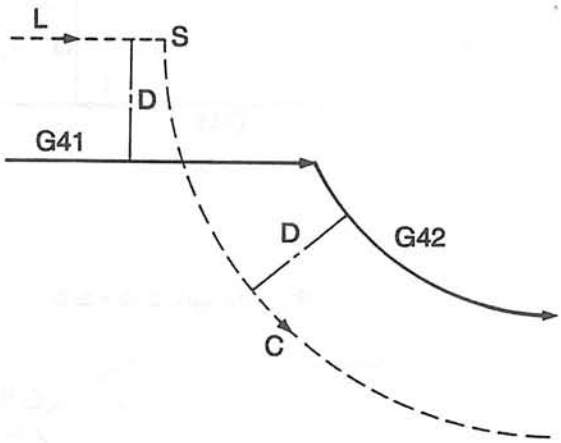
The following descriptions assume that the compensation amount is positive.

(1) With cross point

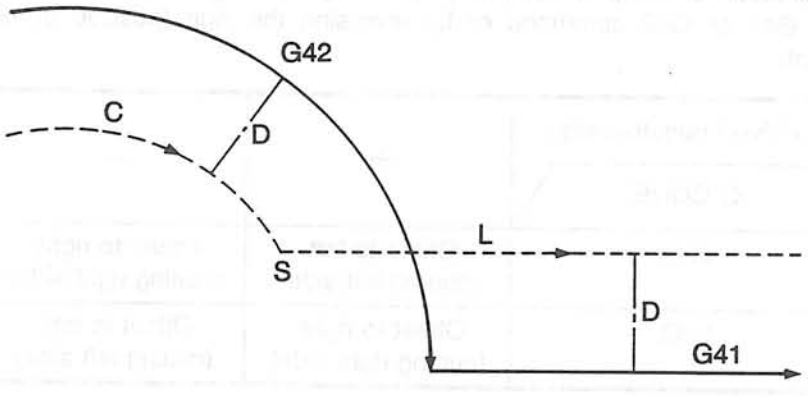
(a) Straight line - straight line



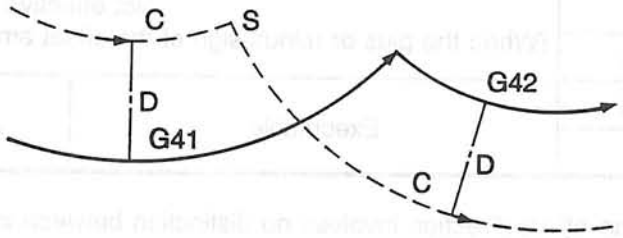
(b) Straight line - arc



(c) Arc - straight line

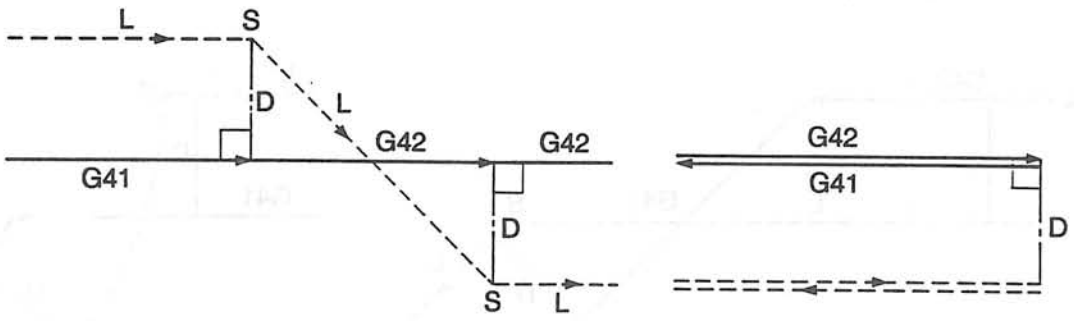


(d) Arc - arc

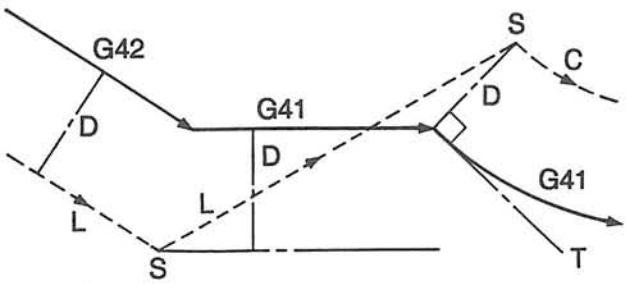


(2) Without cross point

(a) Straight line - straight line

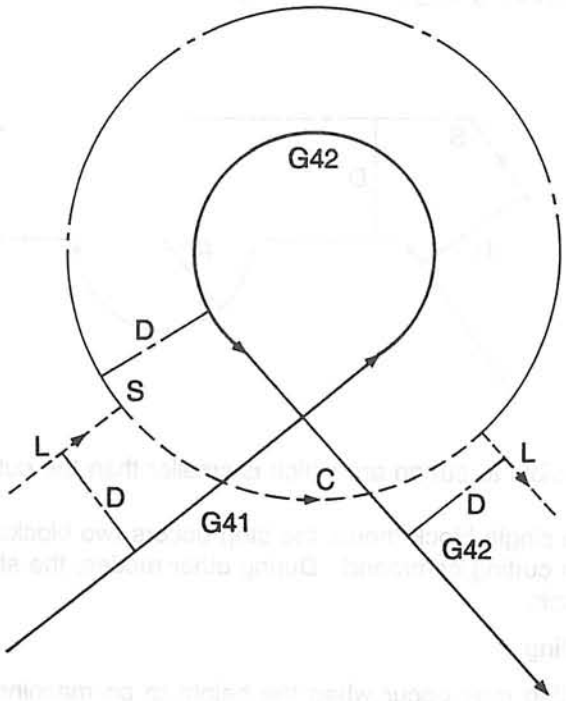


(b) Straight line - arc



(3) Circular arc forming an overlapping circle

The tool path may form an overlapping circle after compensation direction has been changed. The cutter will move along an arc and reach the target point without forming a full circle. To avoid this, the commands should be given in more than two blocks.



2-3-6. Notes on Cutter Radius Compensation

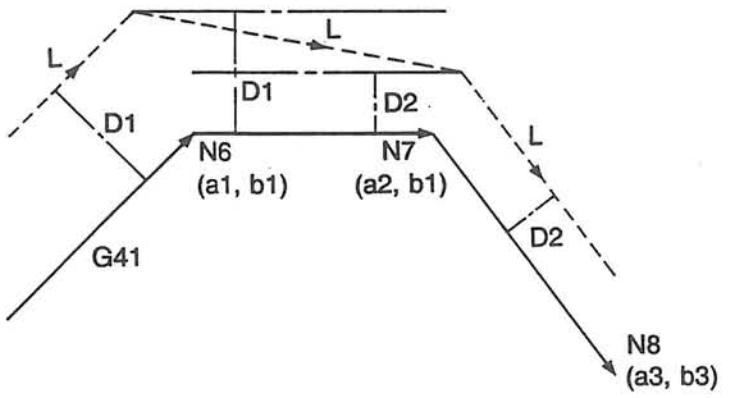
(1) Specifying the cutter radius compensation amount

(a) The compensation amount is specified with a D value which is included in the same block as a G41 or G42 command. If no value is included in a G41 or G42 command, the previous D value is chosen as the default value.

(b) The compensation amount may be selected from the standard values D00 through D50 or from the optional values D100, D200, and D300. A compensation amount of D00 is equal to zero (0). The compensation data is set in the tool data setting mode.

(2) Change of compensation amount

If the compensation amount is changed during the compensation mode, the change will become effective starting at the end of that block.




```
N1  G41X-Y-D1
 :
 :
N6  Xa1 Yb1
N7  Xa2   D2
N8  Xa3 Yb3
```

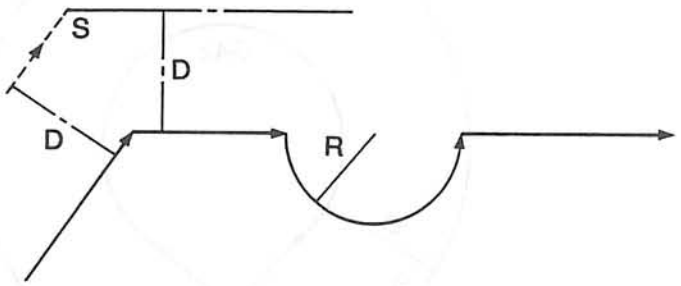
(3) Actual position data display

The coordinates of the cutter center can be displayed.

(4) Inside cutting of an arc smaller than the cutter radius

Alarm stop (during single block mode)

$R < D$

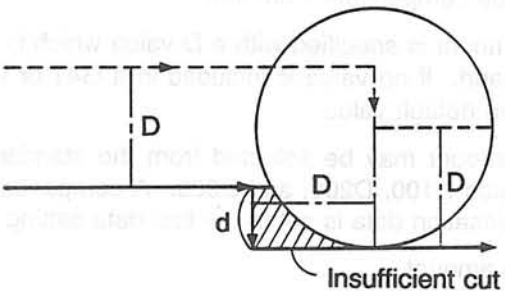


It is impossible to cut an arc which is smaller than the cutter radius, and if attempted, will activate an alarm.
During the single block mode, the stop occurs two blocks preceding the block which contains the impossible cutting command. During other modes, the stop takes place several blocks preceding such a block.

(5) Under-cutting

Under-cutting may occur when the height to be machined is smaller than the cutter radius, and an alarm will sound.

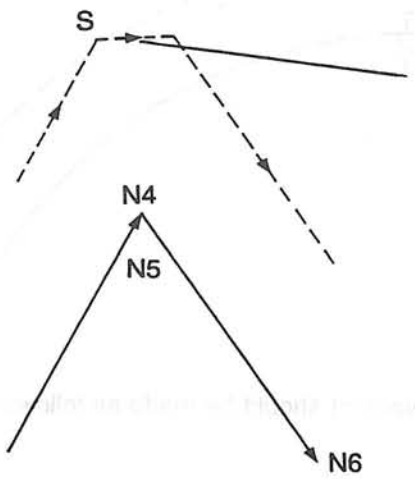
$d < D$



(6) Suggestions for Corner Cutting

(a) Normal corner

To cut an outside corner, a polygonal tool path can be generated. The move mode and feedrate at the corners will follow the command in the next block, except for the G02 and G03 modes, in which case G01 is assumed.



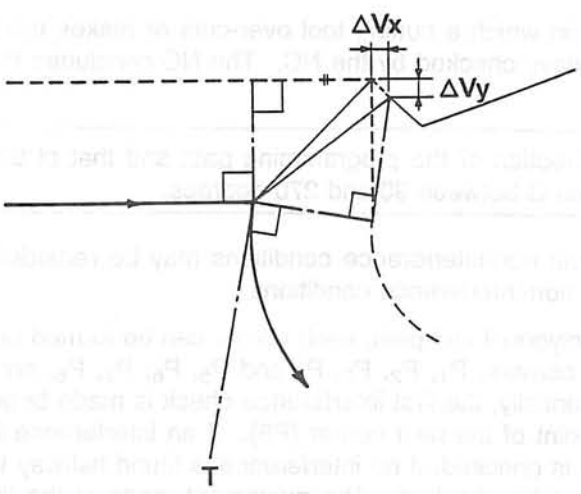
This travel follows the command of N5 (F800).

```
N4 X_ Y_    F500
N5 Z_       F800
N6 X_ Y_
```

The travel of the Z-axis is made at point S.

(b) Sharp corner

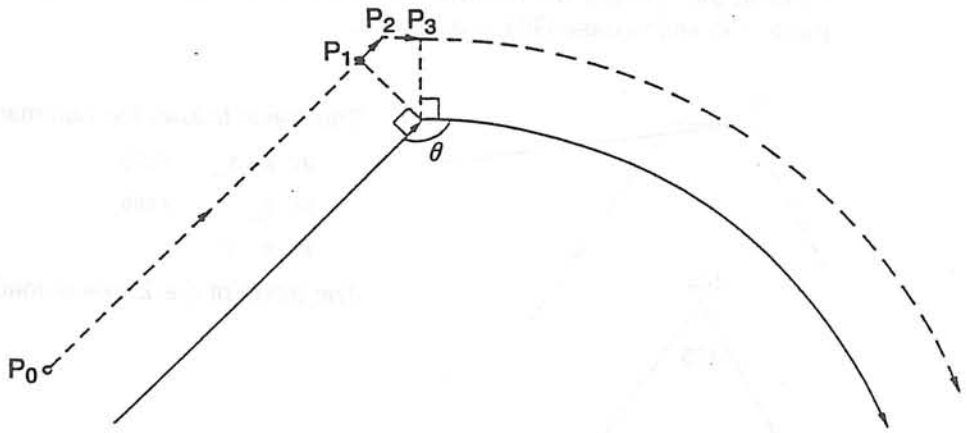
A parameter, delta-V, is used to evaluate whether a corner is too sharp for a polygonal tool path. Delta-V is initially set to 0.050 mm (0.002 in.), but can be changed by setting NC optional parameter (long word) No. 4. The figure below shows how the values delta-Vx and delta-Vy can be determined. If both delta-Vx and delta-Vy are smaller than or equal to delta-V, ($\Delta Vx \leqq \Delta V$ and $\Delta Vy \leqq \Delta V$) the corner is considered too sharp for a polygonal tool path and the extra move command (move point) will be neglected. In this manner, the additional minute travel may be reduced.



This point is neglected when $\Delta Vx \leqq \Delta V$ and $\Delta Vy \leqq \Delta V$.

(c) Corner with full circle

The processing method using delta-V is ineffective when the next block forms a full circle.



In the figure shown above, correct movement should be made as follows:

P_0 - P_1 - P_2	Straight line
P_2 - P_3	Straight line
From point P_3	Full circle

However, if the movement from point P_2 to point P_3 is neglected due to the delta-V (minute travel) processing, the movement up to point P_3 is made as follows:

P_0 - P_1 - P_2	Straight line
P_2 - P_3	Arc

Thus, the program generates a minute circle (ar(c) from P_2 to P_3 and ignores the full circle that should be generated after P_3.

(7) Interference

Interference refers to problems in which a cutting tool over-cuts or makes too deep a cut into a workpiece. Interferences are always checked by the NC. The NC concludes that an interference has occurred if:

> The difference between the direction of the programming path and that of the travel resulting from cutter radius compensation is between 90 and 270 degrees.

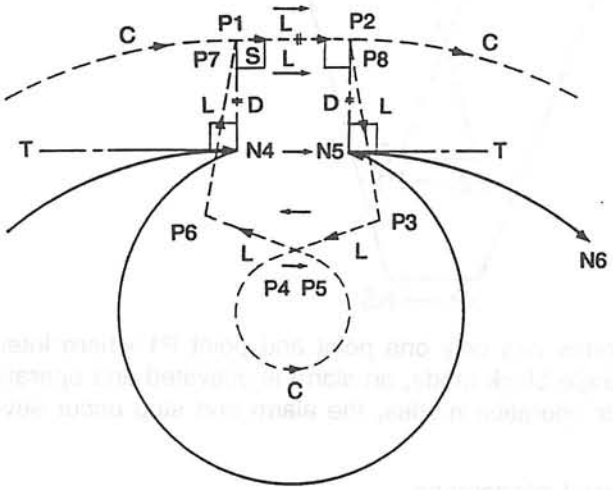
It is therefore possible that some non-interference conditions may be regarded as interferences and some real interferences as non-interference conditions.

When a corner is cut along a polygonal tool path, each corner can be formed of up to four points. To check for interference, two corners, P_1, P_2, P_3, P_4 and P_5, P_6, P_7, P_8, are evaluated. Interference checks are made sequentially; the first interference check is made between the last point of a corner (P4) and the first point of the next corner (P5). If an interference is found, the point is neglected and the next point is checked. If no interference is found halfway through the procedure, the following points will not be checked. The movement mode at the time is straight line movement. During a circular interpolation block, G01 linear interpolation mode is effective.

If an interference remains after all points have been checked, an interference alarm is activated and the very last point is not neglected. As a result, it is possible for on over cut to occur during single block mode.

Typical Examples for Explanation

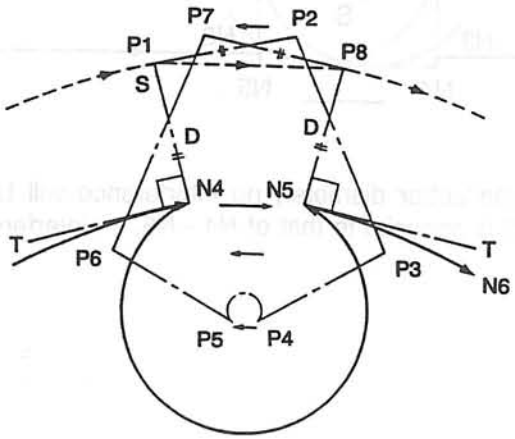
(a) Interference not checked



Since P4 - P5 is good, the other points are not checked.

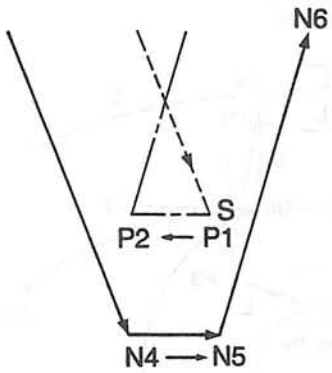
In this example, no interference is found in the first check N4 → N5 or the second check P4 → P5. Therefore, no checks are made on the following points and the interference is not discovered.

(b) Interference check resulting in a path change



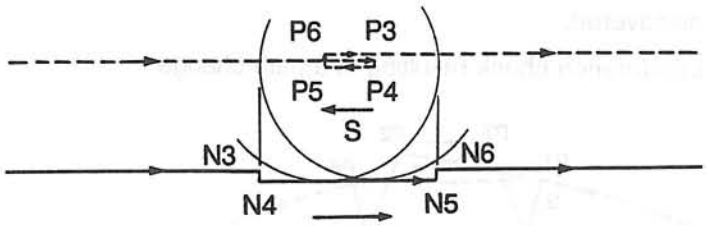
In this example, the following directions of travel are checked and neglected, because interference is discovered: N4 → N5, P4 → P5, P3 → P6 and P2 → P7. However, since the direction P1 P8 is good, this path (P1 → P8) is followed in G01 mode.

(c) Interference check resulting in alarm



In this example, each corner has only one point and point P1 where interference occurs is not neglected. During single block mode, an alarm is activated and operation stops when P1 is reached. During other operation modes, the alarm and stop occur several blocks before P1.

(d) Non-interference considered interference

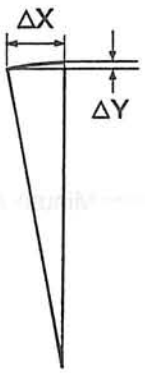


If N4 - N5 is smaller than the cutter diameter, no interference will take place. However, since the direction of P4 - P5 is opposite to that of N4 - N5, an interference alarm will be activated.

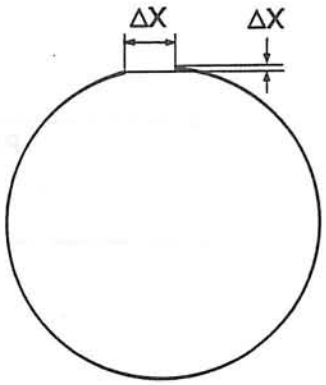
(e) Minute arc and quasi-full circle

A minute arc is defined as an arc in which the horizontal and vertical distances from beginning to end is smaller than the data set in NC optional parameter (long word) No. 9.
A quasi-full circle is defined as an arc which is close to a full circle; the horizontal and vertical distances of the break is smaller than the data set in the NC optional parameter (long word) No. 9.

Minute Arc

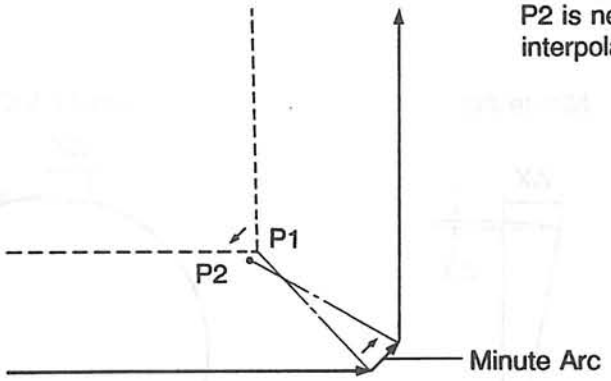


Quasi-full Circle



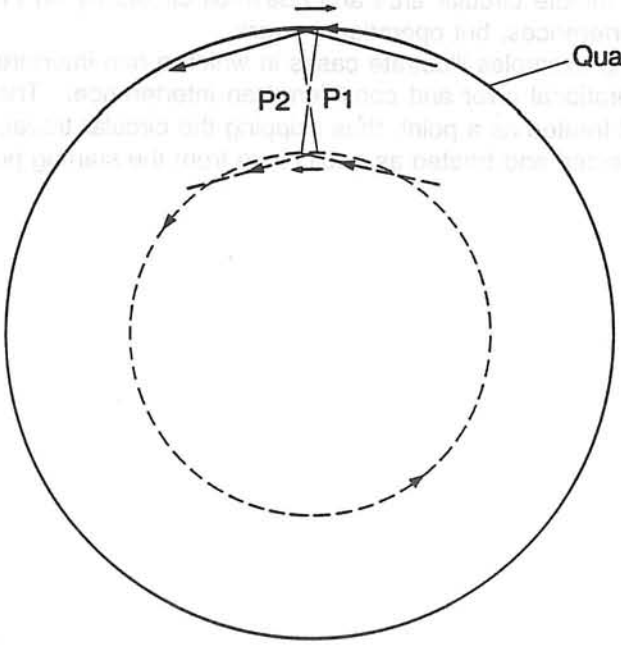
The geometry indicated above is assumed when $\Delta X \leqq \Delta V$ and $\Delta Y \leqq \Delta V$, where ΔV is the data set in NC optional parameter (long word) No. 9.
These two types of arcs are subjected to special interference checks. "Problem" conditions detected in minute circular arcs and quasi-full circles by an interference check are not considered interferences, but operational errors.
The following examples illustrate cases in which a non-interference condition is viewed as an internal operational error and considered an interference. The end of the minute arc is neglected and treated as a point, thus skipping the circular travel. The end of a quasi-full circle is also neglected and treated as a full circle from the starting point.

Minute arc



P2 is neglected and no circular interpolation occurs.

Quasi-full Circle

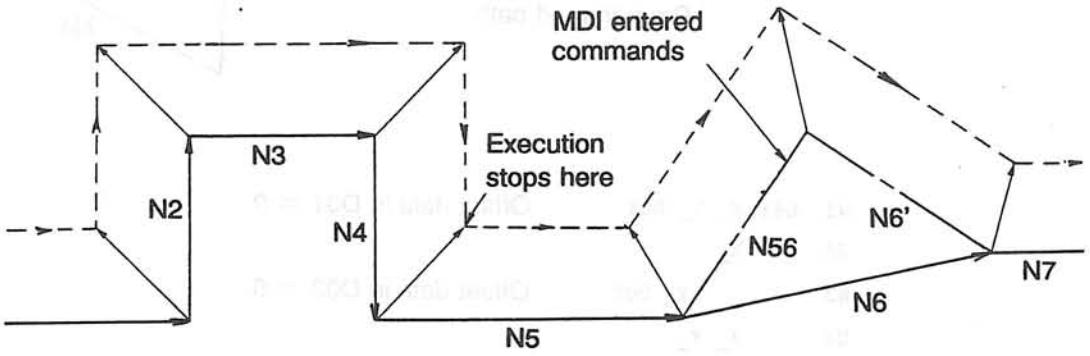


P2 is neglected and a full circle from P1 is formed.

(8) Input Command from the Keyboard

(a) Commands can be entered from the keyboard in MDI mode after the cutter radius compensation mode has already been established. However, a block containing axis movement commands which is manually inputted will be executed only after another axis movement command block or after four successive non-axis-movement commands.

(b) The AUTO mode can be switched to MDI mode only with the single block mode off. Blocks will continue to be executed automatically up to the block which was stored in the buffer when the MDI mode was turned on. (The block in the buffer is identified by a "⩾" symbol on the CRT.) The system stops, and commands may be entered through the keyboard. The keyboard input will be stored after the block in the buffer. Cutter radius compensation will then resume operation. The block in the buffer will be executed first, followed by the manually entered commands, and then the programmed blocks.



(c) Example: Suppose that MDI mode has been established while block N1 is being executed. The CRT displays the program as shown in Figure 1. Operation continues until block N4 has been reached. Figure 2 shows the CRT display at this point.

```
      :
      :
↑ N1  X10
  N2        Y30
  N3  X30
  N4        Y-30
⩾ N5  X50
  N6  X80   Y10
  N7  X100
      :
      :
```

Figure 1.

```
      :
      :
  N1  X10
  N2        Y30
  N3  X30
↑ N4        Y-30
⩾ N5  X50
  N6  X80   Y10
  N7  X100
      :
      :
```

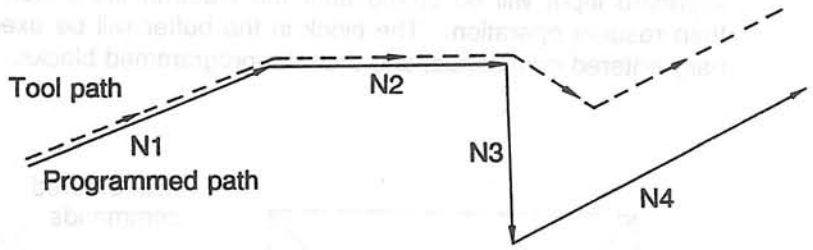
Figure 2.

(d) Commands may now be entered manually from the keyboard. Block N56 is entered. The system executes block N5 and comes to a halt. Select the AUTO mode and press the **CYCLE START** key. The system will execute N56, N6 and N7, in that order.

(9) Zero Cutter Radius Compensation Amount

(a) During start-up

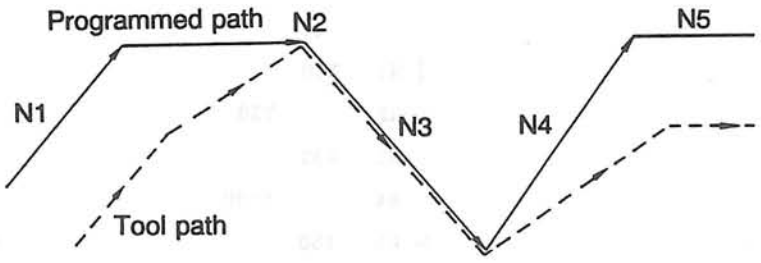
The cutter radius compensation mode may be started, while in cancel mode, with the compensation amount (D) set to zero. The G41 or G42 command will be executed with zero offset. During compensation mode, the compensation amount (offset dat(a) can be changed to another value. The effect is the same as when the compensation amount is changed as explained in 2-3-6. (2) "Change of compensation amount" in this section.




```
N1  G41 X_ Y_ D01        Offset data in D01 = 0
N2      X_
N3          Y_ D02       Offset data in D02 ≠ 0
N4      X_ Y_
```

(b) During compensation mode

Changing the compensation amount (D, offset dat(a) to zero during cutter radius compensation mode will neither cause an offset cancel nor result in the cancel mode. The effect is the same as when the compensation amount is changed as explained in Section 2-3-6. (2) A similar operation takes place when the compensation amount is changed from zero to a non-zero value.




```
N1      X_ Y_
N2      X_ D01           Offset data in D01 = 0
N3      X_ Y_
N4      X_ Y_ D02        Offset data in D02 ≠ 0
N5      X_
```

2-4. Cutter Radius Compensation Mode Override Function

(1) Automatic Override at Corners

During cutter radius compensation mode, the cut depth may increase while cutting the inside of a corner, resulting in an increased tool load. To reduce the load applied to the tool, an automatic override operation is put into effect.

(a) System parameters

The system requires four values to be set directly in the PARAMETER SET mode:

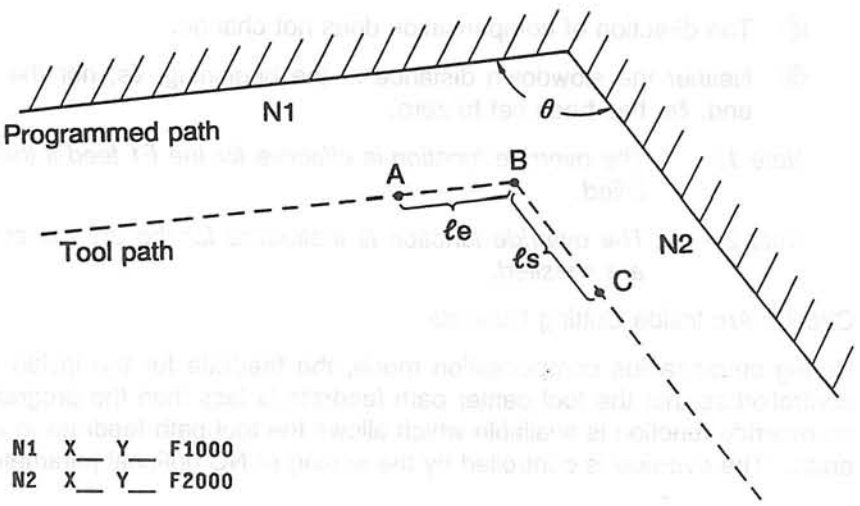
① Slowdown distance at end of corner
ℓe : NC optional parameter (long word) No. 5.

② Slowdown distance at beginning of corner
ℓe : NC optional parameter (long word) No. 6.

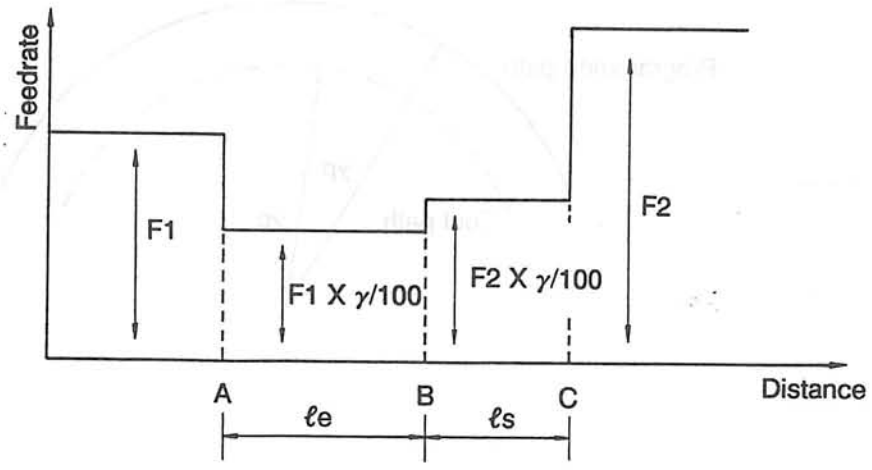
③ Slowdown rate:
r : NC optional parameter (word) No. 3.

④ Angle of inside cut
θ : NC optional parameter (word) No. 4.

A : Override ON position
B : Position at which positioning is made by N1 block commands
C : Override OFF position




```
N1  X__ Y__ F1000
N2  X__ Y__ F2000
```



(b) The following table lists the possible range of values along with the initially set values:

	ℓe	ℓs	r	θ
Setting Range	0 - 99999.999 mm 0 - 9999.9999 inch	0 - 99999.999 mm 0 - 9999.9999 inch	1 - 100%	1 - 179°
Initial Setting	0	0	100%	90°

(c) Requirements

The override function will be turned on if both blocks before and after a corner, including the corner, satisfy the following requirements.

① The system is in the cutter radius compensation mode.

② Either G01, G02, or G03 mode has been established.

③ The corner's inner angle is smaller than value, θ, set for the angle of inside cut.

④ No more than three blocks involving no axis movements have been executed in between.

⑤ No blocks contain a G41, G42, or G40 command.

⑥ No rotary axis commands have been given.

⑦ The slowdown rate, has not been set to 100 percent.

⑧ The direction of compensation does not change.

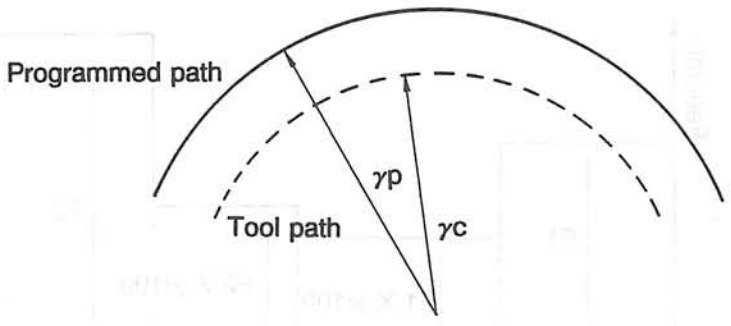
⑨ Neither the slowdown distance at the beginning, ℓs, nor the slowdown distance at the end, ℓe, has been set to zero.

Note 1: The override function is effective for the F1 feed if the above conditions are satisfied.

Note 2: The override function is ineffective for the dry run even if the above conditions are satisfied.

(2) Circular Arc Inside Cutting Override

During cutter radius compensation mode, the feedrate for the inside cutting of arcs is normally controlled so that the tool center path feedrate is less than the programmed feedrate. However, an override function is available which allows the tool path feedrate to equal the programmed feedrate. The override is controlled by the setting of NC optional parameter (word) No. 5.



(a) In the example above, there are two actual feedrates possible:

$\gamma c/\gamma p \times 100 \geqq$ parameter setting

Actual feedrate = Programmed feedrate x $\gamma c/\gamma p$

$\gamma c/\gamma p \times 100 <$ parameter setting

Actual feedrate = Programmed feedrate x Parameter setting

(b) The parameter, therefore, sets the lower limit of arc inside cutting. The allowable setting range is from 1 to 100 percent and the initial setting is 100 percent.

(c) The override function for arc inside cutting will be turned on if the system meets the following requirements:

① The system is in cutter radius compensation mode.

② The inside cutting of a circular arc is being executed.

The above two override functions are effective for F4 and F1 command during tool radius compensation.

(d) Overlapping Overrides

If there is any overlap between the automatic corner override, the arc inside cutting override, and the standard F4 command override, the following formula will be used to determine the actual feedrate:

Actual feedrate = specified speed x $\gamma c/\gamma p$ x FO/100 x CO/100

where

FO is the standard F command override and

CO is the automatic corner override.

2-5. Three-Dimensional Tool Offset (G43, G44) (Optional)

This function may be used to provide three-dimensional tool offset based on the axis move commands and the I, J, and K values which specify the tool offset direction.

(1) Three-dimensional Tool Offset Start-up

The three-dimensional offset mode will begin when the following command is executed during the cancel mode for tool offset:

G44 Xp__ Yp__ Zp__ I__ J__ K__ D__

(a) The three-dimensional space where the offset will be applied is determined by the axis addresses (Xp, Yp, Zp) given in this start-up block. If addresses are not provided, the default settings assumed are the X-, Y-, and Z-axes.

Example:

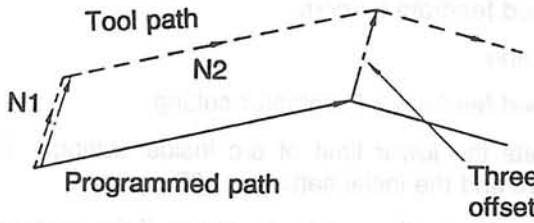
```
G44  X_ I_ J_ K_          X, Y, Z space
G44  U_ V_ Z_ I_ J_ K_    U, V, Z space
G44  W_ I_ J_ K_          X, Y, W space
```

Where Xp is the X-axis or its parallel axis, U

Yp is the Y-axis or its parallel axis, V

Zp is the Z-axis or its parallel axis, W

(b) If none of the X, Y, and Z axes have been specified, the axis moves by the tool offset amount, D.




```
N1  G44     I_ J_ K_
N2          X_ Y_ Z_
```

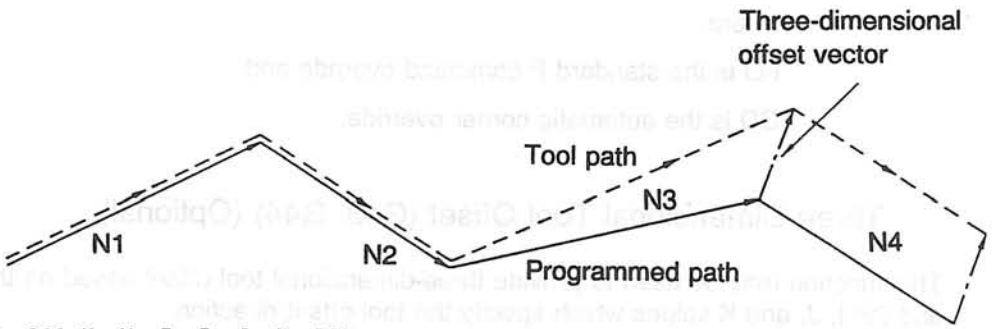
(c) During start-up, if none of the I, J, and K values have been specified, the corresponding direction is taken as zero.

(d) The tool offset number, or compensation amount, is specified by a D command, just as with cutter radius compensation.

(e) The standard D commands range from D00 to D50; D100, D200, and D300 are optionally available.

(f) If no D command is given, the previously specified D number will be used. If there is no previously specified D command, D00 will be assumed.

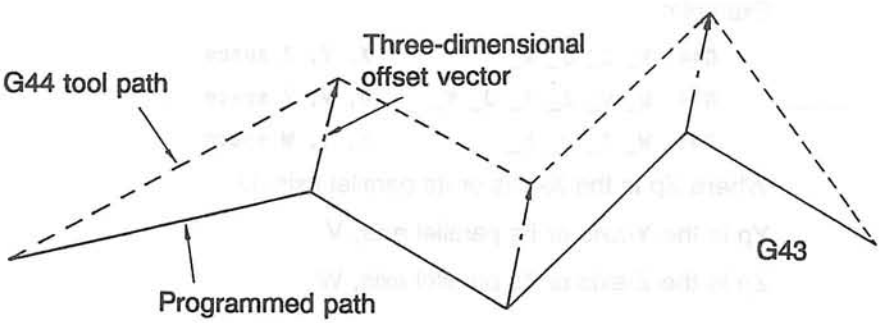
(g) Even with a tool offset amount of zero (D00), the three-dimensional offset mode will be started, but no compensation will take place, until a new D number is specified.




```
N1  G44 X_ Y_ Z_ I_ J_ K_ D00
N2      X_ Y_ Z_
N3      X_ Y_ Z_ D01
N4      X_ Y_ Z_
```

(2) Three-dimensional Tool Offset Vector

During the three-dimensional tool offset mode, a three-dimensional offset vector is formed at the end of each block, as shown in the figure below:



(a) Vector expressions

The vectors are expressed by the following equations:

Xp-axis vector component: $Vx = i*\gamma/p$

Yp-axis vector component: $Vy = j*\gamma/p$

Zp-axis vector component: $Vz = k*\gamma/p$

i, j, k are the values specified by the addresses I, J, and K; γ is the tool offset amount specified by the D command; and p is the value set by NC optional parameter (long word) No. 7.

When "0" is set at this parameter, value P is obtained by the following formula.

$$p = \sqrt{i^2 + j^2 + k^2}$$

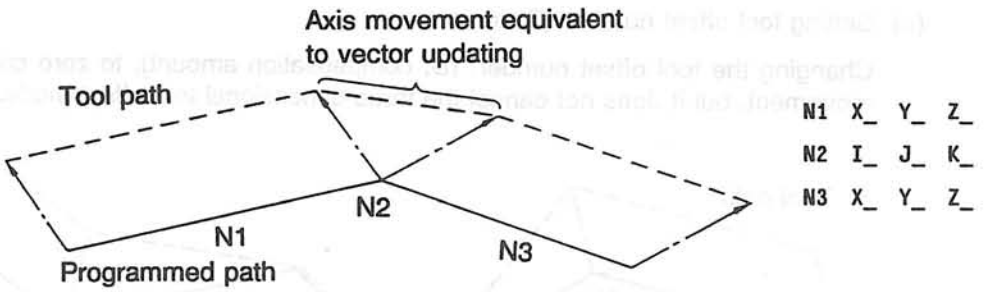
The initial value of parameter "p", the NC optional parameter (long word) No. 7, is zero. The possible setting range is given in the following table.

Metric System	Inch System
0 - ±99999.999 mm	0 - ±9999.9999 inch

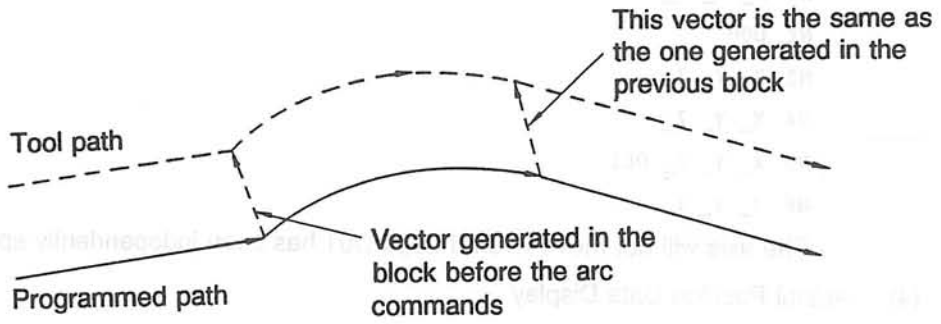
(b) In a block where no I, J, and K commands have been specified, the resulting vector is the same as the one formed in the previous block.

(c) In a block where one of the I, J, and K commands have not been specified, the component for that direction of the resulting vector is zero.

(d) If no axis commands, Xp, Yp, and Zp, but only I, J, and K commands have been specified, then only the vector is updated and the axis moves by the change amount of the vector.



(e) The I, J, and K commands specified in a G02 or G03 block (arc or helical cutting) are used to indicate the coordinates of the center of that arc. The resulting vector is the same as the one generated in the previous block.



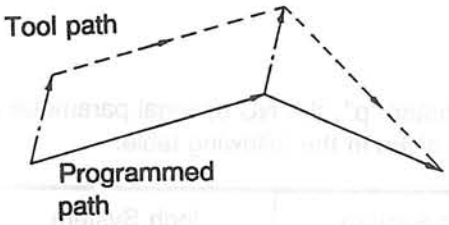
(f) Once a G44 block has been given which specifies the three-dimensional space, further commands which specify an additional, parallel axis will be ignored. No alarm will be activated.

```
N1  G44     W_ I_ J_ K_  ..   Specifies XYW space
N2          X_ Y_
N3          Z_  .........    No offset for the Z-axis possible (Command is ignored.)
```

(3) Canceling Three-dimensional Tool Offset

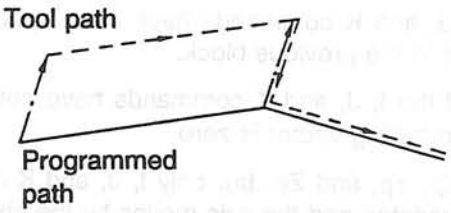
The G43 command is used to cancel the three-dimensional tool offset mode.

(a) Canceling in a block with an axis command




```
N1          X_ Y_ Z_
N2  G43     X_ Y_ Z_
```

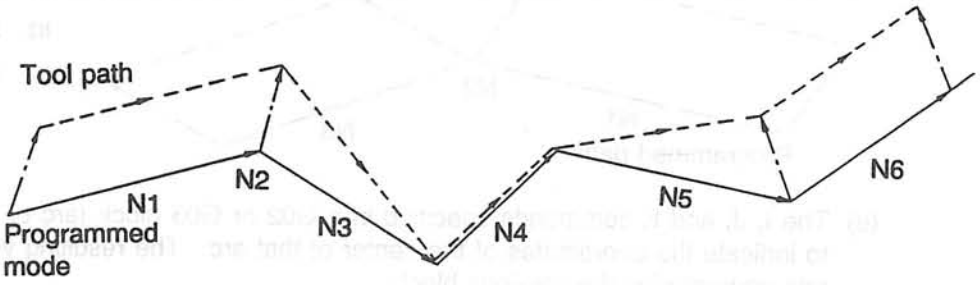
(b) Canceling in a block without other commands




```
N1  X_ Y_ Z_
N2  G43
N3  X_ Y_ Z_
```

(c) Setting tool offset number (D) to zero

Changing the tool offset number, (D, compensation amount), to zero cancels the offset axis movement, but it does not cancel the three-dimensional tool offset mode.




```
N1  X_ Y_ Z_
N2  D00
N3  X_ Y_ Z_
N4  X_ Y_ Z_
N5  X_ Y_ Z_ D01
N6  X_ Y_ Z_
```

The axis will not move even though D01 has been independently specified in block N5

(4) Actual Position Data Display

The coordinate values on the position display page indicate the actual tool path with offset.

(5) Feedrate

The feedrate is controlled so that the axes move along the offset tool path at a programmed tool path.

(6) Relationship to Other Commands

(a) G code commands

The following G codes may not be specified during three-dimensional tool offset mode:

G15, G16, G40, G41, G42, G92,
G codes for area machining
G codes for coordinate system parallel shift/rotation
G codes calling a fixed cycle

(b) Fixed cycle mode

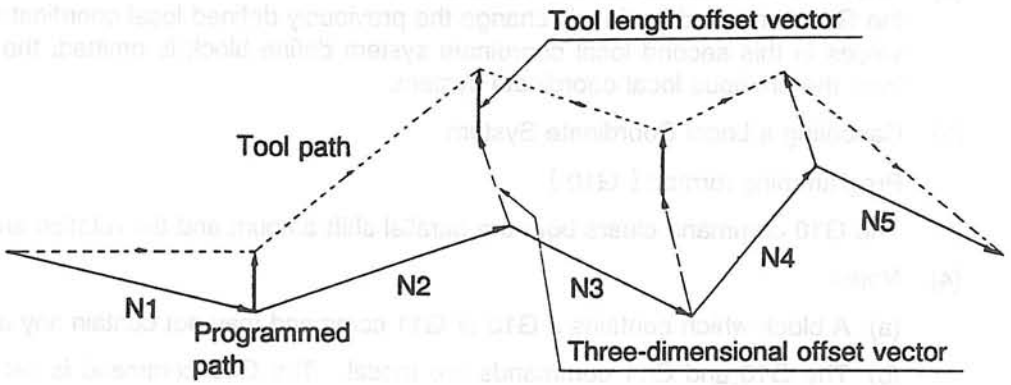
The three-dimensional tool offset command may not be issued during the fixed cycle mode.

(7) Relationship to Other Tool Offset Functions

(a) Tool length offset (G53 - G56)

The three-dimensional tool offset mode may be executed during tool length offset mode; and the tool length offset mode may be executed during three-dimensional tool offset mode. Tool length offsetting may be carried out at the same time as three-dimensional offsetting.

Example: Three-dimensional tool offset command is issued during tool length offset mode.




```
N1 G56  X_ Y_ Z_ H01 ..... Tool length offset mode
N2 G44  X_ Y_ Z_ I_ J_ K_ D02 ..... Three-dimensional offset mode
N3      X_ Y_ Z_
N4 G53  X_ Y_ Z_ ..... Cancel tool length offset
N5 G43  X_ Y_ Z_ ..... Cancel three-dimensional offset
```

(b) Cutter radius compensation (G40 - G42)

An alarm will be activated if a cutter radius compensation command (G41 or G42) is specified during the three-dimensional tool offset mode, or if a three-dimensional tool offset command (G44) is issued during cutter radius compensation mode. Therefore, these two offset operations may not be performed simultaneously.

3. Coordinate System Conversion Functions (Optional)

3-1. Parallel Shift and Rotation of Coordinate Systems (G11, G10)

A new coordinate system which may be established by a parallel shift or rotation of a work coordinate system is referred to as a local coordinate system. This section describes the commands which may be used to change the coordinate system.

(1) Specifying a Local Coordinate System

Programming format: 【 G11 IP__ P__ 】

IP : Parallel shift amount to establish a local coordinate system

The shift amount should be given as an absolute value, regardless of the selected dimensioning mode: G90 (absolute), G91 (incremental), or G62 (mirror image).

P : Rotation amount to establish a local coordinate system

A numerical value should be given in units of 1 degree, 0.001 degree, or 0.0001 degree in accordance with the selected unit system. The rotation angle should be given as an absolute value, regardless of the selected dimensioning mode.
The rotation is performed in the plane which has been selected (G17, G18, or G19) when G11 is specified, and does not affect the axes located outside the plane. The rotation direction is counter-clockwise viewed from the forward direction of the axis outside the selected plane.
A value of P0 or the omission of a P value will result in a parallel shift without rotation.

(2) Once the G11 command has been given, the local coordinate system is defined. Programming the G11 command again will change the previously defined local coordinate system. If one of the values in this second local coordinate system define block is omitted, the default value is taken from the previous local coordinate system.

(3) Canceling a Local Coordinate System

Programming format: 【 G10 】

The G10 command clears both the parallel shift amount and the rotation amount.

(4) Notes

(a) A block which contains a G10 or G11 command may not contain any other G codes.

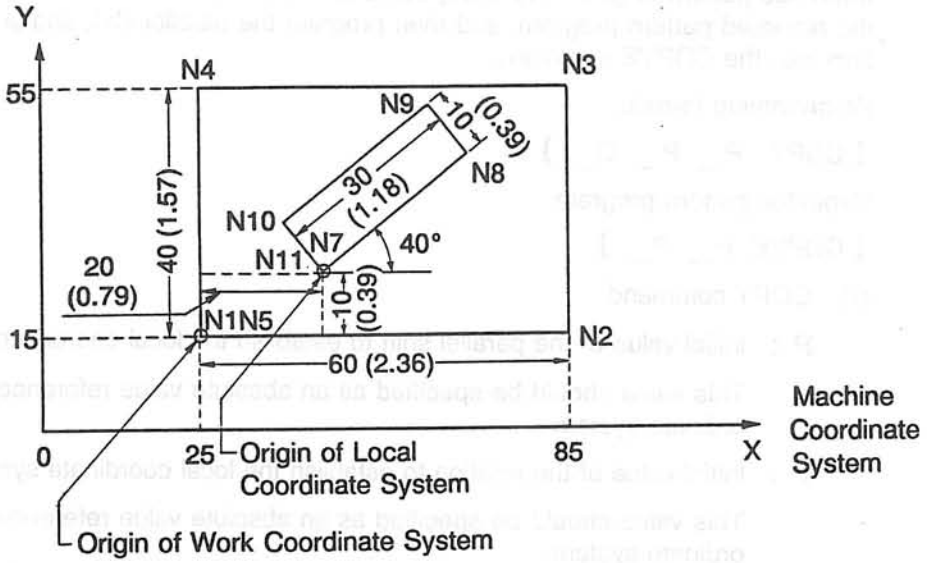
(b) The G10 and G11 commands are modal. The G10 command is set when power is turned on and when the NC is reset.

(c) G11 may not be specified during the following modes:

- Workpiece geometry enlargement/reduction mode
- Machine coordinate system mode (H00)
- Copy function mode

Example:

A local coordinate system may be used effectively to machine a part as shown in the example below.




```
N1   G15 G90 G00 X0 Y0 H01 ..... Selection of work coordinate system 1
N2   G01 X60 F100
N3   Y40
N4   X0
N5   Y0
N6   G17 G11 X20 Y10 P40 ..... Setting of local coordinate system
N7   X0  Y0
N8   X30
N9   Y10
N10  X0
N11  Y0
N12  G10 ..... Cancel of local coordinate system
```

The zero offset values of work coordinate system 1 are:

x = 25, y = 15

3-2. Copy Functions (COPY, COPYE) (Optional)

The copy functions may be used to simplify the cutting programming of a workpiece which consists of the same pattern repeated with a parallel shift or rotation. First, set a local coordinate system for the reference pattern or geometry using the COPY command instead of the G11 command. Next, specify the repeated pattern program, and then program the parallel shift and/or rotary movements of the pattern with the COPYE command.

Programming format:

【 COPY ℙ_ P_ Q_ 】

Repeated pattern program

【 COPYE ℙ_ P_ 】

(1) COPY command

ℙ : Initial value of the parallel shift to establish the local coordinate system

This value should be specified as an absolute value referenced to the origin of the work coordinate system.

P : Initial value of the rotation to establish the local coordinate system

This value should be specified as an absolute value referenced to the origin of the work coordinate system.
The numerical value should be given in units of 1 degree, 0.001 degree, or 0.0001 degree in accordance with the selected unit system.
If values for P or P are omitted, the default values are taken as the most recently entered values.

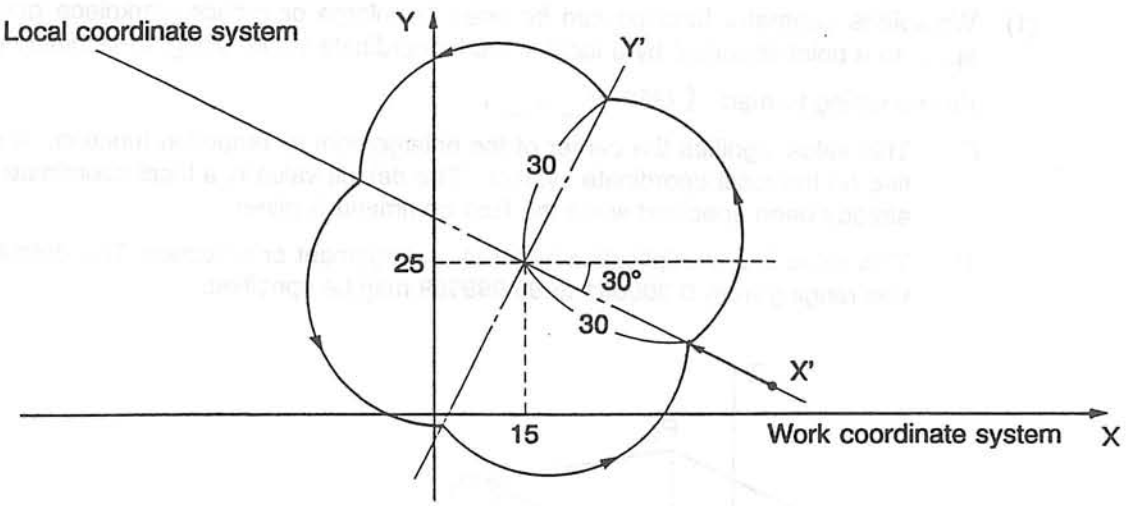
Q : The Q value represents the number of times the pattern will be repeated. The default value is 1. The programmable range is from 1 to 9999.

(2) COPYE command:

ℙ : Incremental value for parallel shift of local coordinate systems. The default value is 0.

P : Incremental value for rotation of local coordinate systems. The default value is 0.

Example




```
G11 X15 Y25 P-30
G01 X30 F100 M03
COPY    Q4
G01 X30 Y0 .........  (Positioning point agrees with point (A) in the second and later positioning)
G03 X0 Y30 I-20 J10
G01 X0 Y30 .........  (A)
COPYE   P80
```

No circular interpolation commands may be contained in the blocks in between the COPY and COPYE command blocks.

(3) Notes

(a) Both the G11 and COPY commands may be specified after the G11 command has already been defined. However, neither the G11 nor the COPY command may be given after the COPY command has already been specified.

(b) The following restrictions apply to the COPY command when it is used in a main program for the B-method (large capacity tape operation mode): No IF or GOTO statements may be used; The tape length of the program from the COPY command to the COPYE command may not be more than 10 m (33 ft).

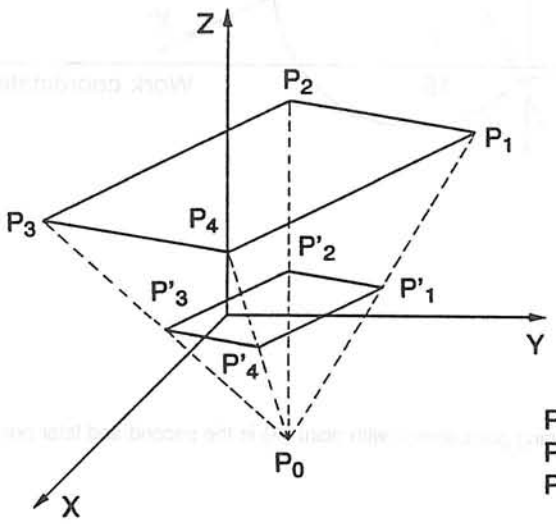
4. Workpiece Geometry Enlargement/Reduction Functions (G51, G50) (Optional)

(1) Workpiece geometry function can be used to enlarge or reduce workpiece geometries with respect to a point specified by a local or work coordinate value, using a machining program.

Programming format: 【 G51 ℙ_ P_ 】

ℙ : This value signifies the center of the enlargement or reduction function. It should be specified on the local coordinate system. The default value is a local coordinate value which has already been specified when the G51 command is given.

P : This value is a multiplication factor for enlargement or reduction. The default value is 1. Values ranging from 0.000001 to 99.999999 may be specified.



P_0 : Center of enlargement/reduction
P_1 - P_4 : Programmed geometry
P'_1 - P'_4 : Geometry after reduction

(2) Cancel function

The following command can be used to cancel the workpiece geometry function mode.

Programming format: 【 G50 】

(3) *Notes:*

(a) The enlargement/reduction function may be used on individual axes by setting NC optional parameter (bit) No. 6. However, an alarm will be activated if the parameter setting is different from the selected plane axis specified in the circular interpolation command.

(b) The enlargement/reduction function does not affect the following:

① Local coordinate system setting values (G11)

② Cutter radius compensation values and three-dimensional offset values (G41, G42, G43)

③ Tool length offset values (G56 through G59)

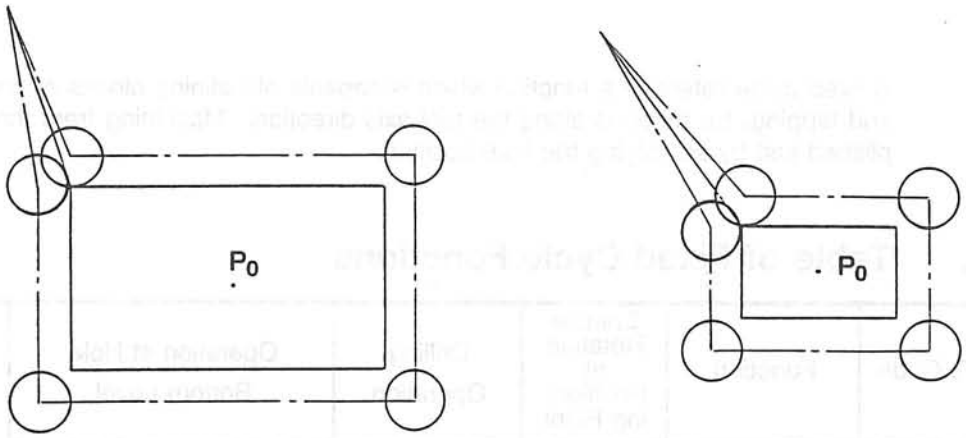
④ Work coordinate system setting values (G92)

⑤ Pecking feed and retraction amount during deep hole drilling fixed cycle along Z-axis (G73, G83)

⑥ X- and Y-shift amount during fine boring and back boring fixed cycle along Z-axis (G76, G87)

Example 1:

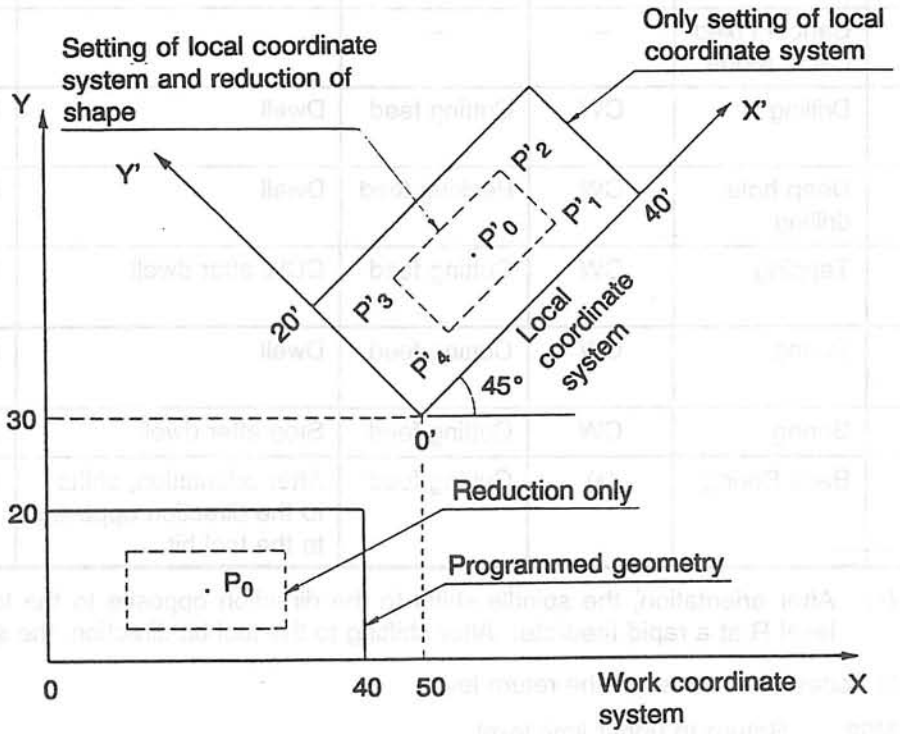
Cutter radius compensation and enlargement and reduction of workpiece geometry



Example 2:

Setting the local coordinate system and enlargement and reduction of workpiece geometry

```
N1 G17  G11 X50 Y30 P45      Setting of local coordinate system
N2 G90  G51 X20 Y10 P0.5     Reduction of geometry
N3 G01  X40                  Positioning at P'1
N4      Y20                  Positioning at P'2
N5      X0                   Positioning at P'3
N6      Y0                   Positioning at P'4
```



SECTION 6 FIXED CYCLES

A fixed cycle refers to a function which is capable of defining blocks of commands, like drilling, boring and tapping, for motions along the tool axis direction. Machining from the second on may be accomplished just by specifying the hole position.

1. Table of Fixed Cycle Functions

G Code	Function	Spindle Rotation at Positioning Point	Drilling Operation	Operation at Hole Bottom Level	Retraction Operation	Spindle Rotation at Return Level
G71	Return Level Command	–	–	–	–	–
G73	High Speed Deep Hole Drilling	CW	Pecking feed	Dwell	Rapid feed	CW
G74	Reverse Tapping	CCW	Cutting feed	CW, Spindle rotation after dwell	Cutting feed	CCW after dwell
G76	Fine Boring	CW	Cutting feed	After dwell, the spindle stops at a specified position and shifts to the direction opposite to the tool bit.	Rapid feed	CW after shifting to tool bit direction
G80	Cancel Fixed Cycle Mode	–	–	–	–	–
G81 G82	Drilling	CW	Cutting feed	Dwell	Rapid feed	CW
G83	Deep hole drilling	CW	Pecking feed	Dwell	Rapid feed	CW
G84	Tapping	CW	Cutting feed	CCW after dwell	Cutting feed	CW after dwell
G85 G89	Boring	CW	Cutting feed	Dwell	Cutting feed	CW
G86	Boring	CW	Cutting feed	Stop after dwell	Rapid feed	CW
G87	Back Boring	(*)	Cutting feed	After orientation, shifts to the direction opposite to the tool bit	Rapid feed	CW after shift to tool bit direction

(*) After orientation, the spindle shifts to the direction opposite to the tool bit and advances to the level R at a rapid feedrate. After shifting to the tool bit direction, the spindle rotates clockwise.

M codes used to select the return level:

M52 ..... Return to upper limit level

M53 ..... Return to a specified point level set by G71

M54 ..... Return to the point R level

2. Fixed Cycle Operations

All fixed cycle functions are composed of the following six operations:

Operation 1 Positioning to Hole Machining Position (Rapid Feedrate)

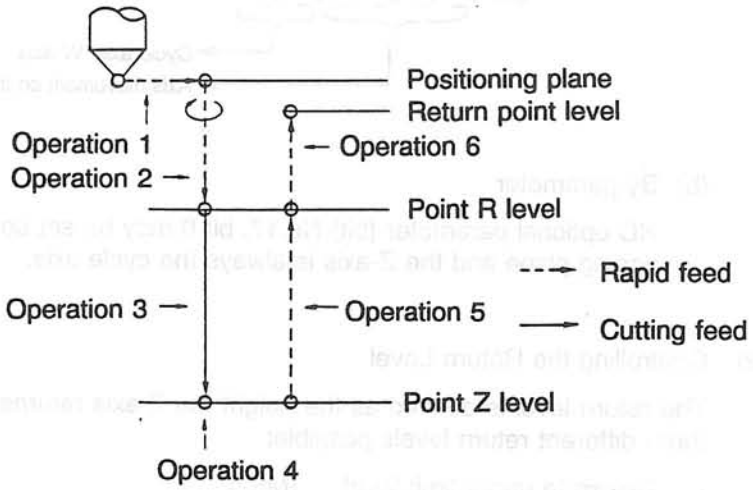
Operation 2 Rapid Feed to Point R Level

Operation 3 Hole Machining

Operation 4 Operation at Hole Bottom Position (Point Z Level)

Operation 5 Retraction to Point R Level

Operation 6 Rapid Feed to the Upper End Level or a Specified Point Level



Operation 1 is referred to as the operation in the positioning plane. Operations 2 through 6 are referred to as the cycle axis operation.
A fixed cycle may also be executed on a rotary axis after the positioning axis has been specified.

(1) Determining Positioning Plane and Cycle Axis

(a) By command

The positioning plane may be determined by selecting a plane using the G17, G18, and G19 commands. The cycle axis is then chosen as the axis which is vertical or parallel to the selected positioning plane.

G Code	Positioning Plane	Cycle Axis
G17	Xp-Yp plane	Zp
G18	Zp-Xp plane	Yp
G19	Yp-Zp plane	Xp

Xp = X- or U-axis

Yp = Y- or V-axis

Zp = Z- or W-axis

After the positioning plane has been selected, two different cycle axes are possible. The cycle axis is determined by an axis address (not on the positioning plane) included in the block containing the fixed cycle mode G code (G73 - G89).

Wrong: `G17 X_ Y_`
`G81 X_ Y_ Z_ W_ R_ F_`

The G81 block contains both the Z- and W-axis and, therefore, the cycle axis cannot be determined. An alarm is activated.

Correct: `G17 X_ Y_`
`G81 X_ Y_ U_ V_ W_ R_`

Cycle axis: W-axis
Axis movement on the positioning plane

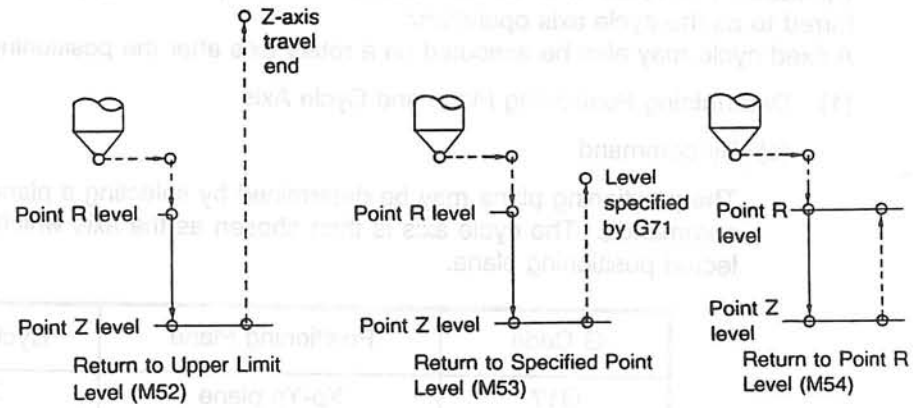
(b) By parameter

NC optional parameter (bit) No.17, bit 0 may be set so that the G17 plane is always the positioning plane and the Z-axis is always the cycle axis.

(2) Controlling the Return Level

The return level is defined as the height the Z-axis returns to at the end of one cycle. There are three different return levels possible:

- Return to upper limit level (M52)
- Return to specified point level (M53)
- Return to point R level (M54)



(a) Notes:

1) The M52 command is possible when the Z-axis has been chosen as the cycle axis and when the machining operation is performed in the negative direction of the Z-axis. During M52 mode. The Z-axis returns to a position located 0.1 mm (0.004 in.) before the travel end of the Z-axis.

2) Before the M53 command may be specified, a return level must be defined, in advance, by the G71 command.

3) During the back boring mode (G87), a return to the point R level (M54 mode) is not possible. If M54 is selected, the M53 mode will be used.

4) The M52 command is a one-shot command, effective only in a programmed block, which takes priority over the M53 and M54 commands. Since both the M53 and M54 commands are modal, one of them is always effective. After power has been turned on, or the NC has been reset, the M54 mode is set.

(3) Fixed Cycle Mode

(a) A fixed cycle mode is specified by a hole machining definition command, (G73 through G76, and G81 through G89). The fixed cycle mode may be canceled using one of the following G codes:
G00, G01, G02, G03 or G80.

(b) If a second hole machining definition command is given during the fixed cycle mode, the hole machining data will be updated accordingly and the fixed cycle mode will remain without being canceled.

(c) When the fixed cycle mode is canceled using the G80 code, the interpolation mode, (G00, G01, G02, G03, or G60) selected before the start of the fixed cycle mode, is recovered and the M05 code is generated.

Example:

```
G17
G01  X_  Y_
G81  X_  Y_  Z_  R_  F_
     X_                          } Fixed cycle mode
     X_  Y_
G80  X_  Y_ ..........  Same as G01 X_  Y_
```

Axis movement commands, which are programmed in a block containing the G80 code, will be executed only after the fixed cycle mode has been canceled.

(4) Cycle Operation Conditions

(a) Cycle axis operation possible

During the fixed cycle mode, the cycle axis operates in the following blocks:

① Hole machining definition command blocks (which define the operation of the cycle axis) If the hole position commands are omitted, the position where the axes are presently located is regarded as the hole position where the cycle axis will operate.

② Blocks in between the hole machining definition command block and the G80 block. The blocks must include hole position data for one or more axes.

Example: Fixed cycle mode

```
G17  G00  X_  Y_          ..... 1)
G81  Z_  R_  F_           ..... 2)
        (empty block)
F_                        ..... 3)
M_
G04  F_
X_                        ..... 4)
G80  X_  Y_               ..... 5)
```

1) Positioning

2) No hole position data is provided here, so drilling is carried out at the present position, obtained from the positioning step 1).

3) No hole machining is performed, since no hole position data is provided.

4) Hole machining is carried out at this point, since hole position data is provided.

5) No drilling is performed at this block, since the fixed cycle mode is canceled and no new fixed cycle is called.

(b) Cycle axis operation not possible

Cycle axis operation will be prevented during the following modes, even if the above conditions have been met. The following modes are used together with a coordinate computation function.

① NCYL (NO CYCLE)

If the NCYL mode is specified during fixed cycle mode, positioning to the defined hole position is performed, but the cycle axis does not operate.

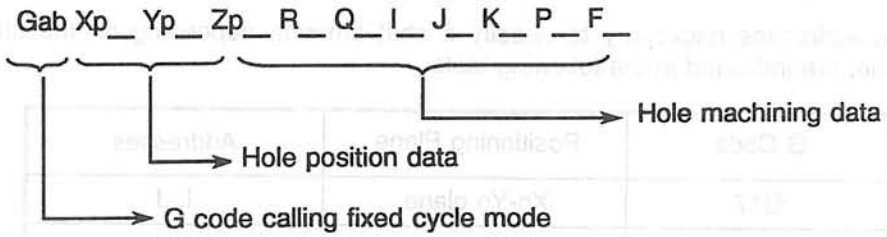
② NOEX (NO EXECUTE)

If the NOEX mode is specified during fixed cycle mode, no axis movements may be performed.

3. General Rules for Programming Fixed Cycles

This section describes the general rules of programming hole machining data which is specified in blocks containing the codes G73 through G76 and G81 through G89. It is assumed in the following explanations that the positioning plane is the Xp-Yp plane and the cycle axis is the Zp-axis.

3-1. Programming Format



(1) G code

Refer to Section 6-1 "Table of Fixed Cycle Functions" for a list of the fixed cycle functions and their G codes. All of the G codes used to call fixed cycles are modal and, once programmed, they remain effective until another fixed cycle G code or fixed cycle canceling G code is programmed.

(2) Hole position data

The hole position data specifies the position at which the fixed cycle (hole machining) will be performed.
The addresses used for specifying the position are determined by the plane selection G code (G17, G18 and G19) which defines the positioning plane.
The hole position data may be given in the absolute (G90) or incremental mode (G91) as needed.
Positioning is performed in the specified mode. Normally, the G00 mode, rapid feedrate, is programmed. However, the G60, unidirectional positioning, is also possible.

(3) Hole machining data

(a) Once programmed, the hole machining data remains active until it is changed or the fixed cycle mode is canceled.
The individual addresses used to specify the hole machining data is described below.

(b) Zp: Specifies the hole bottom position.

During the absolute programming mode (G90), the hole bottom position should be specified as an absolute value. During the incremental programming mode (G91), the distance between the point R level and the hole bottom should be specified.

(c) R: Specifies the point R level.

During the absolute programming mode (G90), the point R level should be specified as an absolute value. During the incremental programming mode (G91), the distance from the tool position where the fixed cycle mode begins to the point R level should be programmed.

(d) Q: ① G73, G83 mode

Specifies, in a positive, incremental value, the infeed amount during the first infeed motion.

② G76, G87 mode

Specifies the shift amount in a positive, incremental amount.

③ G74, G84 mode

Specifies the dwell period at the point R level.
The unit system for the dwell time, specified with NC optional parameter (bit) No. 3, may be selected from four possibilities: 1, 0.1, 0.01 and 0.001 seconds.

Note 1: If a negative value is entered in ① ②, the minus sign will be ignored.

Note 2: If a Q value is not entered, the default value will be the last, most recently entered Q value, if one exists.

(e) Specifies, in an incremental amount, the shift amount during modes G76 and G87.

The addresses necessary to specify a shift amount, depending on the chosen positioning plane, are indicated in the following table.

G Code	Positioning Plane	Addresses
G17	Xp-Yp plane	I, J
G18	Zp-Xp plane	K, I
G19	Yp-Zp plane	J, K

(f) P: Specifies the dwell time period.

The unit system for the dwell time, specified with NC optional parameter (bit) No. 3, may be selected from four possibilities: 1, 0.1, 0.01 and 0.001 seconds. The maximum possible dwell time is 99999.999 seconds.

(g) F: Specifies the feedrate during cycle axis operation.

The programmed feedrate is effective on all interpolation commands and remains effective even after the fixed cycle mode has been canceled.
The feedrate can only be changed by entering a new feedrate value.

(h) Position check for the actual position, return point, point R level, and point Z level is conducted as indicated below.

1) In the M52 and M53 modes, the infeed direction is judged from point R and point Z levels, and whether or not the retracting direction is reverse to the infeed direction is checked. (The relation to the actual position is not checked.)
2) In the M54 mode, position check is not conducted.

3-2. Command Items Necessary for Fixed Cycle Function Commands

The table below was set up using the XY plane as the positioning plane, and the Z-axis as the cycle axis.

FUNCTION / Command Item / ADDRESS		Hole Position		Point Z Level	Point R Level	Pecking Amount	Shift Amount		Dwell Time		Feed-rate	Retrac-tion Amount
		ABS/INC		ABS /INC	ABS /INC	INC	INC		-		-	INC
		X	Y	Z	R	Q	Q	I,J,K	P	Q	F	Para-meter
G73	High speed, deep hole drilling	○	○	◎	◎	△			△		◎	d_1
G74	Reverse tapping	○	○	◎	◎				△	△	◎	
G76	Fine boring	○	○	◎	◎		△	△	△		◎	
G81 G82	Drilling	○	○	◎	◎				△		◎	
G83	Deep hole drilling	○	○	◎	◎	◎			△		◎	d_2
G84	Tapping	○	○	◎	◎				△	△	◎	
G85 G89	Reamer	○	○	◎	◎				△		◎	
G86	Boring	○	○	◎	◎				△		◎	
G87	Back boring	○	○	◎	◎		△	△	△		◎	

(1) Notes

(a) "ABS/INC" indicates that either the absolute (G90) or the incremental command (G91) may be selected for specifying the command item.
INC: Indicates that the command item should be specified as an incremental (G91) amount, regardless of the established dimensioning mode.

(b) "○" indicates that the command item may be omitted from the program block. The most recently entered values will be used as the default values.

(c) "◎" indicates that the command item may be omitted from the fixed cycle program block, but must be specified at some point before cycle axis operation.

(d) "△" indicates that the command item may be omitted from the fixed cycle program block. However, if a command item has been specified during a previous fixed cycle command, the previously specified value will remain in effect.

(e) The shift amount should be specified using Q when the cycle axis has already been fixed at the Z-axis, otherwise, it should be specified using I, J, or K.

(f) Data which is not designated for a shift amount value in G76 and G87, will be saved and activated in other cycle functions.

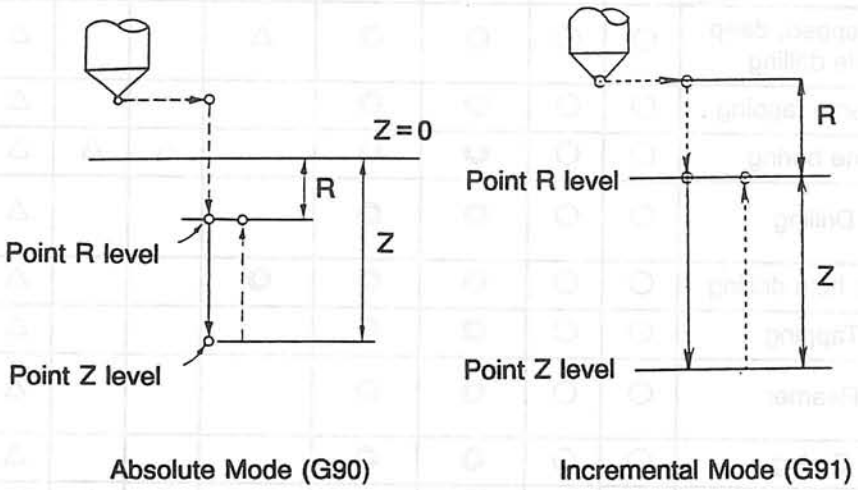
(g) All data set using parameters, except the retraction amount (d_1 and d_2) and feedrate (F), are effective only during the fixed cycle mode.
The retraction amount values,d_1 and d_2 , are deleted when the fixed cycle mode is canceled or the NC is reset. The feedrate, F, is canceled when the NC is reset.

(h) The values for P, Q, I, J, and K must be specified in a block where a cycle axis operation is performed. If one of these values is specified in a block without a cycle axis operation, no modal data will be established.

(i) The shift amount must be specified for the G76 and G87 codes, otherwise an alarm will be activated.

3-3. Absolute Mode and Incremental Mode

(1) Point R and point Z levels must be specified as shown below, depending on the selected mode: absolute (G90) or incremental (G91). The hole machining data should also be programmed in accordance with the mode selected when it is defined.

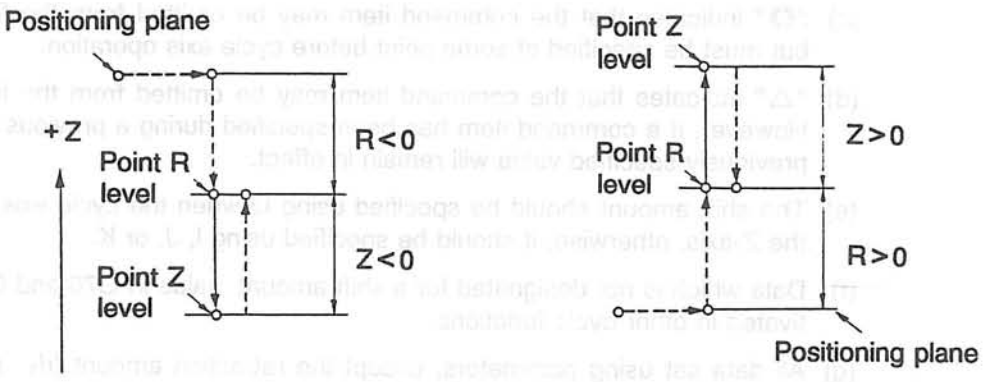


Absolute Mode (G90) Incremental Mode (G91)

(2) Incremental mode

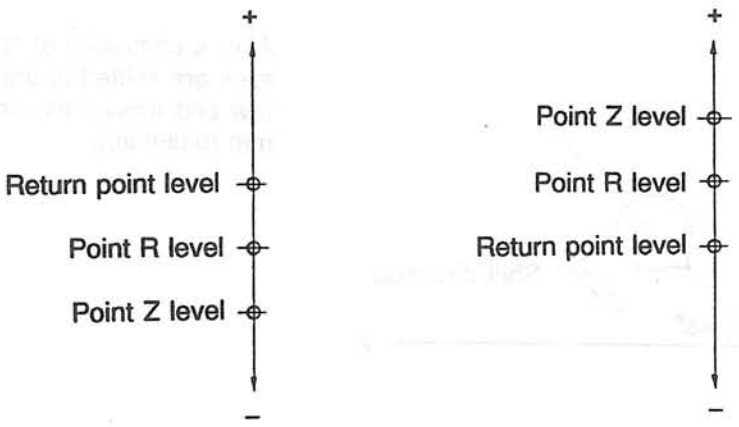
During incremental mode programming, place and minus signs placed before R and Z values are acknowledged. Generally, the cycle axis is first fed in the negative direction (infeed) and then in the positive direction (retraction). However, the cycle axis may be moved in the opposite direction to the positioning plane. During incremental mode programming, the direction of cycle axis movement can be specified by assigning the proper sign to the R and Z values. Refer to the example below.

Example: Cycle axis movement direction during incremental mode programming



3-4. Positional Relationship among Return Point Level, Point R Level and Point Z Level

The positional relationship among the three levels along the cycle axis direction must be one of the two cases shown below. (The only exception is G87 back boring, where the point R and point Z levels are switched.)



3-5. Axis Shift

During the G76 (fine boring) and G87 (back boring) modes, an axis shift is made at a rapid feedrate. The shift amount and direction may be set using one of the following two methods.

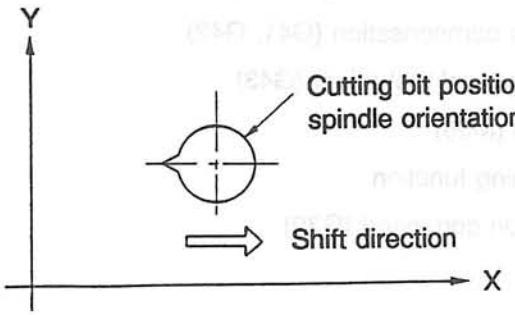
(1) Q command

The shift amount and direction may be set using a Q command only when the Z-axis has been chosen as the cycle axis by setting NC optional parameter (bit) No.17, bit 0.
The value of Q is a positive, incremental value.
Q commands are given priority over I, J, and K commands, explained in the next section. Therefore, if a Q command is specified, all I, J, and K commands will be ignored.
The shifting direction should be selected in advance by setting NC optional parameter (bit) No. 17, bits 1 and 2. Refer to the following table for the necessary bit settings.

Shift Direction	Bit 2	Bit 1
+X	0	0
– X	1	0
+Y	0	1
– Y	1	1

The shifting direction is defined in the machine coordinate direction and is opposite to the direction of the bit during spindle operation.

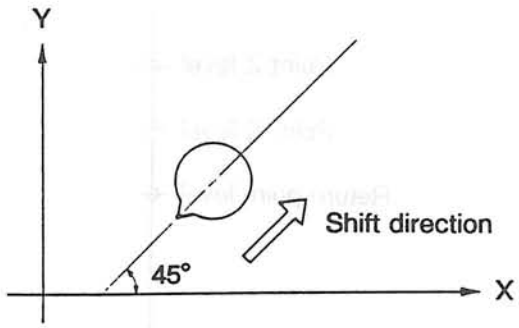
Example: Shift direction +X has been specified by parameter settings.



(2) I, J, and K commands

I, J, and K commands are sued when the cycle axis has not been fixed. The shift amount and direction of the tool axis may be specified using (I, J), (K, I), or (J, K) depending on the selected positioning plane. The shift direction is defined in the machine coordinate system.If no value is given for I, J, or K in the combinations of (I, J), (J, K), or (K, I), it is assumed to be zero.

Example:



After a command of "I1.0 J1.0" has been given, the axes are shifted in the direction indicated by the arrow and moved the amount of $\sqrt{1^2 + 1^2} = 1.414$ mm (0.056 in.).

3-6. Relationships between Fixed Cycle Functions and Other Functions

(1) Axis movement call mode (MODIN, MODOUT)

If the fixed cycle mode and the axis movement call mode overlap, the command MODIN will call the axis movement after the cycle axis operation has been completed.

(2) Unidirectional positioning

During the unidirectional positioning mode (G60), the cycle axis is as follows:

(a) Unidirectional positioning is performed in the positioning plane

(b) Two-directional positioning is performed along the cycle axis direction.
The G60 command may be specified during the fixed cycle mode. The fixed cycle mode is thus different from other interpolation modes (G00 through G03) because it is not canceled when the G60 command is given. When the fixed cycle mode is canceled using the G80 command, the G60 mode is canceled simultaneously.

(3) Geometry enlargement/reduction function

If the fixed cycle mode is specified during the geometry enlargement/reduction mode, the following factors will not be subjected to the enlargement/reduction processing.

(a) Pecking amount (Q) and retraction amounts (d1 and d2) in the G73 and G83 modes

(b) Shift amount (Q, or I, J, and K) in the G76 and G87 modes

(4) The following items cannot be specified during the fixed cycle mode:

(a) Plane selection which changes the positioning plane

(b) Cutter radius compensation (G41, G42)

(c) Three-dimensional tool offset (G43)

(d) Tool change (M06)

(e) Area machining function

(f) Home position command (G30)

(5) Coordinate calculation function

If a coordinate calculation function command with an M52 code is specified during the fixed cycle mode, the Z-axis will be returned to the upper limit level.

Example:

```
G81  X_ Y_ Z_ R_ F_
LAA  X_ Y_ I_ K_ I_ K_ J_ M52
```

The Z-axis is returned to the upper limit level after the final point has been calculated.

3-7. Notes for Programming a Fixed Cycle

(1) Using the dwell command with G74, G84, and G86 commands

There are three fixed cycle modes which control the spindle rotation: G74, G84, and G86. During one of these modes, if a hole machining operation, (such as drilling), is to be consecutively carried out with only a short distance between hole positions and between the point R level and the specified level, the spindle may fail to reach the programmed speed before starting the drilling operation. Therefore, it is necessary to program a dwell command (G04) in between the drilling operations.

(2) Canceling with commands G00 through G03

The fixed cycle mode may also be canceled using a G00 through G03 code. The fixed cycle will be canceled as soon as the code is read. Therefore, the code should be placed after the fixed cycle command in a block.

Example:

```
G01  G73  X_ Y_ Z_ R_ P_ Q_ F_
```

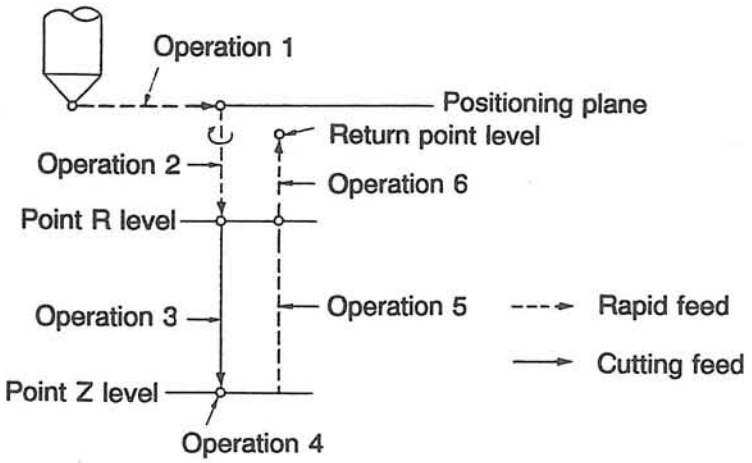
In this example, the fixed cycle function, G73, will not be canceled since the G01 code is placed before it.

(3) NC reset

The fixed cycle mode and the programmed hole machining data are all cleared when the NC is reset.

(4) Single block mode operation

During single block mode operation, the cycle motion will stop at the end of operations 1 and 5, when code M54 has been programmed, or at the end of operation 6, when code M52 or M53 has been programmed, and at the completion of the cycle.



It is also possible to stop the cycle after the completion of operation 2 in the illustration above, by setting bit 0 of NC optional parameter (bit) No. 34. The following points then apply:

(a) The point R level at which the Z-axis stops during single block mode operation is the first point R level it reaches after the fixed cycle mode has started. However, during the G83 deep hole drilling cycle operation, the Z-axis does not stop at the point R level until the point Z level has been reached. At this point, the Z-axis retracts to the point R level and stops there.

(b) During the G87 back boring cycle, the first point R level reached is used as the starting point for the hole bottom machining. A single block mode stop is possible at this point R level.

(c) If the positioning point and the point R level are at the same point, a single block mode stop will occur twice at the same point, since the stop occurs at the positioning point and the point R level.

(5) Slide hold function

If the slide hold function is activated during a tapping cycle (G74 or G84), cycle motion does not stop until after operation 5 has been completed, even though the SLIDE HOLD lamp came on earlier.
However, the slide hold function will be activated immediately when the **SLIDE HOLD** pushbutton is pressed during operations 1, 2, and 6.

(6) Override settings with the G74 and G84 commands

During G74 and G84 tapping cycle modes, both the cutting feedrate override and the spindle speed override are fixed at 100% and cannot be changed. The rapid traverse rate override, however, may be changed.

4. Details of Fixed Cycles

In the following explanations, it has been assumed that the XY plane is the positioning plane and the Z-axis is the cycle axis. However, the explanations may also be applied to the other planes.

4-1. Specification of Return Level (G71)

The G71 command specifies the return level to be used during the M53 mode (return to specified level).

(1) Programming format:

G71 Z_

"Z" indicates the cycle axis. One of a possible six axes should be chosen. The value should be specified as an absolute value in the local coordinate system, regardless of whether the G90 or G91 mode (absolute or incremental) has been selected.

(2) Notes

(a) The G71 code must precede the M53 code in the program.

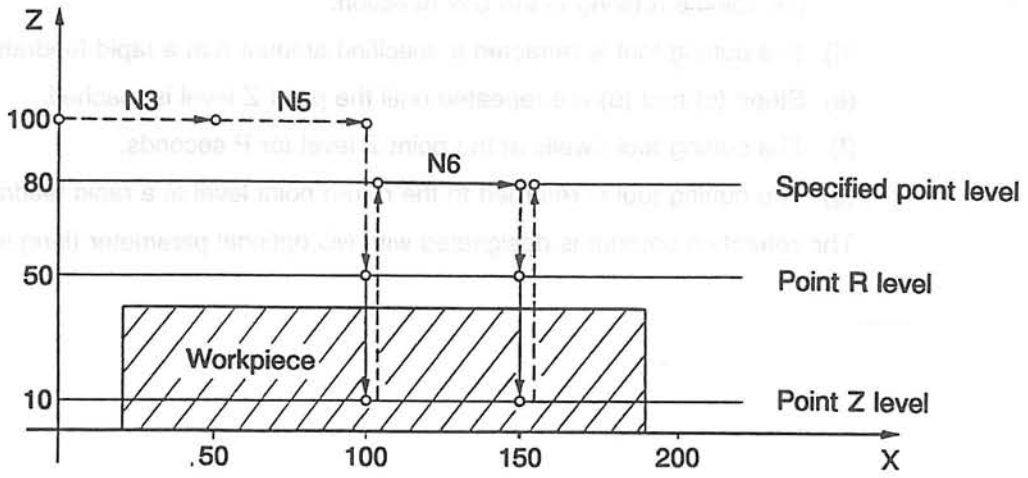
(b) The return level will be cleared when the NC is reset.

(c) The return level value is modal for each axis, and is not changed until it is defined again.

(d) If the local or work coordinate systems are changed, the return level value remains as it was.

Example:

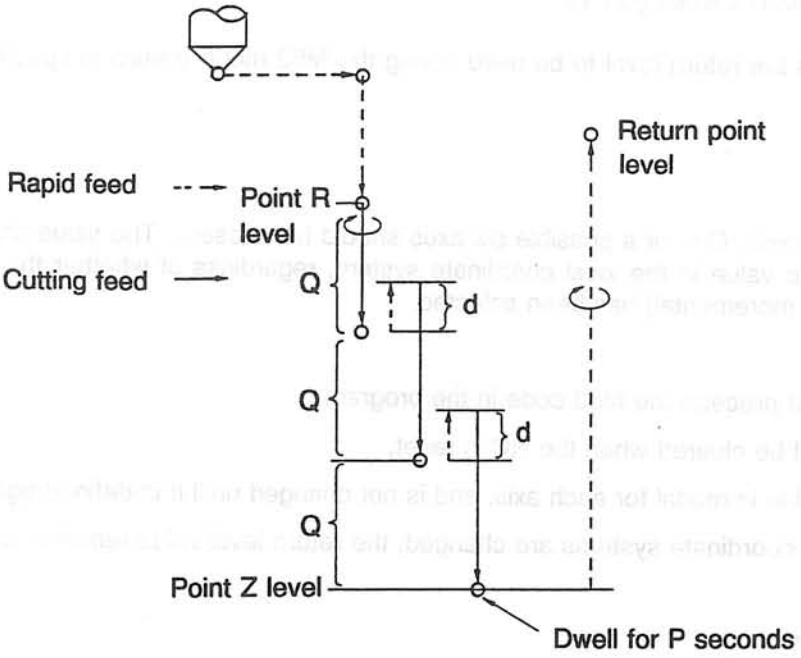
```
N1 G90 X0 Y0 Z100.0
N2 G91
N3 G17 G00   X50.0 Y50.0
N4 G71 Z80.0
N5 G81 X50.0 Y50.0 Z-40.0 R-50.0 F100 M53
N6     X50.0 Y50.0
N7 G80
```



4-2. High Speed Deep Hole Drilling Cycle (G73)

(1) Programming format:

G73 X_ Y_ Z_ R_ P_ Q_ F_



X, Y : Hole location

Z : Hole bottom level

In G90 mode: referenced from zero of the selected coordinate system

In G91 mode: referenced from the point R level

R : Point R level

P : Dwell time at hole bottom

Q : Pecking amount (positive, incremental value)

F : Feedrate

d : Retraction amount. (Set by NC optional parameter (long word) No. 1.)

(2) Machining Sequence

(a) Positioning is performed along the X- and Y-axis at a rapid feedrate.

(b) Positioning is performed to the point R level at a rapid feedrate.

(c) Drilling is performed up to the pecking amount specified by Q at a cutting feedrate and with the spindle rotating in the CW direction.

(d) The cutting tool is retracted a specified amount d at a rapid feedrate.

(e) Steps (c) and (d) are repeated until the point Z level is reached.

(f) The cutting tool dwells at the point Z level for P seconds.

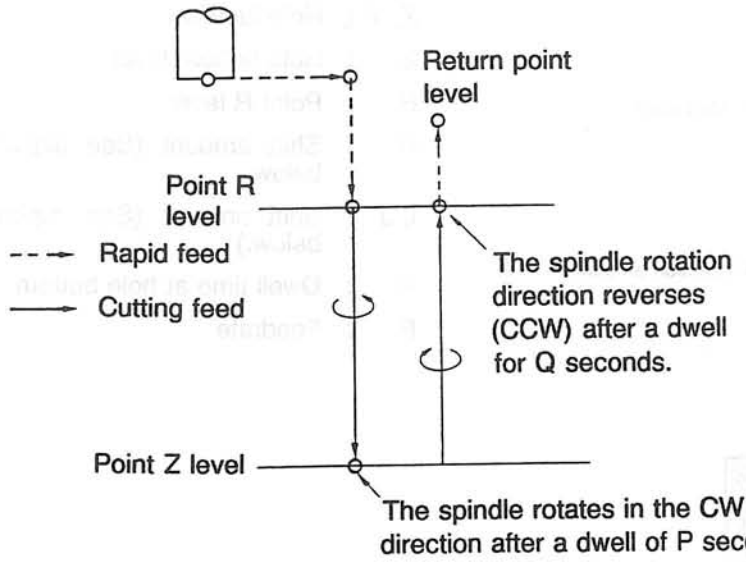
(g) The cutting tool is returned to the return point level at a rapid feedrate.

The retraction amount is designated with NC optional parameter (long word) No.1.

4-3. Reverse Tapping Cycle (G74)

(1) Programming format:

G74 X_ Y_ Z_ R_ P_ Q_ F_



X, Y : Hole location

Z : Hole bottom level

In G90 mode: referenced from zero of the selected coordinate system
In G91 mode: referenced from the point R level

R : Point R level

P : Dwell time at hole bottom

Q : Dwell time at point R level

F : Feedrate

(2) Machining Sequence

(a) Positioning is performed along the X- and Y-axis at a rapid feedrate.

(b) Positioning is performed to the point R level at a rapid feedrate.

(c) Tapping is performed up to the point Z level at a cutting feedrate and with the spindle rotating in the CCW direction.

(d) The tapping tool dwells at the point Z level for P seconds and then the spindle rotation direction is reversed to the CW direction.

(e) The tapping tool is returned to the point R level at a cutting feedrate.

(f) The tapping tool dwells at the point R level for Q seconds and then the spindle rotation direction is again reversed to the CCW direction.

(g) The tapping tool is returned to the return point level at a rapid feedrate.

(3) Notes

(a) No dwell motion will be performed if the P and Q values have not been specified.

(b) A feed override will be ignored during the reverse tapping operation.

(c) If the SLIDE HOLD button is pressed during the return of the tap from the point Z level to the point R level, the stop will not be performed until after the tap has been returned to the point R level.

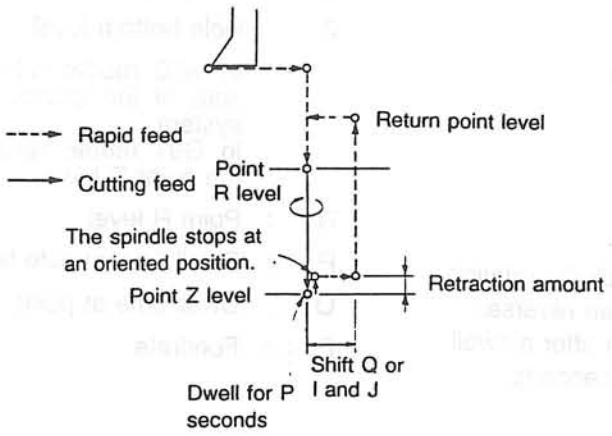
(d) If positioning to the next point is to be performed at the point R level after the start of the spindle counterclockwise rotation but before the tapping tool is disengaged from the workpiece completely, a dwell motion should be specified at the Q level.

(e) Both the cutting feedrate override and the spindle speed override command are fixed at 100%.
A rapid feed override command can be set.

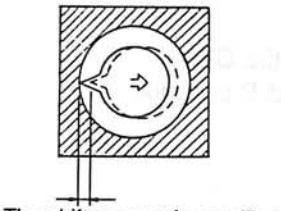
4-4. Fine Boring (G76)

(1) Programming format:

G76 X_ Y_ Z_ R_ Q_ (I_ J_) P_ F_



X, Y : Hole location
Z : Hole bottom level
R : Point R level
Q : Shift amount (See explanation below.)
I, J : Shift amount (See explanation below.)
P : Dwell time at hole bottom
F : Feedrate



The shift amount is specified by Q or I and J.

(2) Machining Sequence

(a) Positioning is performed along the X- and Y-axis at a rapid feedrate.

(b) Positioning is performed to the point R level at a rapid feedrate.

(c) Cutting is performed up to the point Z level at a cutting feedrate and with the spindle rotating in the CW direction.

(d) The cutting tool dwells at the point Z level for P seconds, the Z-axis is retracted an amount set by a parameter, the spindle is stopped at an oriented position, and the cutting tool is shifted the shift amount, Q, to the direction opposite the cutting tool.

(e) The cutting tool is returned to the return point level at a rapid feedrate.

(f) The cutting tool is shifted back the shift amount, Q, at the return point level.

Note: The Z-axis retraction amount from the point Z level is set by the NC optional parameter (long word) No. 11.

(3) Shift amount

The shift amount may be specified using either a Q value, or I and J values.

(a) Shift amount, Q

The Q value may be used as the shift amount when the Z-axis has been set as the cycle axis using a parameter. The Q value is always positive. The direction of the shift motion, +X, −X, +Y, −Y, must be set by a parameter in advance. The Q value is modal and is also used for the G73 and G83 commands. A Q value is given priority over the I and J values.

(b) Shift amount, I and J

The I and J values are used when the positioning plane has been selected using the G17 command. If the G18 or G19 command has been used to select the positioning plane, then the K, I and J, K values can be used, respectively.

G17 ..... I, J

G18 ..... K, I

G19 ..... J, K

All I, J, and K values are programmed as incremental values.
The shift direction is on the machine coordinate system.

Note: When I, J, or Q value is not given, an alarm occurs.

4-5. Fixed Cycle Cancel (G80)

The G80 command cancels all fixed cycle modes (G73, G74, G76, G81 through G87, and G89). All hole machining defining commands, such as point R and point Z levels are canceled. The interpolation mode (G00 through G03, G60) specified before the fixed cycle mode is reestablished. At the same time, the M05 code is generated to stop the spindle.

(1) However, if G00 or G01 is specified before the G80 block, the spindle will not stop.

```
G81  X_ Y_ Z_ R_ F_
G00  X_ Y_
G80
```

(2) In this example, the spindle does not stop rotating.

```
G81  X_ Y_ Z_ R_ F_
G80
G00  X_ Y_
```

(3) In this example the spindle stops rotating.Cancel with G00, G01, G02, G03

The fixed cycle mode may also be canceled by specifying a G00, G01, G02, or G03 interpolation command during the fixed cycle.

(4) Axis movement commands in G80 block

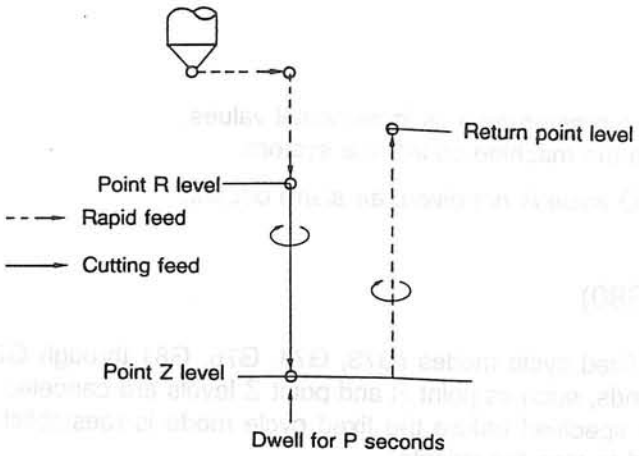
If axis movement commands are programmed in the G80 block, the fixed cycle mode will be canceled and then the axis movement will be executed.

4-6. Drilling Cycle (G81, G82)

(1) Programming format:

G81 X_ Y_ Z_ R_ P_ F_
G82 X_ Y_ Z_ R_ P_ F_

X, Y : Hole location
Z : Hole bottom level
R : Point R level
P : Dwell time at hole bottom
F : Feedrate



Note: The G81 and the G82 can be used equally.

(2) Machining Sequence

(a) Positioning is performed along the X- and Y-axis at a rapid feedrate.

(b) Positioning is performed to the point R level at a rapid feedrate.

(c) Cutting is performed up to the point Z level at a cutting feedrate and with the spindle rotating in the CW direction.

(d) The cutting tool dwells at the point Z level for P seconds.

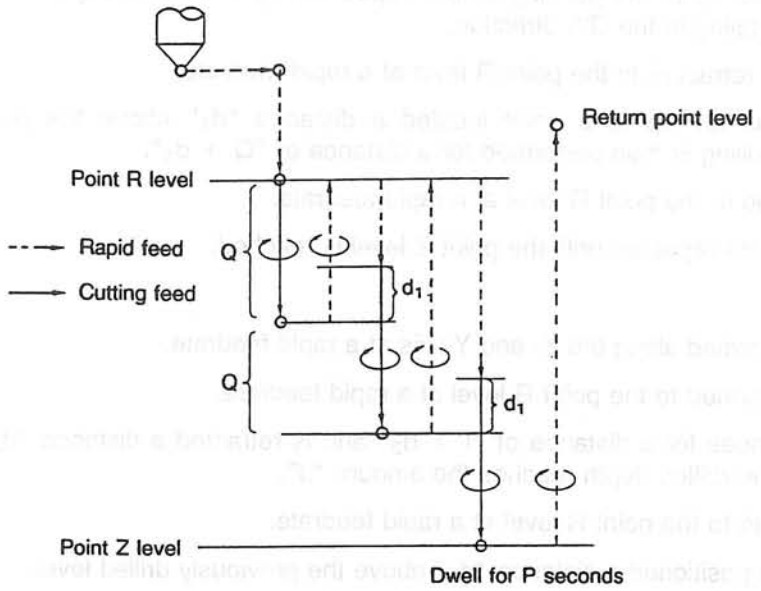
(e) The cutting tool is returned to the return point level at a rapid feedrate.

4-7. Deep Hole Drilling Cycle (G83)

(1) Programming format

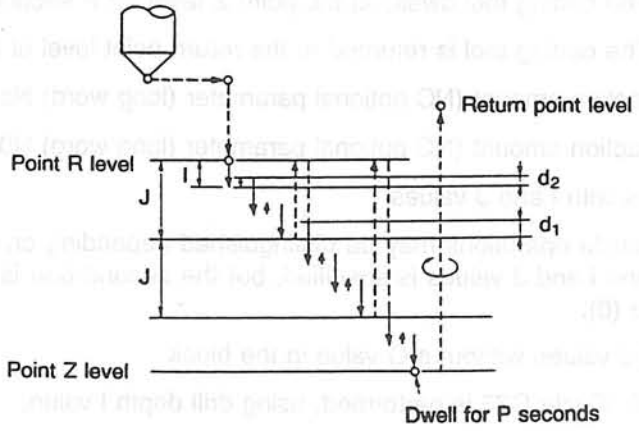
G83 X_ Y_ Z_ R_ Q_ (I_ J_) P_ F_

(2) With Q command



X, Y : Hole location
Z : Hole bottom level
R : Point R level
Q : Packing amount (Positive,incremental Valve)
I : Packing amount
J : Tool tip extract amount
P : Dwell time at hole bottom
F : Feedrate

(3) With I, J command



If a Q value is programmed in the same block as I and J values, the Q value will be given priority.

(4) Machining Sequence

(a) With Q value

① Positioning is performed along the X- and Y-axis at a rapid feedrate.

② Positioning is performed to the point R level at a rapid feedrate.

③ Drilling is performed up to the pecking amount specified by Q at a cutting feedrate and with the spindle rotating in the CW direction.

④ The cutting tool is retracted to the point R level at a rapid feedrate.

⑤ The cutting tool is fed up to a point located a distance "d_1" above the previously machined level. Drilling is then performed for a distance of "Q + d_1".

⑥ The tool is returned to the point R level at a rapid feedrate.

⑦ Steps ⑤ and ⑥ are repeated until the point Z level is reached.

(b) With I and J values

① Positioning is performed along the X- and Y-axis at a rapid feedrate.

② Positioning is performed to the point R level at a rapid feedrate.

③ Drilling then continues for a distance of "I + d_2" and is retracted a distance "d_2". This is repeated until the drilled depth reaches the amount "J".

④ The tool is returned to the point R level at a rapid feedrate.

⑤ The cutting tool is positioned a distance "d_1" above the previously drilled level.

⑥ Drilling is performed for a distance of "I + d_1" and then retracted an amount "d_2". This is repeated until the drilled depth from this cycle reaches the amount "J".
The tool is returned to the point R level at a rapid feedrate.

⑦ Steps ⑤ and ⑥ are repeated until the point Z level is reached.

⑧ The cutting tool dwells at the point Z level for P seconds.

⑨ The cutting tool is returned to the return point level at a rapid feedrate.

(5) d_1 : Retraction amount (NC optional parameter (long word) No. 2.)

(6) d_2 : Retraction amount (NC optional parameter (long word) N0. 1.)

(7) Operations with I and J values

Different cycle operations may be distinguished depending on the I and J values used. Note that if one of the I and J values is specified, but the second one is not, the second value is assumed to be zero (0).

(a) I and J values without a Q value in the block

J = 0 Cycle G73 is performed, using drill depth I value.

I ≧ J Cycle G83 is performed, using drill depth J value.

I < J Cycle G83 is performed; the definitions of I and J, as explained in (4) (b) above, are reversed.

I = 0 An alarm is activated. (Level Q alarm.)

(b) I and J values with a Q value in the block

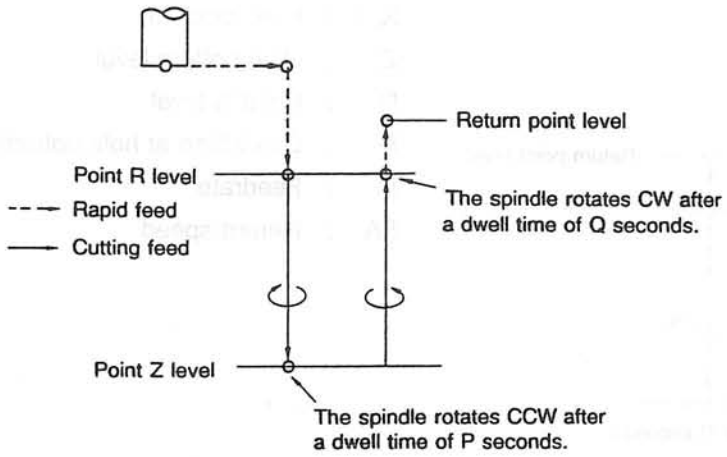
The Q value is given priority over I and J and the sequence described in (4) (a) above is performed.

4-8. Tapping Cycle (G84)

(1) Programming format:

G84 X_ Y_ Z_ R_ (P_) (Q_) F_

X, Y : Hole location
Z : Hole bottom level
R : Point R level
P : Dwell time at hole bottom
Q : Dwell time at point R level
F : Feedrate



(2) Machining Sequence

(a) Positioning is performed along the X- and Y-axis at a rapid feedrate.

(b) Positioning is performed to the point R level at a rapid feedrate.

(c) Tapping is performed up to the point Z level at a cutting feedrate and with the spindle rotating in the CW direction.

(d) The tapping tool dwells at the point Z level for P seconds and then the spindle rotation direction is reversed to the CCW direction.

(e) The tapping tool is returned to the point R level at a cutting feedrate.

(f) The tapping tool dwells at the point R level for Q seconds and then the spindle rotation direction is again reversed to the CW direction.

(g) The tapping tool is returned to the return point level at a rapid feedrate.

(3) Notes

(a) No dwell motion will be performed if the P and Q values have not been specified. Refer to Section 8-3-1, (3) for details on the units of the P and Q values.

(b) Both the cutting feedrate override and the spindle speed override command are fixed at 100%. Any other settings will be ignored during the tapping operation. However, a rapid feed override command can be used.

(c) If the **SLIDE HOLD** button is pressed during the return of the tap from the point Z level to the point R level, the stop will not be performed until after the tap has been returned to the point R level.

(d) If positioning to the next point is to be performed at the point R level after the start of the spindle counterclockwise rotation but before the tapping tool is disengaged from the workpiece completely, a dwell motion should be specified at the Q level.

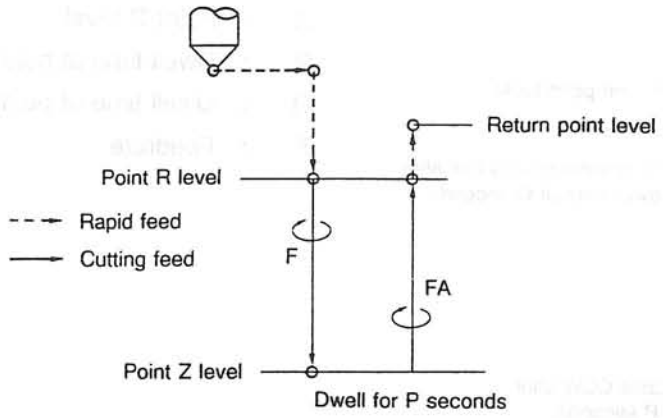
(e) Both the cutting feedrate override and the spindle speed override command are fixed at 100%. A rapid feedrate override command can be set.

4-9. Boring Cycle (G85, G89)

(1) Programming format:

G85 X_ Y_ Z_ R_ P_ F_ FA=_
G89 X_ Y_ Z_ R_ P_ F_ FA=_

X, Y : Hole location
Z : Hole bottom level
R : Point R level
P : Dwell time at hole bottom
F : Feedrate
FA : Return speed



(2) Machining Sequence

(a) Positioning is performed along the X- and Y-axis at a rapid feedrate.

(b) Positioning is performed to the point R level at a rapid feedrate.

(c) Cutting is performed up to the point Z level at a cutting feedrate and with the spindle rotating in the CW direction.

(d) The cutting tool dwells at the point Z level for P seconds.

(e) The tool is returned to the point R level at the return speed, FA. If no value for FA has been specified, the tool will be returned at the feedrate, F.

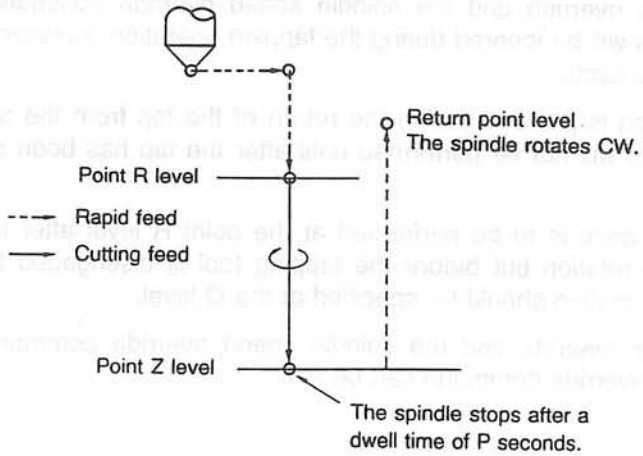
(f) The tool is returned to the return point level at a rapid feedrate.

4-10. Boring Cycle (G86)

(1) Programming format:

G86 X_ Y_ Z_ R_ P_ F_

X, Y : Hole location
Z : Hole bottom level
R : Point R level
P : Dwell time at hole bottom
F : Feedrate



(2) Machining Sequence

(a) Positioning is performed along the X- and Y-axis at a rapid feedrate.

(b) Positioning is performed to the point R level at a rapid feedrate.

(c) Cutting is performed up to the point Z level at a cutting feedrate and with the spindle rotating in the CW direction.

(d) The cutting tool dwells at the point Z level for P seconds and the spindle stops rotating.

(e) The tool is returned to the return point level at a rapid feedrate.

(f) The spindle rotates in the CW direction at the return point level.

(3) The difference between the G86 boring cycle and the G81/G82 drilling cycles is that the spindle stops at the hole bottom level during the G86 cycle.

(4) Note

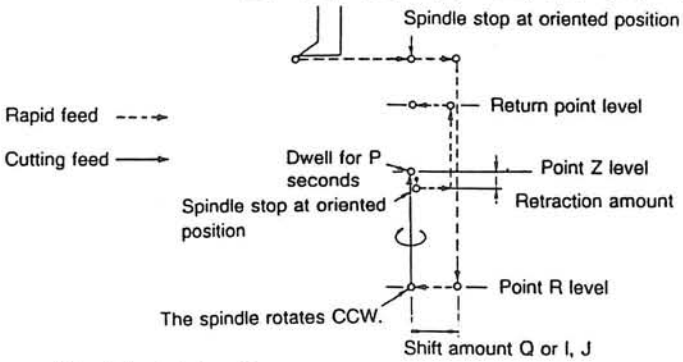
If the distance to the point R level of the next hole is short, the spindle, which has started to rotate clockwise at the return point level, may not reach the commanded speed. Therefore, it is recommended to program a dwell command (G04) with a proper dwell time, before the boring of the next point is started.

4-11. Back Boring Cycle (G87)

Note that this cycle differs somewhat from the other fixed cycles. The G71 command must be specified in advance.

(1) Programming format:

G87 X_ Y_ Z_ R_ Q_ (I_ J_) P_ F_



X, Y : Hole location
Z : Machining end level (Point Z level)
R : Machining start level (Point R level)
Q : Shift amount Refer to Section 8-4-4 (3) for explanation
I, J : Shift amount
P : Dwell time at point Z level
F : Feedrate

(2) Machining Sequence

(a) Positioning is performed along the X- and Y-axis at a rapid feedrate. The spindle then stops at an oriented position.

(b) The tool shifts in the direction opposite the cutting bit by the shift amount Q.

(c) Positioning is performed to the point R level at a rapid feedrate.

(d) At the point R level, the tool is shifted back an amount Q, so that it is directly beneath the original position in step 1). The spindle then starts to rotate in the CW direction.

(e) Boring is performed up to the point Z level in the positive direction.

(f) The cutting tool dwells at the point Z level for P seconds, the Z-axis is retracted down by an amount set with a parameter, and the spindle stops at an oriented position. The tool is then shifted in the direction opposite the bit position by the shift amount Q.

(g) The tool is returned to the return point level at a rapid feedrate.

(h) The shifted tool is returned to its original position in step (a) by the shift amount Q.

(3) Return point level

There are three ways to specify the return point level:

(a) M52 upper limit level

(b) M53 specified point level (to be set in advance with G71)

(c) M54 start point level

SECTION 7 COORDINATE CALCULATION FUNCTION (OPTIONAL)

This function calculates the coordinate values of points on a line, grid, or circumference using one command.
Combining this function with the fixed cycle function and the axis movement call function allows machining such as drilling to be conducted at points on a line, grid, or circumference by designating one command.

1. List of Functions

Item	Mnemonic Code	Function Outline
Omit	OMIT	Deletes coordinate calculation of a specified point
Restart	RSTRT	Starts coordinate calculation from a certain point on a line, grid, or circumference.
Line at angle	LAA	Outputs the coordinate values of points on a line which has an angle.
Arc	ARC	Outputs the coordinate values of points on an arc.
Grid X	GRDX	Outputs the coordinate values of points on a grid. (The calculating sequence is distinguished by X and Y.)
Grid Y	GRDY	
Double grid X	DGRDX	Outputs the coordinate values of points on two grids. (The calculating sequence is distinguished by X and Y.)
Double grid Y	DGRDY	
Square X	SQRX	Outputs the coordinate values of points on two grids. (The calculating sequence is distinguished by X and Y.)
Square Y	SQRY	
Bolt hole circle	BHC	Outputs the coordinate values of points on a circumference.

2. General Rules of Coordinate Calculation

(1) Programming Format for Coordinate Calculation

The programming format is as indicated below.

【 (Mnemonic code) Hp_Vp_I_J_K_P_Q_R_ 】

Hp and Vp represent the coordinate values of the reference point for starting coordinate calculation.

I, J, K, P, and Q are parameters for coordinate calculation.

(a) Reference point for starting coordinate calculation

The reference point is specified using the names of axes which constitute the currently selected plane. The relation of plane selection to Hp (horizontal axis) and Vp (vertical axis) is as listed below.

Plane	G17				G18				G19			
Axis	X-Y	X-V	U-Y	U-V	Z-X	Z-U	W-X	W-U	Y-Z	Y-W	V-Z	V-W
Hp (horizontal axis)	X	X	U	U	Z	Z	W	W	Y	Y	V	V
Vp (vertical axis)	Y	V	Y	V	X	U	X	U	Z	W	Z	W

1) The reference point is specified using axes which constitute the currently selected plane. Designation other axes is ignored. When the coordinate value of the reference point is not designated, the actual position value is taken as the coordinate value of the reference point.

2) The reference point is designated in the local coordinate system. The designation method varies between the absolute dimensioning mode (G90) and the incremental dimensioning mode (G91).

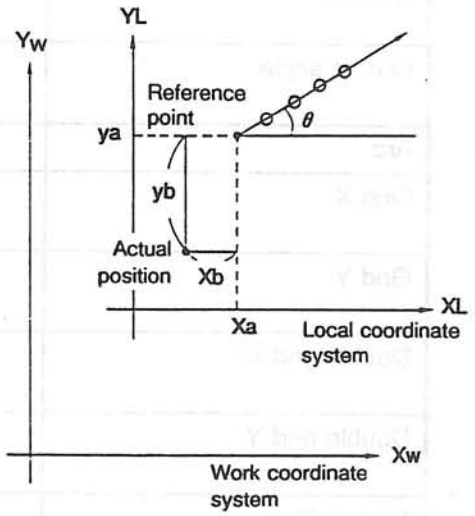
Example:

Absolute dimensioning mode

G90 LAA Xxa Yya

Incremental dimensioning mode

G91 LAA Xxb Yyb



(b) Parameters used for coordinate calculation

Parameters used in coordinate calculation must be designated in the same block as corresponding mnemonic codes. These parameters are effective only in the block where they are designated, and are cleared after the completion of coordinate calculation.
The relation between the coordinate calculation function and the parameters is as indicated below.

Item	Mnemonic Code	Reference Point		Parameter						Remarks
		H_P	V_P	I	J	K	P	Q	R	
Omit	OMIT								◎	More than one parameter can be specified.
Restart	RSTRT								◎	Only one parameter can be specified.
Line at angle	LAA	○	○	◎	◎	△1				and K must be specified in pairs. These parameters can be omitted when K is equal to 1.
Arc	ARC	○	○	◎	◎	△1		◎		Q and K must be specified in pairs. These parameters can be omitted when K is equal to 1.
Grid	GRDX GRDY	○	○	◎	◎	◎	◎			
Double grid	DGRDX DGRDY	○	○	◎	◎	◎	◎	△2	△3	
Square	SQRX SQRY	○	○	◎	◎	◎	◎			
Bolt hole circle	BHC	○	○	◎	◎	◎				

○: Can be omitted. The default is actual position data.

◎: Cannot be omitted. If omitted, an alarm is activated.

△: Can be omitted.

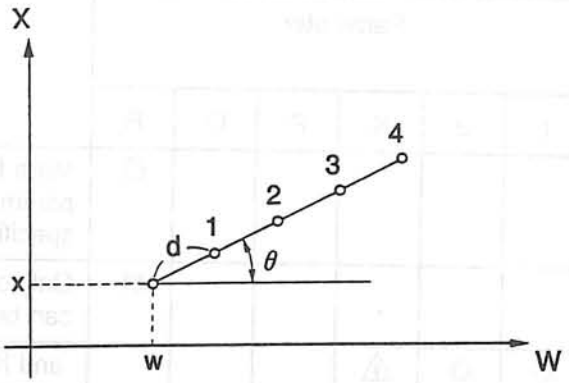
If omitted, △1 is regarded as 1, △2 as I/2, and △3 as J/2.

Data specified in blank columns will be ignored.

(2) Plane on Which Coordinate Calculation is Made and Motion Axes

Coordinate values are calculated on the plane which is selected when a pattern command is designated, and positioning at each calculated point is accomplished using the axes which are determined in accordance with plane selection.

Example: Pattern development on the WX plane




```
G90
G18W_X_
NCYL G81 Y_R_F_
LAA Ww Xx Id Jθ K4
```

(a) When the coordinate calculation function is used in combination with a fixed cycle function, the coordinate value of the axis which is perpendicular to the plane at the end of pattern designation is not necessarily identical to the coordinate value which was active right before pattern designation.

(b) Refer to the explanation for point R return and specified point return of the fixed cycle function.
When return to the upper limit level (M52) is specified, axes will return to the upper limit only after the last point of the pattern has been machined.

If output of the coordinate value of the last point has been deleted using the OMIT command, axes will not return to the upper limit level.

(3) Positioning at Calculated Pattern Points

Positioning is made in the G00, G60, or G01 mode, or in the mode determined by the fixed cycle command. An alarm occurs if positioning is attempted in the M02 or M03 mode.

(4) When the Single Block Mode is Effective

When the single block mode is effective, the operation halts after positioning at the calculated pattern points has been completed.

(5) A maximum of 65535 points can be calculated by one pattern command.

(6) A pattern command cannot be designated while any of the following modes is effective.

(a) Circular interpolation mode (G02, G03)

(b) Cutter radius compensation mode (G41, G42)

(c) 3-D tool offset mode

(d) Coordinate calculation is in progress

3. Details of Coordinate Conversion Function (Pattern Function)

3-1. Omit (OMIT)

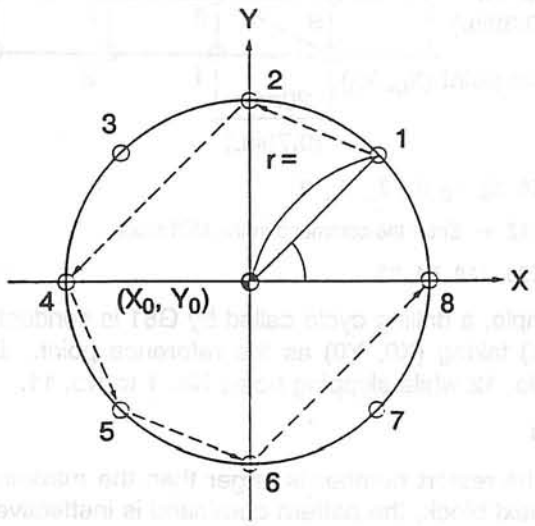
This function is normally used in combination with other coordinate calculation functions and deletes output of the coordinate value which is calculated using the coordinate calculation function.

Programming format: 〖 OMIT R_{n1}__ R_{n2}__ R_{n3}__ ... R_{nm}__ 〗

R: Coordinate point number (n) to be deleted ($m \leqq 30$)

Example:

```
NCYL G81 G56 R_ Z_ F_ H04
OMIT R3 R7
BHC X0 Y0U50 J45 K8
```



In this example, a drilling cycle called by G81 is conducted at positions on the circumference (BHC) taking (X_0, Y_0) as the reference point. The 3rd and 7th holes are omitted by designating R3 and R7.

(1) Precautions

(a) A maximum of 30 R's (omit numbers) can be designated in one pattern command.

(b) The maximum number for R is 65,535.

* If (a) and (b) are not satisfied in one pattern, the pattern must be divided into two or more patterns.

(c) Designate the OMIT command before a coordinate calculation function is designated. The OMIT command is cleared automatically after the execution of calculation.

(d) Define the omit number (R) in the order by which coordinate calculation is conducted. However, the coordinate value can be designated in any order. If the programmed omit number is larger than the maximum number of the points calculated using this function, the programmed omit number is ignored.

(e) When more than one OMIT command is designated, the OMIT command right before the coordinate calculation command is effective.

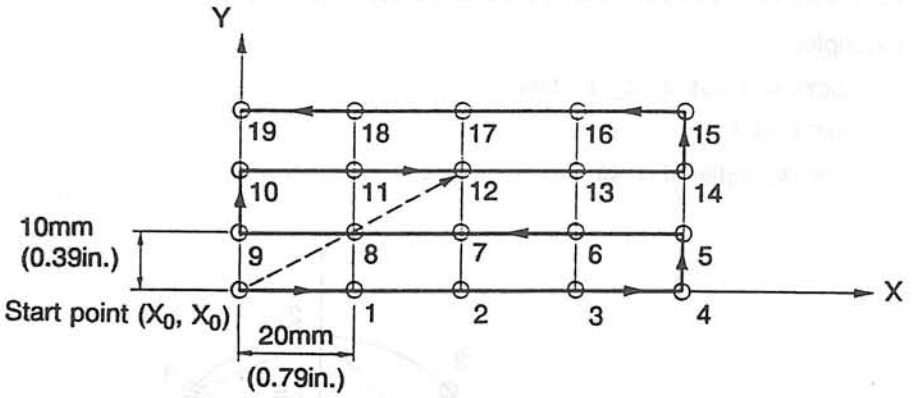
3-2. Restart (RSTRT)

This function restarts machining from the required coordinate point which is calculated using the coordinate calculation function.
Generally, restart data is entered in the MDI mode.

Programming format: 【 RSTRT Rn 】

R: Coordinate point number (n) at which machining is restarted

Example:




```
N011 G81 G56 X0 Y0 R_ Z_ F_ H
    RSTRT R12 ← Enter the command in the MDI mode.
N012 GRDX I20 J10 K4 P3
```

In this example, a drilling cycle called by G81 is conducted at positions on the circumference of a grid (GRDX) taking (X0, Y0) as the reference point. Designation of RSTRT R12 starts drilling from hole No. 12 while skipping holes No. 1 to No. 11.

(1) Precautions

(a) When the restart number is larger than the maximum point number of the pattern command in the next block, the pattern command is ineffective.

(b) When more than one restart command is designated, the last restart command, or, in other words, the restart command designated right before the pattern command in the next block becomes effective.

3-3. Line at Angle (LAA)

This function calculates the coordinate values of points placed at irregular intervals (d1, d2, and so forth) on a line which forms an angle θ to the horizontal axis. Here, the actual position value or commanded coordinate value is taken as the reference point.
When points are placed at regular intervals, designate the number of these points (n).

Programming format: 【 LAA Hp__ Vp__ I$\pm$d$_1$ K$_{n1}$ I$\pm$d$_2$ K$_{m2}$ .... J$\pm\theta$ 】

Hp : Coordinate value of the reference point on the horizontal axis

Vp : Coordinate value of the reference point on the vertical axis

I : Interval (d)

When d has a negative value, the coordinate value is determined at the point which is symmetric taking the start point as the center.

K : Number of points placed at regular intervals (n)

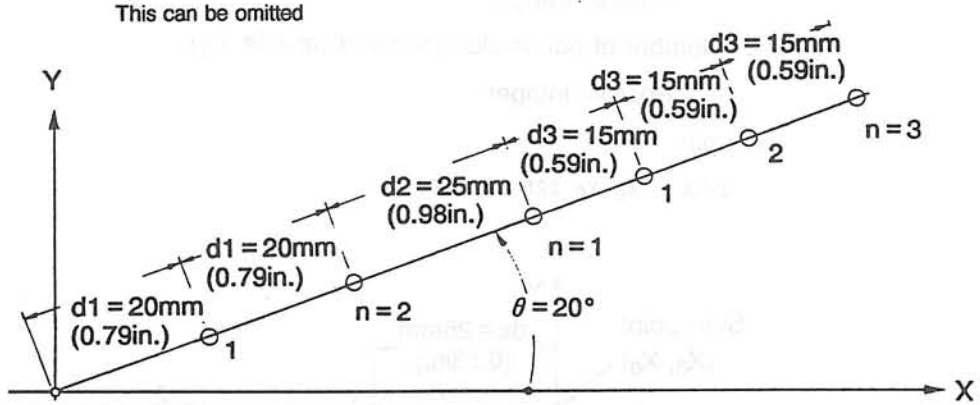
The maximum number is 65,535.

J : Angle with reference to the horizontal axis (θ)

The angle is set in units of 1 degree, 0.001 degree, or 0.0001 degree in accordance with the selected unit system.
The angle increases in the CCW direction.

Example:

```
LAA X0 Y0 I20 K2 I25 (K1) I15 K3 J20
                          ↑
                          This can be omitted
```



Start point (X_0, X_0)

(1) Precautions

(a) The default of K is 1.

(b) Up to ten intervals can be designated in one block.

(c) When I and K in pairs and J are designated, the LAA command can be used as a unit-pitch line at angle command.

(d) The coordinate values of the reference point are not output.

(e) The order for designating I, J, and K must be either J, I, K, I, K ... or I. K. I. K ... J.

3-4. Grid (GRDX, GRDY)

This function calculates the coordinate values of a grid which is composed of the points (nx) placed at an interval of (dx) in parallel with the horizontal axis and of the points (ny) placed at an interval of (dy) in parallel with the vertical axis. Here, the actual position value or commanded coordinate value is taken as the reference point.
The coordinate values are calculated along the horizontal axis by designating GRDX while the coordinate values are calculated along the vertical axis by designating GRDY.

Programming format: 【 {GRDX GRDY} Hp__ Vp__ I±dx J±dy Knx Pny 】

Hp : Coordinate value of the reference point on the horizontal axis

Vp : Coordinate value of the reference point on the vertical axis

I : Interval along the horizontal axis (dx)

When dx has a positive value, coordinate calculation is performed in the horizontal-axis positive direction.
When dx has a negative value, coordinate calculation is performed in the horizontal-axis negative direction.

J : Interval along the vertical axis (dy)

When dy has a positive value, coordinate calculation is performed in the vertical-axis positive direction.
When dy has a negative value, coordinate calculation is performed in the vertical-axis negative direction.

K : Number of points along the horizontal axis (nx)

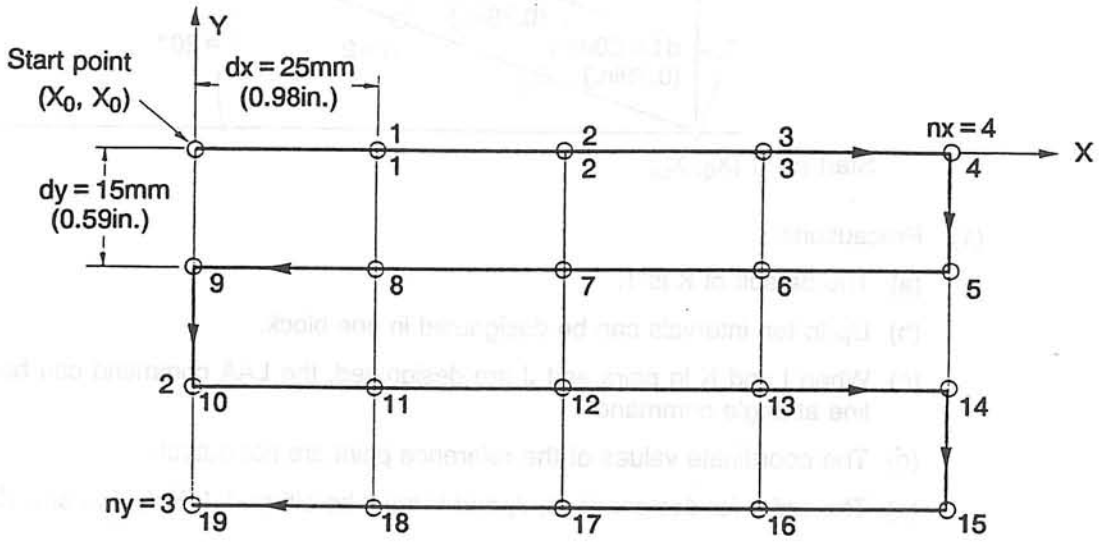
Positive integer

P : Number of points along the vertical axis (ny)

Positive integer

Example:

```
GRDX   X0 Y0 I25 J-15 K4 P3
```



(1) Precautions

(a) The maximum number of points on a grid ((nx + 1) × (ny + 1) − 1) is 65,535.

(b) The number of the last point is (nx + 1) × (ny + 1) − 1.

(c) The coordinate values of the reference point are not output.

3-5. Double Grid (DGRDX, DGRDY)

This function calculates the coordinate values of a grid which is composed of the points (nx) placed at regular intervals (dx_1) in parallel with the horizontal axis and of the points (ny) placed at regular intervals (dy_1) in parallel with the vertical axis, and the coordinate value of the other grid which is obtained by shifting the reference grid both in the X- and Y-axis directions by "dx_2" and "dy_2", respectively. Here, the actual position value or commanded coordinate value is taken as the reference point,
The coordinate values are calculated along the horizontal axis by designating DGRDX while the coordinate values are calculated along the vertical axis by designating DGRDY.

Programming format: 【 {DGRDX DGRDY} Hp__ Vp__ I±dx_1 J±dy_1 Knx Pny Q±dx_2 Rdy_2 】

Hp : Coordinate value of the reference point on the horizontal axis

Vp : Coordinate value of the reference point on the vertical axis

I : Interval along the horizontal axis (dx_1)

When dx_1 has a positive value, coordinate calculation is performed in the horizontal-axis positive direction.
When dx_1 has a negative value, coordinate calculation is performed in the horizontal-axis negative direction.

J : Interval along the vertical axis (dy_1)

When dy_1 has a positive value, coordinate calculation is performed in the vertical-axis positive direction.
When dy_1 has a negative value, coordinate calculation is performed in the vertical-axis negative direction.

K : Number of points along the horizontal axis (nx)

Positive integer

P : Number of points along the vertical axis (nx)

Positive integer

Q : Shift interval in the horizontal-axis direction (dx_2)

When dx_2 has a positive value, coordinate calculation is performed in the horizontal-axis positive direction.

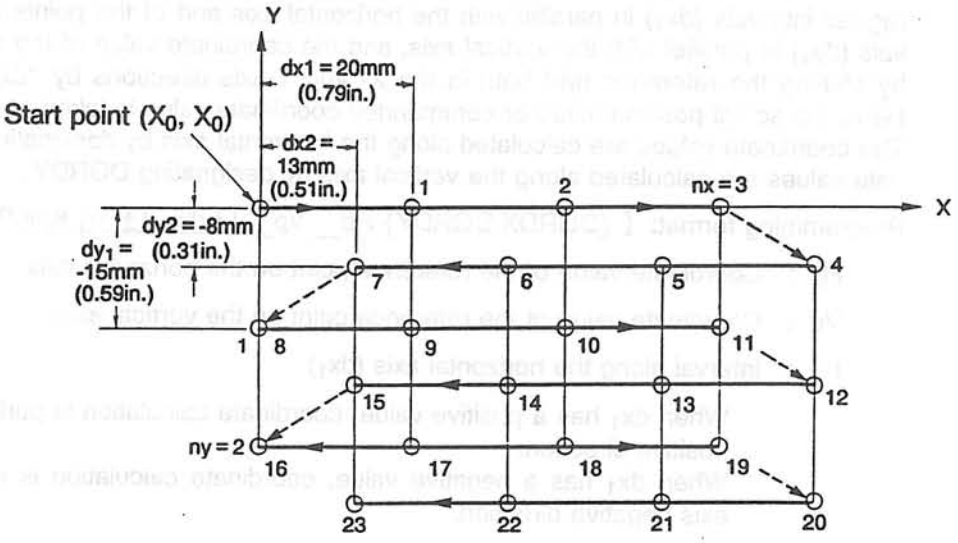
When dx_2 has a negative value, coordinate calculation is performed in the horizontal-axis negative direction.

R : Shift interval in the vertical-axis direction (dy_2)

When dy_2 has a positive value, coordinate calculation is performed in the vertical-axis positive direction.
When dy_2 has a negative value, coordinate calculation is performed in the vertical-axis negative direction.

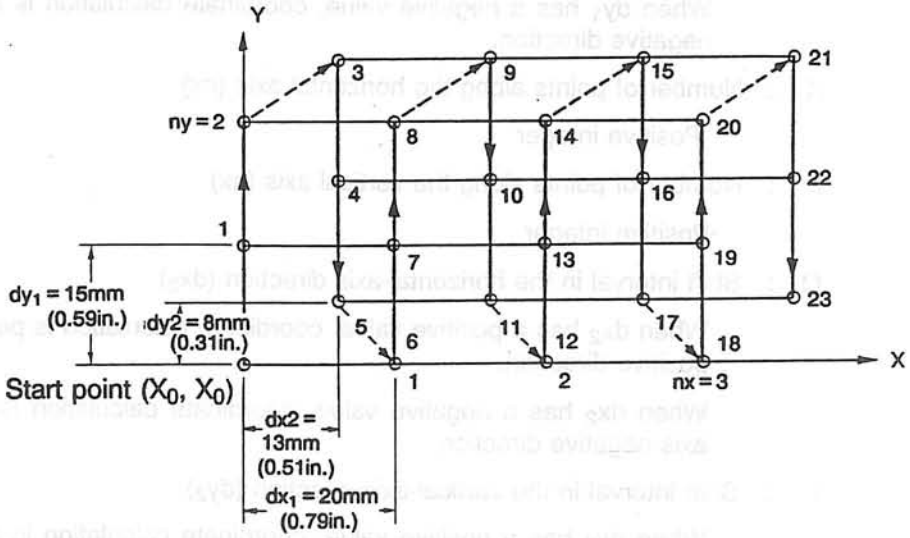
Example 1: Double grid along the horizontal axis

```
DGRDX X0 Y0 I20 J-15 K3 P2 Q13 R-8
```



Example 2: Double grid along the vertical axis

```
DGRDX X0 Y0 I20 J15 K3 P2 Q13 R8
```



(1) Precautions

(a) When dx_2 is equal to $dx_1/2$, designation of Q can be omitted. In the same manner, when dy_2 is equal to $dy_1/2$, designation of R can be omitted.

(b) The maximum number of points on a grid $(2(nx+1)\times(ny+1)-1)$ is 65,535.

(c) The number of points to be calculated at an interval of the shift amount is the same as (nx, ny).

(d) The coordinate values of the reference point are not output.

3-6. Square (SQRX, SQRY)

This function calculates the coordinate values of a square which is composed of the points (nx) placed at regular intervals (dx) in parallel with the horizontal axis and of the points (ny) placed at regular intervals (dy) in parallel with the vertical axis. Here, the actual position value or commanded coordinate value is taken as the reference point.
The coordinate values are calculated in the horizontal-axis direction by designating SQRX while the coordinate values are calculated in the vertical-axis direction by designating SQRY.

Programming format: [SQRX SQRY Hp__ Vp__ I±dx J±dy Knx Pny]

Hp : Coordinate value of the reference point on the horizontal axis

Vp : Coordinate value of the reference point on the vertical axis

I : Interval along the horizontal axis (dx)

When dx has a positive value, coordinate calculation is performed in the horizontal-axis positive direction.
When dx has a negative value, coordinate calculation is performed in the horizontal-axis negative direction.

J : Interval along the vertical axis (dy)

When dy has a positive value, coordinate calculation is performed in the vertical-axis positive direction.
When dy has a negative value, coordinate calculation is performed in the vertical-axis negative direction.

K : Number of points along the horizontal axis (nx)

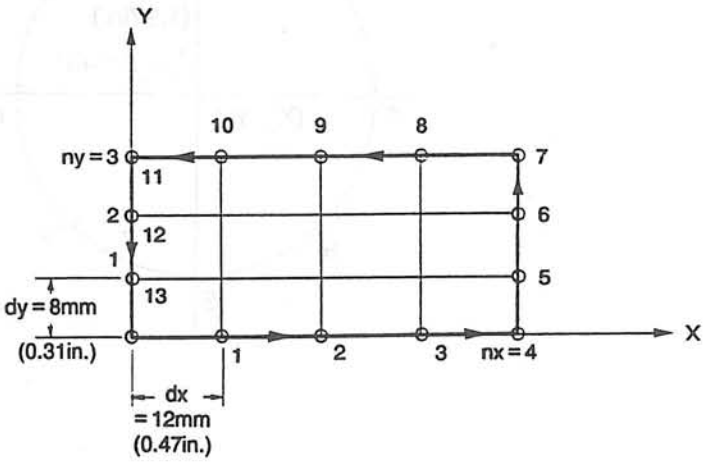
Positive integer

P : Number of points along the vertical axis (ny)

Positive integer

Example:

```
SQRX X0 Y0 I12 J8 K4 P3
```



(1) Precautions

(a) The maximum number of points on a square 2(nx + ny) − 1 is 65,535.

(b) The coordinate values of the reference point are not output.

3-7. Bolt Hole Circle (BHC)

This function calculates the coordinate values of the points (n points) which are placed at regular intervals on the circumference of a circle which has the actual position value or commanded coordinate value as the center and radius "r". Here, the start angle of the points is "θ" with reference to the horizontal axis.

Programing format : 【 BHC H_P__V_P__Ir J±θ K±n 】

Hp : Center coordinate value on the horizontal axis

Vp : Center coordinate value on the vertical axis

I : Radius of the circle (r)

A positive value is set.

J : Start angle (θ)

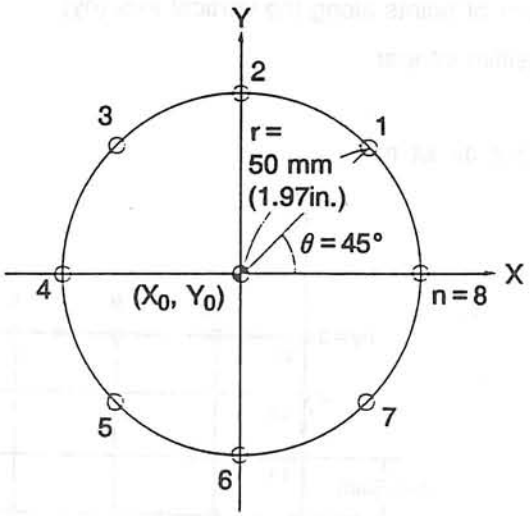
The angle is set in units of 1 degree, 0.001 degree, or 0.0001 degree in accordance with the selected unit system.
The angle increases in the CCW direction.

K : Number of points (n)

The coordinate value is calculated in the CCW direction when n has a positive value.
The coordinate value is calculated in the CW direction when n has a negative value.
The maximum number of points is 65,535.

Example:

```
BHC X0 Y0 I50 J45  K8
```



3-8. Arc (ARC)

This function calculates the coordinate values of the points (n points) which are placed at irregular intervals ($\triangle\theta_1$, $\triangle\theta_2$) on the circumference of a circle which has the actual position value or commanded coordinate value as the center and radius "r". Here, the start angle of the points is "θ" with reference to the horizontal axis.
If more than two points are placed at regular intervals ($\triangle\theta$) consecutively, designate the number (n) of such points.

Programing format : 〖 ARC Hp__Vp__IrQ$\pm\triangle\theta_1$K$_{n1}$ Q$\pm\triangle\theta_2$ K$_{n2}$ ·····J$\pm\theta$ 〗

Hp : Center coordinate value on the horizontal axis

Vp : Center coordinate value on the vertical axis

I : Radius of the circle (r)

A positive value is set.

Q : Irregular angle ($\triangle\theta$)

The angle is set in units of 1 degree, 0.001 degree, or 0.0001 degree in accordance with the selected unit system.
The angle increases in the CCW direction.

K : Number of points placed at regular intervals consecutively (n)

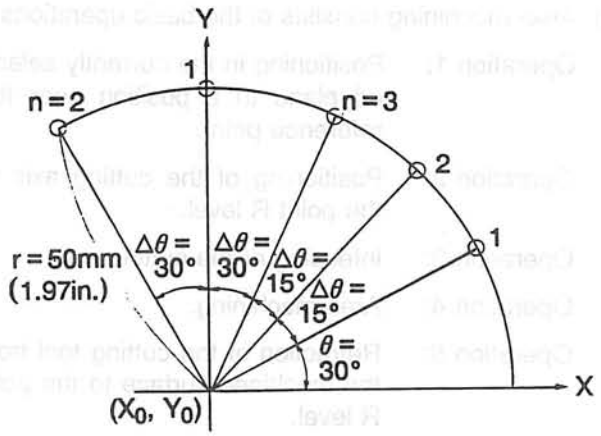
The maximum number is 65,535.

J : Start angle (θ)

The angle is set in units of 1 degree or 0.001 degree in accordance with the selected unit system.
The angle increases in the CCW direction.

Example:

```
ARC X0 Y0 I50 Q15  K3 Q30 K2 J30
```



(1) Precautions

(a) The default of K is 1.

(b) Up to 10 irregular angles (preceded by Q) can be set in one block.

(c) When I and K in pairs and J are designated, the ARC command can be used as a unit-pitch arc command.

SECTION 8 AREA MACHINING FUNCTIONS (OPTIONAL)

Area machining functions are used to machine the top, periphery or inside surface of a rectangular area with a single command. The area to be machined must be formed by four straight lines which intersect at right angles. The direction of cutting is in the longitudinal direction of the rectangular area.

1. List of Area Machining Functions

Category	Mnemonic Code	Area to be Machined	Cutting Direction	Remarks
Face Milling (Surface Milling)	FMILR	Top (Surface)	Longitudinal direction of rectangular area	Tool-ON
	FMILF	Top (Surface)	Same as above	Tool-OFF
Pocket Milling (Hole Machining)	PMIL	Inside	Same as above	Zigzag
	PMILR	Inside	Same as above	Spiral
Round Milling (Peripheral machining)	RMILO	External periphery	Same as above	
	RMILI	Internal periphery	Same as above	

2. Area Machining Operations

(1) Basic Operations

(a) Area machining consists of the basic operations shown below.

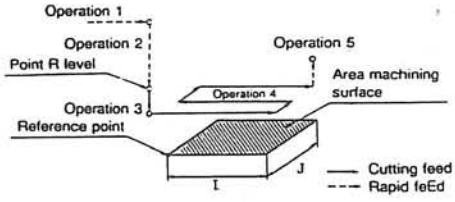
Operation 1: Positioning in the currently selected plane to a position near the reference point.

Operation 2: Positioning of the cutting axis to the point R level.

Operation 3: Infeeding of the cutting axis.

Operation 4: Area machining.

Operation 5: Retraction of the cutting tool from the machined surface to the point R level.

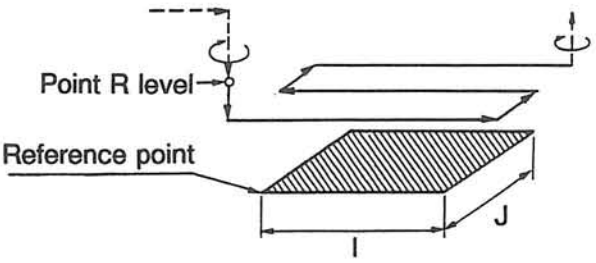


(b) Operations 3, 4, and 5 are repeated as necessary depending on the amount of stock to be machined. The next cutting position is defined as the first positioning point.

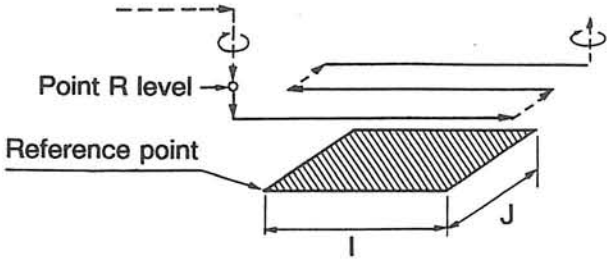
(c) The plane on which the area machining is to be performed is taken as the currently selected plane, and the cutting tool is infed on the axis vertical to the plane.

(d) By programming the M52 command, the Z-axis will be returned to the upper limit level after the final cutting. This is possible only when the cycle axis is the Z-axis.

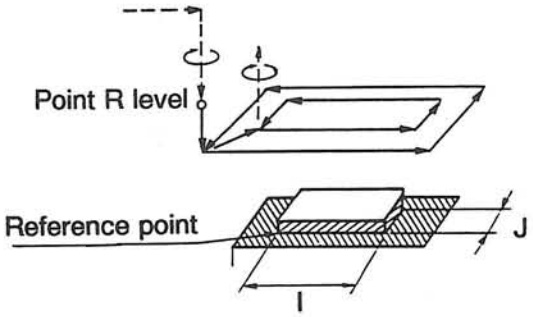
(2) Tool Movement



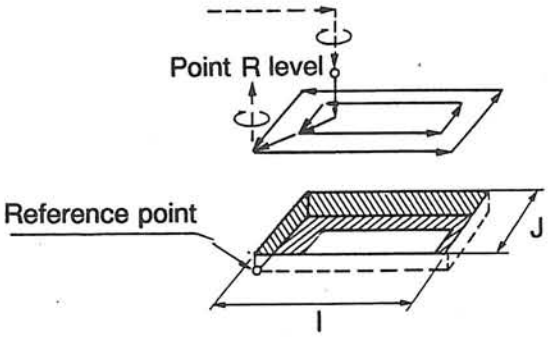
(a) Face Milling (FMILR)



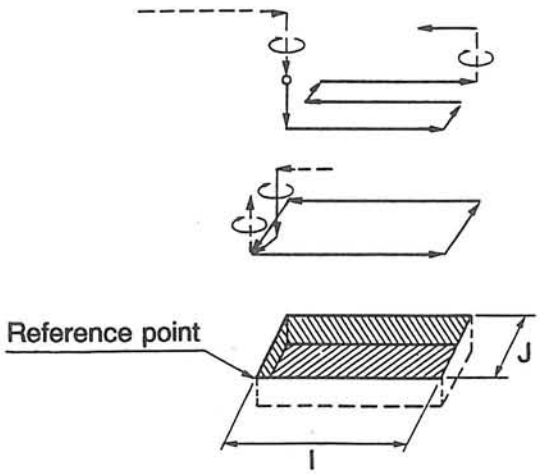
(b) Face Milling (FMILF)



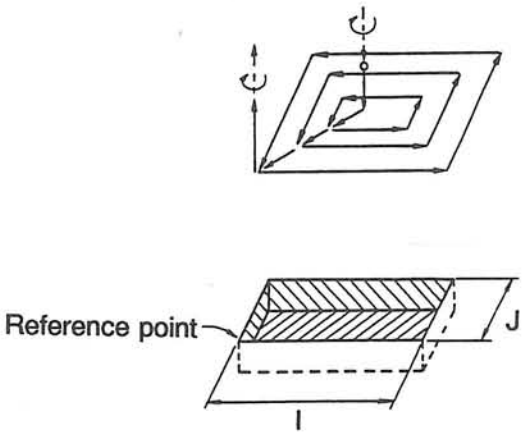
(c) Round Milling (RMILO)



(d) Round Milling (RMILI)



(e) Pocket Milling (PMIL)



(f) Pocket Milling (PMILR)

3. Area Machining Plane and Infeed Axis

(1) The area machining plane is selected with a plane selection command G17, G18, or G19. Operations 1 and 4, defined in Section 2, are performed on this plane.

(2) The infeed axis is perpendicular to the area machining plane.

The table below lists the possible area machining planes and their corresponding infeed axis.

Plane Selection Code	Area Machining Plane	Infeed Axis	Axes Composing Plane	
			Horizontal Axis	Vertical Axis
G17	Xp-Yp plane	Zp	Xp	Yp
G18	Zp-Xp plane	Yp	Zp	Xp
G19	Yp-Zp plane	Xp	Yp	Zp

Xp: X- or U-axis
Yp: Y- or V-axis
Zp: Z- or W-axis

(3) The plane selection code automatically determines two infeed axes; the area machining command block contains an address for the finishing surface level which determines the actual infeed axis.

4. General Rules

The following explanations assume that the area machining plane is the XY plane and that the infeed axis is the Z-axis. The explanation is similar for the other planes.

4-1. General Command Format

[(Mnemonic code) Xp__ Yp__ Zp__ I__ J__ K__ P__ Q__ R__ D__ F__ FA=__ FB=__]

Mnemonic code: Area machining function code

Xp : X-coordinate of reference point (Horizontal axis coordinate value)

Yp : Y-coordinate of reference point (Vertical axis coordinate value)

Zp : Z position level of finished surface (Cycle axis coordinate value)

I : Length from reference point in X direction (Horizontal length from reference point)

J : Length from reference point in Y direction (Vertical length from reference point)

K : Finish allowance

P : Percentage of cutting width

Q : Depth of cut of each cycle

R : Rapid retraction position (for cycle axis)

D : Cutter radius compensation

F, FA, FB : Feedrate

(1) Notes:

(a) Xp, Yp, and Zp represent the X-axis or U-axis, the Y-axis or V-axis, and the Z-axis or W-axis, respectively.

(b) The Xp and Yp addresses should designate the axes which define the area machining plane.

(c) The Zp address should designate the infeed axis, perpendicular to the plane defined by Xp and Yp.

(d) FA and FB are used in the round milling and pocket milling cycles.

(e) The K value must be given in such a way so that the machining allowance in the roughing cycle is positive. The following requirements must be met:

During G90 mode: $|K| < |Z|$

During G91 mode: $|K| < |R - Z|$

4-2. Area Machining Functions and Commands to Be Used

Item / Address	Reference point		Finish surface position	Area definition		Finish allowance	Cutting width (%)	Depth of cut per cycle	Level R	Offset No.	Feedrate		
	Xp	Yp	Zp	I	J	K	P	Q	R	D	F	FA	FB
Face Milling	○	○	◎	◎	◎	△	△	△	◎	◎	◎		
Pocket Milling	○	○	◎	◎	◎	△	△	△	◎	◎	◎	△	△
Round Milling	○	○	◎	◎	◎	△	△	△	◎	◎	◎	△	

(1) Notes

(a) Addresses designated with a single circle (○) may be omitted. The default value is the current position.

(b) Addresses designated with a double circle (◎) must always be specified. If a value is not specified, an alarm will be activated. However, F may be omitted if it has been given in a previous block, since F values are saved.

(c) Addresses designated with a triangle (△) may be omitted. The default values are described in the following table.

ADDRESS	DESCRIPTION	DEFAULT	REMARKS
K	Finish allowance	0 : No finish allowance	
P	Cutting width ratio (%)	70: 70 percent	
Q	Infeed amount in single cut	A single cut reaches the finish surface on which only the finish allowance is left.	When K is also omitted, cutting is made up to the finishing level.
FA	Feedrate	4 × F	Applies only to pocket milling and round milling
FB	Feedrate	F/4	Applies only to pocket milling

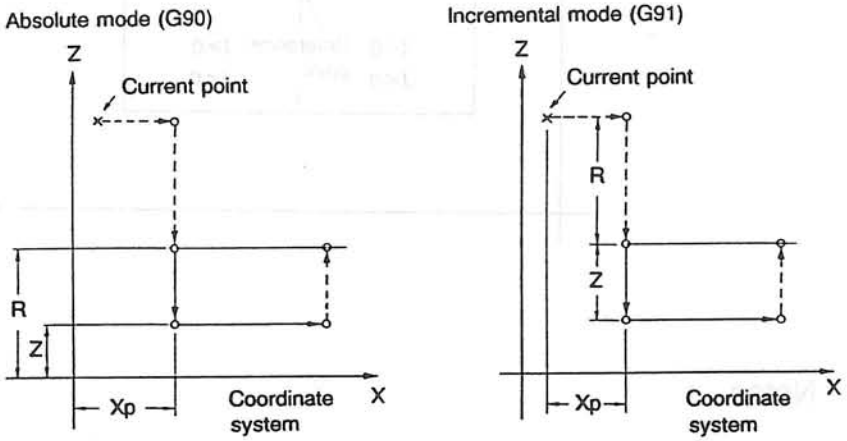
(d) The offset number, D, may not be given a value of zero (D00), or a number which calls for zero tool radius.

(e) The values of the cutting width, P, must be between 1 and 100. Any value outside this range will result in an alarm.

(f) Values which are given to addresses shown with blank spaces in the above table will be ignored.

4-3. Data Entry in Increment/Absolute Mode

(1) Four addresses for an area machining function must be specified depending on the selected dimensioning mode, incremental or absolute. They are: the coordinate values of the reference point (Xp, Yp), the finish surface position (Zp) and the rapid retraction level (R). The differences are illustrated in the figures below.

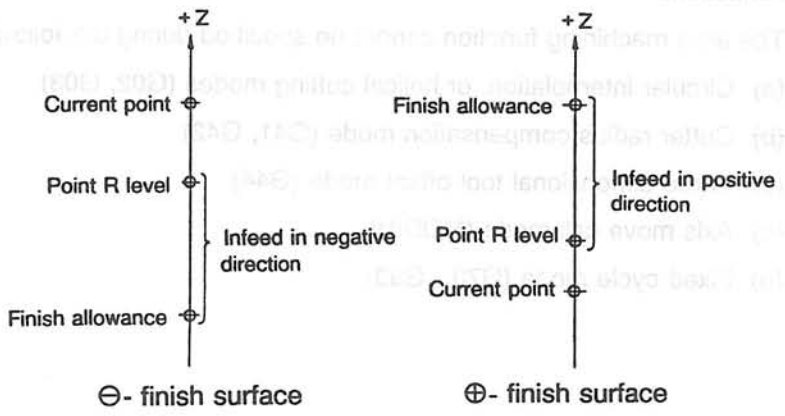


(2) During incremental mode, sign values preceding R and Z are given a meaning; the direction of axis infeed depends on these signs. In the illustration shown above, both R and Z have been assigned negative values.

4-4. Relationship between Present Point, Point R Level, and Finish Surface Level

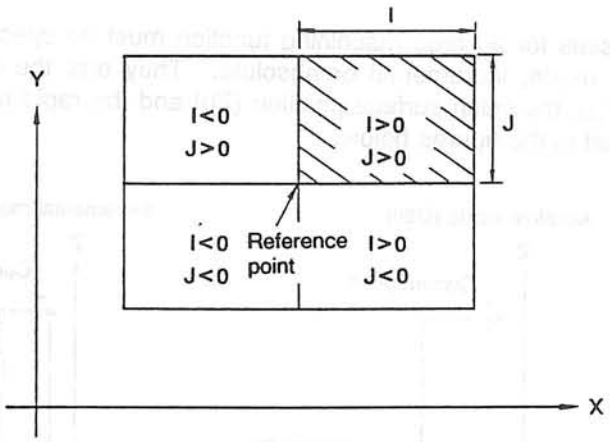
(1) The two possibilities for the positional relationship among the present point, point R level, and finish surface level are shown below.

(2) Also, the surface level on which the finish allowance is left must lie between the point R level and finish surface level.



4-5. Definition of Machining Area

The machining area is defined by I and J words. Shown below are the four different possible areas, which depend on the signs of the I and J words.
This definition is independent of the direction of the axis infeed as explained on 4-4. in this section.



4-6. Notes

(1) The first positioning point will be located near the specified reference point (Xp, Yp). The position varies depending on the specified area machining mode. Refer to 5. "Details of Area Cutting Functions" in the following page.

(2) The finish allowance may be identical for both the side and top surface.

(3) The spindle rotation direction must be in the forward direction (M03) before the area machining mode can begin. If the spindle is at a standstill or in the reverse rotation direction, the M03 code will be automatically generated and executed. The M03 mode remains effective even after the completion of the area machining function.

(4) The cutter diameter is calculated as twice the cutter radius compensation value. During the area machining mode, this value is used as the practical cutter diameter. Therefore, ~~great care should be taken when the compensation value stored in "Dnn" is different from the actual cutter radius.

(5) Limitations

The area machining function cannot be specified during the following modes:

(a) Circular interpolation, or helical cutting modes (G02, G03)

(b) Cutter radius compensation mode (G41, G42)

(c) Three-dimensional tool offset mode (G44)

(d) Axis move call mode (MODIN)

(e) Fixed cycle mode (G73 - G89)

5. Details of Area Cutting Functions

The following explanations assume that the area machining plane is the XY plane and that the infeed axis is the Z-axis. The explanation is similar for the other planes.

5-1. Face Milling Functions (FMILR, FMILF)

The face milling function uses the specified coordinate values as a reference point and cyclically machines the workpiece surface at a certain depth of cut (Q) over the range specified by the X and Y-axis lengths (I and J) until the final finish allowance (K) remains on the finish surface level (Z).

There are two types of face milling operations, depending on the tool movement:

FMILR in which the tool remains on the workpiece during operation

FMILF in which the tool moves off of the workpiece in between cutting strokes

Programming Format

〖 FMILF (FMILR) X±x Y±y Z±z I±dx J±dy Kfl P%
Qdp R±rz Dnn F_ 〗

X : X coordinate value (x) of reference point

Default value is the X coordinate of the current point.

Y : Y coordinate value (y) of reference point

Default value is the Y coordinate of the current point.

Z : Position of finish surface (z)

During G90 mode: Height from programming zero to finish surface
During G91 mode: Distance from point R level

I : Length of surface to be cut from reference point (x) along X-axis (dx)

J : Length of surface to be cut from reference point (y) along Y-axis (dy)

K : Finish allowance (f1)

Default value is 0.

P : Cutting width expressed in percentage

Ratio, in percent, of the cutting width to the cutter diameter. Although the ratio is a percentage, the percent symbol should not be included with the value.
Default value is 70 percent.
As will be explained later, the command value is slightly different from the actual cutting width.

Q : Depth of cut (dp)

Default value is the depth of a single cut which will reach the finish surface, including the finish allowance.

R : Rapid return level (rz)

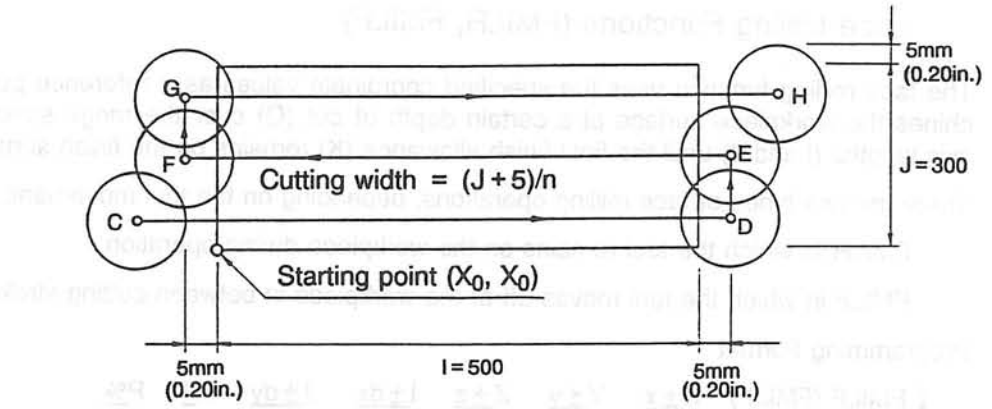
D : Cutter radius compensation number (nn)

F : Feedrate

(1) FMILR (Tool remains on workpiece)

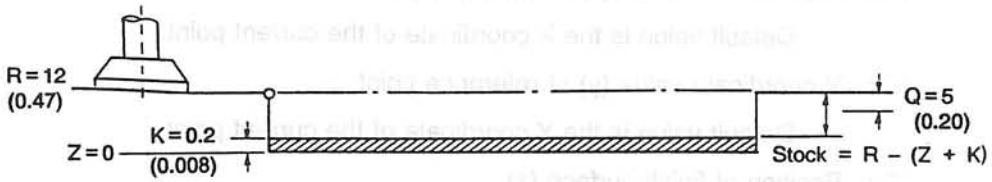
Example:

```
FMILR X0 Y0 Z0 I500 J300 K0.2 P70 Q5 R12 D01 F400
```



----→ Rapid feed P = 70% (cutting width ratio)

——→ Cutting feed D01 = 75 (cutter radius)



Cutter movements:

(a) The reference point is established at (X_0, Y_0).

(b) The cutting surface (rectangle) is defined by I500 and J300.

(c) The cutter has a radius specified by D01.

(d) The rapid traverse level is set by R12.

(e) The cutter is infed by the amount Q = 5 mm (0.20 in.).

(f) At this infed level, area machining is carried out with a cutting width ratio of (J + 5)/n at a feedrate of F400.

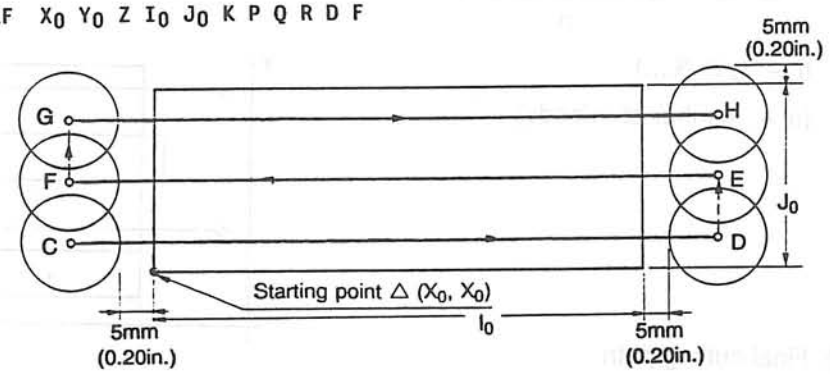
(g) After each cycle of surface machining, the cutter is infed by the amount Q = 5 mm (0.20 in.). Machining is repeated until the level of finish allowance, K = 0.2 mm (0.008 in.), is left. Details of axis movements will be described on later pages.

Note: *Stock = R – (Z + K)*

(2) FMILF (Tool moves off of workpiece)

Example

```
FMILF  X0 Y0 Z I0 J0 K P Q R D F
```



During FMILF mode, although the Z-axis moves in the same way as during the FMILR mode, on the machining surface the cutter is fed to the point where it is off of the workpiece before it is positioned to the next coordinate point.

(3) Number of Infeeds

The number of times an infeed is made is determined by the following equation:

$$n = \text{Fup}\left(\frac{\text{shorter side length} + 5\text{mm (0.20 in.)}}{\text{cutter radius compensation amount} \times 2 \times \frac{P}{100}}\right)$$ (Fup: Fractions are rounded up.)

D indicates the cutter radius compensation value. The calculation is made based on twice the value registered in the corresponding D value.

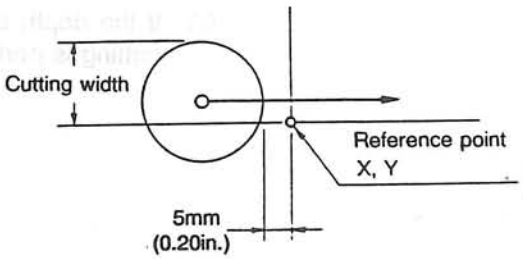
(4) Cutter Positioning

(a) First positioning

① The cutter is positioned so that a specified cutting width* is obtained in the narrower direction

* The cutting width is equal to (J + 5)/n in millimeters. (J: shorter side)

② there is a distance of 5 mm (0.20 in.) from the workpiece in the wider work direction.



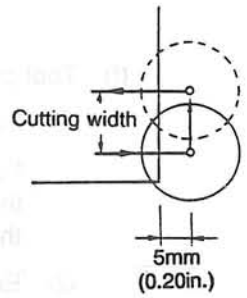
(b) Longitudinal cutting path

① FMILR (ON workpiece)

Machining continues until the center of the cutter is 5 mm (0.20 in.) away from the workpiece.

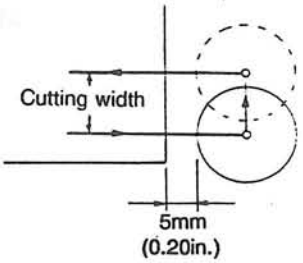
② FMILF (OFF workpiece)

Machining continues until the outside diameter of the cutter is 5 mm (0.20 in.) away from the workpiece.



③ I = J

For both the FMILR and the FMILF function, cutting is performed in the X direction when I equals J.

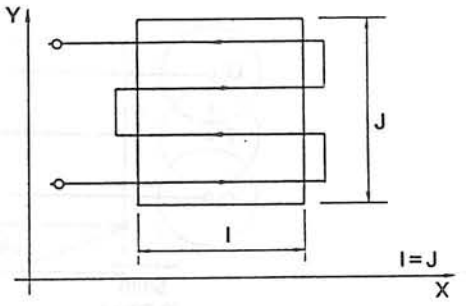


(c) Cutting position in workpiece narrower position from reference point

$$\frac{\text{shorter side length} + 5\text{ mm (0.20 in.)}}{n} \times i - \text{Cutter radius compensation amount}$$

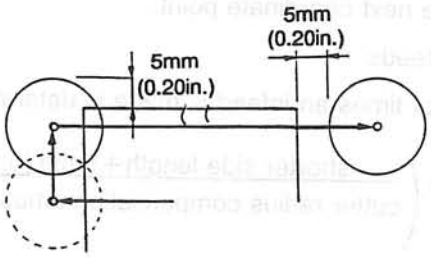
(i = 1, 2, 3,...)

(n = number of infeeds)



(d) Final cutting path

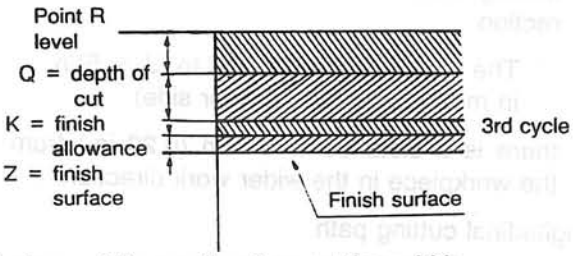
The cutter is positioned so that the outside diameter of the tool projects 5 mm (0.20 in.) from the workpiece. Machining continues until the tool OD is 5 mm (0.20 in.) away from the workpiece.



(e) Depth of cut in the Z-axis direction

① Starting at the specified point R level, machining continues (repeated infeeding of depth (Q) and surface cutting operations) until the finish allowance (K) remains on the finish surface (Z).

② If the depth of the cut (Q) is greater than the stock to be removed (R – (Z + K)), the cutting is performed in one infeed.



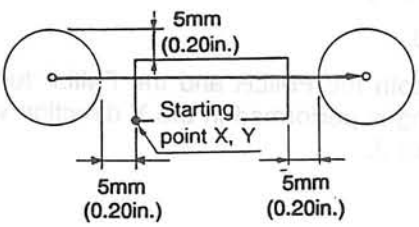
(f) Tool path with workpiece width smaller than cutting width

① Positioning point:

Positioning is made so that the cutter OD projects 5 mm (0.20 in.) from the workpiece in the narrower workpiece direction and so that the cutter OD is 5 mm (0.20 in.) away from the workpiece in the wider workpiece direction.

② End-of-machining point:

Machining continues until the cutter OD is 5 mm (0.20 in.) away from the workpiece.

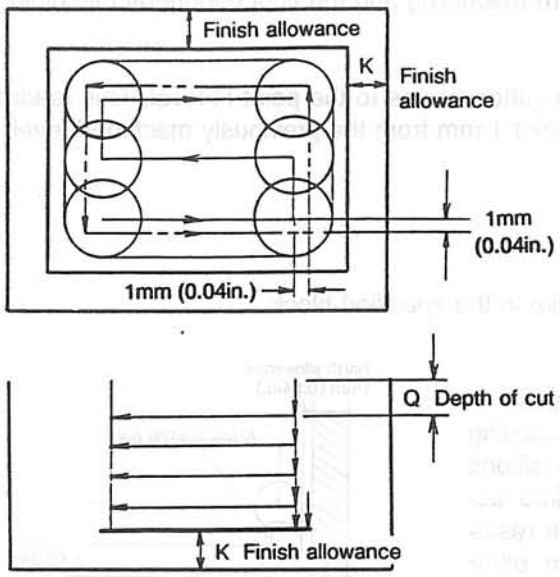


5-2. Pocket Milling

The pocket milling function may be classified into two types: zigzag (PMIL) and spiral (PMILR). Description of these functions is given on the following page.

5-2-1. Pocket Milling Zigzag Function (PMIL)

The pocket milling function, PMIL, uses the specified coordinate values as a reference point and cyclically machines a rectangle (hole) specified by the X and Y-axis lengths (I and J) in the workpiece surface in a zigzag pattern at a certain depth of cut (Q) until the final finish allowance (K) remains on the finish surface with height (Z). The finish allowance (K) also remains in the X- and Y-directions.



(a) The cutter is infed by the specified depth of cut, Q.

(b) The cutter moves inside the specified rectangle in a zigzag pattern.

(c) Steps (a) and (b) are repeated until only the finish allowance remains on the finish surface in the Z-direction. Then, a rectangle 1 mm (0.04 in.) larger than the rectangle made in step 2) is made.

Programming Format

【 PMIL X__ Y__ Z__ I__ J__ K__ P__ Q__ R__ D__ F__ FA=__ FB=__ 】

X : X coordinate value of starting point

Default value is the current point.

Y : Y coordinate value of starting point

Default value is the current point.

Z : Z coordinate value of finish surface

During G90 mode: Height from the selected coordinate system to the bottom of the hole
During G91 mode: Distance from the point R level

I : Length of hole from the starting point in the X-axis direction

J : Length of hole from the starting point in the Y-axis direction

K : Finish allowance in each of the X, Y, and Z directions

Default value is 0.

P : Percentage of cutting width

This is the ratio of the cutting width to the cutter diameter. Although the value is a percentage, the percent symbol is not included in the command.
Default value is 70 percent.

Q : Depth of cut in one cycle

Default value is a single cut that reaches the finish surface with the finish allowance remaining.

R : Z coordinate value of the level to which positioning is made during rapid feed or rapid return

D : Cutter radius compensation number

F : Cutting feedrate

This feedrate is used for the zigzag pattern machining and the final peripheral machining.

FA : Feedrate after point R level

After each cycle of zigzag machining, the cutter returns to the point R level. This feedrate is used between the point R level and the point 1 mm from the previously machined level.

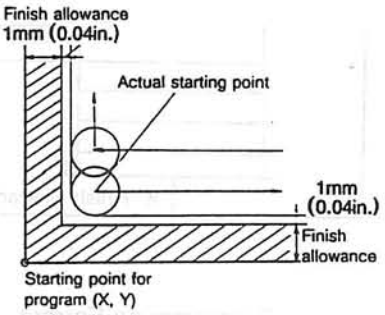
Default value is FA = 4*F

FB : Feedrate for Z-axis infeed

Default value is FB = F/4
Note that FA and FB are both only effective in the specified block.

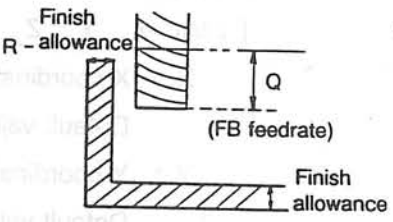
(1) Machining Sequence

(a) The cutter is positioned to the starting point in the XY plane. Using this programmed starting point as a reference point, the system positions the cutter inside a square which takes into account the final finish allowance, K, and the residual finish for both the X and Y-axis. In other words, the cutter is placed at a position located inward from the programmed starting point by the amount of the finish allowance, K, plus 1 mm in both the X and Y directions. Note that the residual finish here is the amount of residual portion to be removed. This amount is fixed to 1 mm (0.04 in.), as shown.

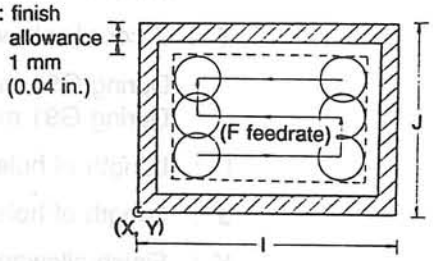


(b) The infeed axis Z is positioned to the commanded point R level at a rapid feedrate.

(c) Starting at the point R level, infeeding is made to the commanded depth of cut, Q, in the Z-axis direction at the feedrate programmed by FB.



(d) The inner rectangle is cyclically machined in a zigzag pattern. The F command specifies the feedrate to be used during this cycle of machining. Note that the cutting width is different from the commanded value as indicated below:

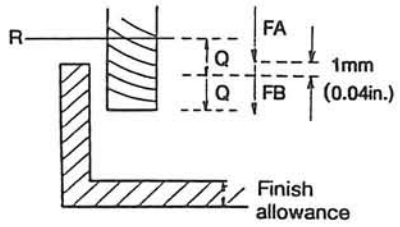


$$\frac{\text{shorter side length} - 2\,(K + \text{cutter radius compensaiton amount} + 1\text{ mm})}{n-1}$$

where n is the number of cuts which is found by rounding up the decimal part of:

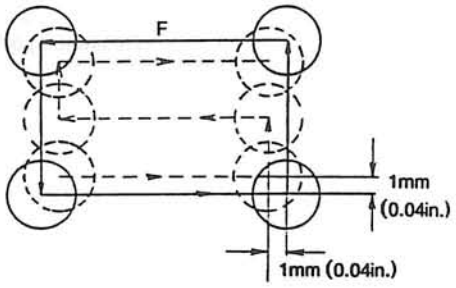
$$n = \left(\frac{\text{shorter side length} - 2\ (\text{K} + \text{cutter radius compensation amount} + 1\ \text{mm})}{\text{cutter radius compensation amount} \times 2 \times \frac{P}{100}} + 1 \right)$$

(e) The cutter returns to the initial positioning point (X,Y,R) at a rapid feedrate. It is then positioned from the point R level to a point 1 mm (0.04 in.) towards the surface level machined in the previous machining cycle. Then, the cutter is infed by the amount Q + 1 mm (0.04 in.). The next machining cycle is performed in a zigzag pattern at a feedrate of F.



(f) Step (e) above is repeated until the final finish allowance remains on the finish surface. Finally, a rectangle 1 mm (0.04 in.) greater than the zigzag-machined rectangle is machined. The feedrate in the final cycle is specified by the F command.

(g) After completion of the cutting cycle, the axes are positioned at a point 5 mm (0.02 in.) away from the workpiece in each axis direction.

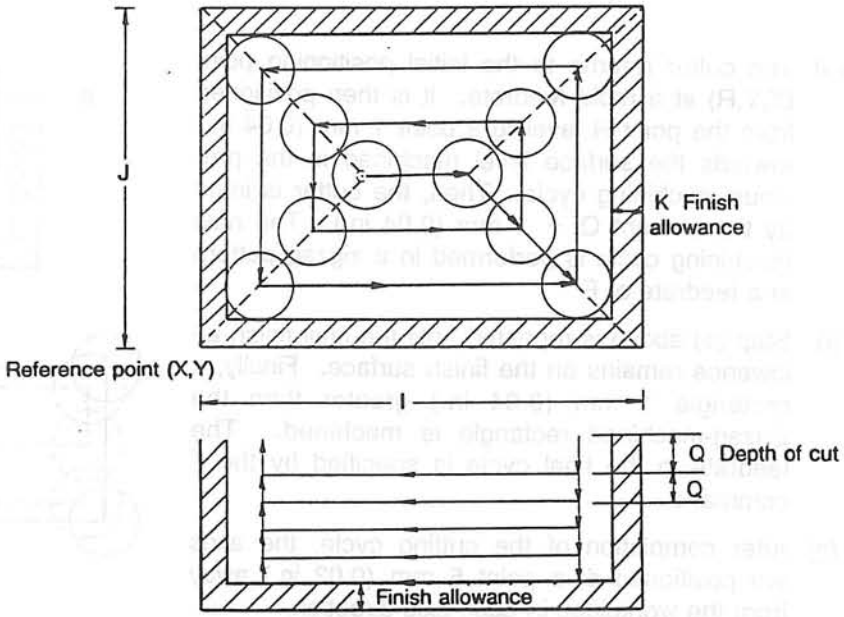


Note: *Before the pocket milling operation in the PMIL mode can be started, the relationship between the workpiece shape and cutter diameter is checked to determine if the programmed cutting is possible. The following inequality must be satisfied, otherwise alarm B "Area machining: area command" will be activated.*

Shorter side length – (1 mm + finish allowance + cutter radius) x 2 > 5 mm

5-2-2. Pocket Milling Spiral Function (PMILR)

The pocket milling function, PMILR, uses the specified coordinate values as a reference point and cyclically machines a rectangle (hole) specified by the X and Y-axis lengths (I and J) in the workpiece surface in a spiral pattern at a certain depth of cut (Q) until the final finish allowance (K) remains on the finish surface with height (Z). The finish allowance (K) also remains in the X and Y-directions.



① The cutter is infed by the specified depth of cut, Q.

② The cutter machines the inside of the specified rectangle in a spiral pattern, leaving the finish allowance K.

③ Steps ① and ② are repeated until only the finish allowance, K, remains on the finish surface in the Z-direction.

Programming Format

【 PMILR X__ Y__ Z__ I__ J__ K__ P__ Q__ R__ D__ F__ FA=__ FB=__ 】

X : X coordinate value of reference point

Default value is the current point.

Y : Y coordinate value of reference point

Default value is the current point.

Z : Z coordinate value of finish surface

During G90 mode: Height from the selected coordinate system to the bottom of the hole
During G91 mode: Distance from the point R level

I : Length of hole from the reference point in the X-axis direction

J : Length of hole from the reference point in the Y-axis direction

K : Finish allowance in each of the X, Y, and Z directions

Default value is 0.

P : Percentage of cutting width

This is the ratio of the cutting width to the cutter diameter. Although the value is a percentage, the percent symbol is not included in the command.Default value is 70 percent.

Q : Depth of cut in one cycle

Default value is a single cut that reaches the finish surface with the finish allowance remaining.

R : Z coordinate value of the level to which positioning is made during rapid feed or rapid return

D : Cutter radius compensation number

F : Cutting feedrate

This feedrate is used for the spiral pattern machining.

FA : Feedrate after point R level

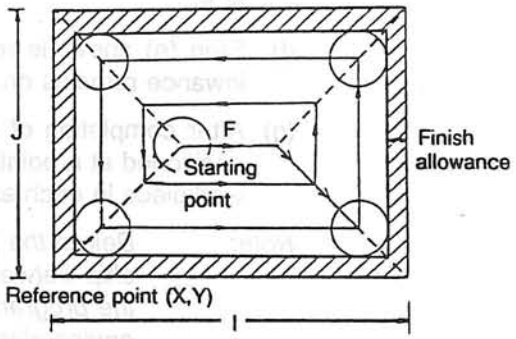
After each cycle of spiral machining, the cutter returns to the point R level. This feedrate is used between the point R level and the point 1 mm (0.04 in.) from the previously machined level.
Default value is FA = 4*F

FB : Feedrate for Z-axis infeed

Default value is FB = F/4
Note that FA and FB are both only effective in the specified block.

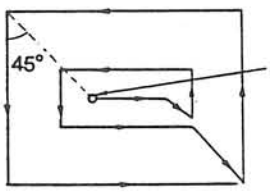
(1) Machining Sequence

(a) The cutter is positioned to the starting point in the XY plane.
Using this programmed starting point as a reference point, the system determines a rectangle from the lengths specified by the I and J words. In the rectangle, the system determines the starting point from the cutting width at the inside from which the finish allowance, K, is removed in both the X and Y directions. The cutter is positioned at that starting point.



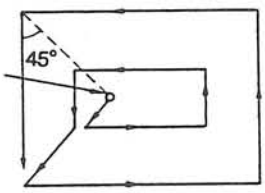
The starting point is always taken as the point nearer the commanded reference point as shown below:

Infeed is made an odd-number of times



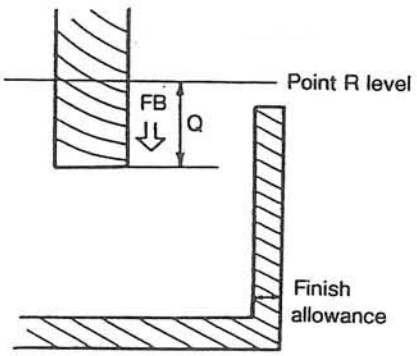
Starting point

Infeed is made an even-number of times



(b) The infeed axis Z is positioned to the commanded point R level at a rapid feedrate.

(c) Starting at the point R level, infeeding is made to the commanded depth of cut, Q, in the Z-axis direction at the feedrate programmed by FB.



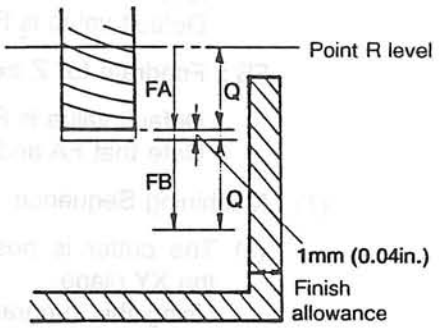
(d) The inner rectangle is cyclically machined in a spiral pattern. The F command specifies the feedrate to be used during this cycle of machining. Note that the cutting width is different from the commanded value as indicated below:

$$\frac{\text{shorter side length} - 2\,(K + \text{cutter radius compensaiton amount} + 1\text{ mm})}{n-1}$$

where n is the number of cuts which is found by rounding up the decimal part of:

$$n = \left(\frac{\text{shorter side length} - 2\,(K + \text{cutter radius compensation amount} + 1\text{ mm})}{\text{cutter radius compensation amount} \times 2 \times \frac{P}{100}} + 1 \right)$$

(e) The cutter returns to the initial positioning point (X,Y,R) at a rapid feedrate. It is then positioned from the point R level to a point 1 mm (0.04 in.) towards the surface level machined in the previous machining cycle. Then, the cutter is infed by the amount Q + 1 mm (0.04 in.). The next machining cycle is performed in a spiral pattern at a feedrate of F.



(f) Step (e) above is repeated until the final finish allowance remains on the finish surface.

(g) After completion of the cutting cycle, the axes are positioned at a point 5 mm (0.20 in.) away from the workpiece in each axis direction.

Note: *Before the pocket milling operation in the PMILR mode can be started, the relationship between the workpiece shape and cutter diameter is checked to determine if the programmed cutting is possible. The following inequality must be satisfied, otherwise alarm B 514 "Area machining: area command" will be activated.*

Shorter side length – (finish allowance + cutter radius) x 2 > 5 mm

5-3. Round Milling Functions (RMILO, RMILI)

The round milling functions, RMILO and RMILI, uses the specified coordinate values as a reference point and cyclically machines around a rectangle which has a machining allowance (Q) on its circumference, specified by the X and Y-axis lengths (I and J), until the final finish allowance (K) remains on the finish bottom surface level (Z).

The round milling function may be classified into two types depending on the machining:

RMILO . . . external machining of rectangles

RMILI . . . . internal machining of rectangles

Programming format:

[RMILO (RMILI) X±x Y±y Z±z I±dx J±dy Kf1
P% Qdp R±rz Dnn F_ FA=_]

X : X coordinate value (x) of reference point

Default value is the current point.

Y : Y coordinate value (y) of reference point

Default value is the current point.

Z : Position of finish bottom surface (z)

During G90 mode: Height from program origin to finish bottom surface
During G91 mode: Distance from the point R level

I : Length (dx) from the reference point (x) along the X-axis

J : Length (dy) from the reference point (y) along the Y-axis

K : Finish allowance (fl)

Default value is 0.

P : Percentage of cutting width

This is the ratio of the cutting width to the cutter diameter. Although the value is a percentage, the percent symbol is not included in the command.
Default value is 70 percent.

Q : Machining allowance (dp)

Default value is a single cut that reaches the finish surface with the finish allowance remaining.

R : Z coordinate value (rz) of the level to which positioning is made during rapid return

D : Cutter radius compensation number (nn)

F : Cutting feedrate

This feedrate is used for machining the external or internal periphery.

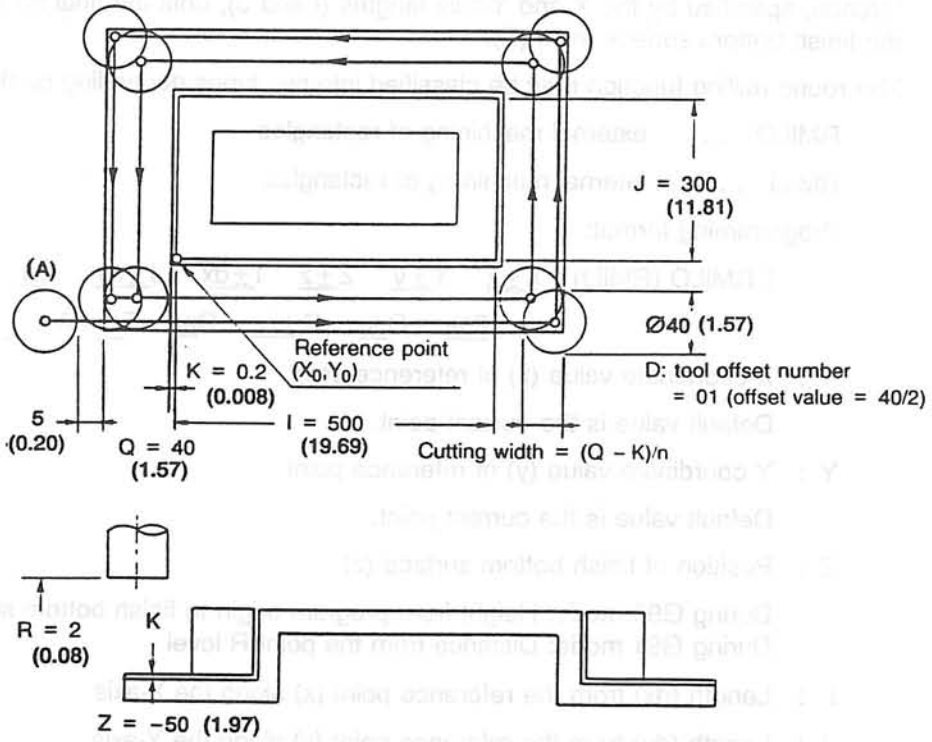
FA : Feedrate after point R level

This feedrate is used for infeeding from the point R level to the finish surface with the finish allowance, K, remaining.
Default value is FA = 4 x F

(1) RMILO - External Cutting

Example:

```
RMILO X0 Y0 Z-50 I500 J300 K0.2 P70 Q40 R2 D01 F400 FA=800
```



(a) The reference point is taken as X_0, Y_0.

(b) Machining is performed outside the rectangle defined by I500 and J300.

(c) A cutter with a specified diameter of D01 is used.

(d) The rapid positioning level is R2 (2 mm (0.078 in.)).

(e) The machining allowance is Q (40 mm (1.57 in.)).

(f) The cycle machining level is "finish level (Z)" + "finishing allowance (K)".
For this example: –50 mm + 0.2 mm.

(g) The cutter is infed by the amount (Q – K)/n from the end face of the blank workpiece. The cycle machining is performed at a feedrate F.

(h) Cycle machining is repeated until the specified finish allowance remains on the surface.

Note 1: *The first positioning point (A) is the point where the cutter OD is 5 mm (0.20 in.) away in the longitudinal direction. The cutter is infed with the programmed cutting width from the blank in the crosswise direction.*

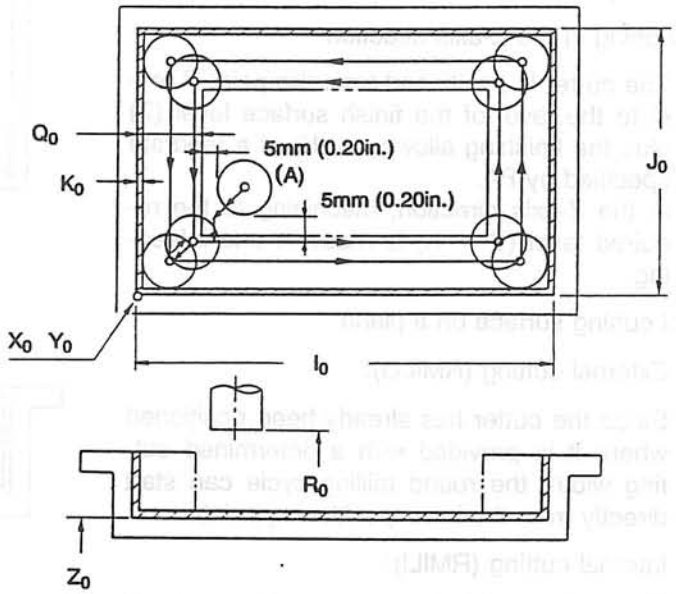
Note 2: *Before the round milling operation in the RMILO mode can be started, the relationship between the finish allowance and the stock removal is checked to determine if the programmed cutting is possible. The following inequality must be satisfied, otherwise alarm B519 "Area machining: large finish allowance" will be activated.*

$$Q \geqq K$$

(2) RMILI - Internal Cutting

Example:

```
RMILI X0 Y0 Z0 J0 K0 PQ0 R0 D F FA CR/LF
```



The first positioning point (A) is located where the cutter OD is 5 mm (0.20 in.) away from the blank end face.

Notes: *Before the round milling operation in the RMILI mode can be started, the relationship between the workpiece shape and machining command is checked to determine if the programmed cutting is possible.*

The following inequality must be satisfied, otherwise alarm B514 "Area machining: area command" will be activated.

(Cutter radius compensation value + Q + 5 mm) × 2 < shorter side length

The following inequality must be satisfied, otherwise alarm B519 "Area machining: large finish allowance" will be activated.

$Q \geqq K$

(3) Number of Cutting Cycles

The number of cutting cycles (infeeds) can be determined by the following equation:

n = Fup

(Fup indicates that the fractions should be rounded up).

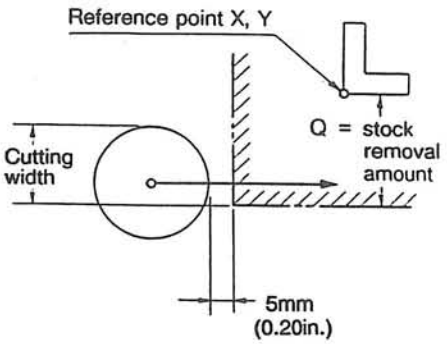
D designates the cutter radius compensation value. The above calculation is based on twice the value given in the D address.

(4) Cutter Positioning

(a) First positioning

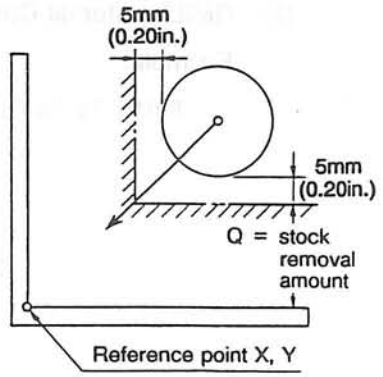
① External cutting (RMILO):

The cutter is positioned to a point where the cutting width (cutter diameter × P) from the workpiece end surface in the narrower workpiece direction can be assured. In addition, the point must be located such that the cutter OD is 5 mm (0.20 in.) away from the workpiece.



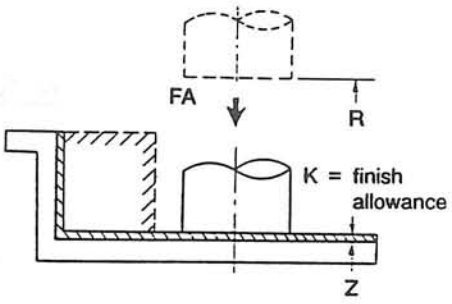
② Internal cutting (RMILI);

The cutter is positioned to a point where the cutter OD is 5 mm (0.20 in.) away from the workpiece both in the X and Y directions.



(b) Positioning in the Z-axis direction

The cutter is positioned from the point R level to the level of the finish surface level (Z) plus the finishing allowance (K) at a feedrate specified by FA.
In the Z-axis direction, machining to the required level (Z + K) is made in one infeeding.



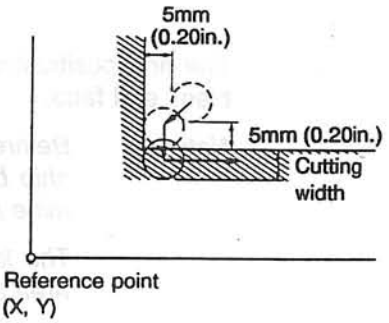
(c) Initial cutting surface on a plane

① External cutting (RMILO):

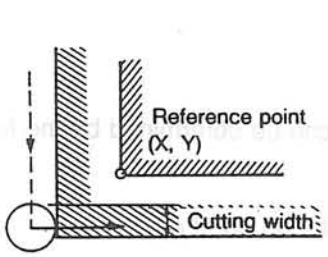
Since the cutter has already been positioned where it is provided with a determined cutting width, the round milling cycle can start directly from the initial positioning point.

② Internal cutting (RMILI):

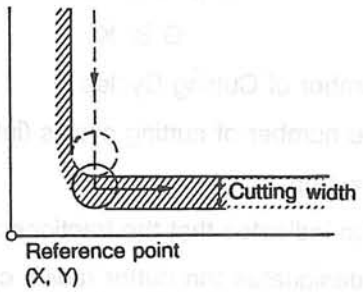
The cutter is infed an amount 5 mm (0.20 in.) from the positioning point to the starting point. The cutter is further infed by the cutter width in the workpiece narrower direction. The round milling cycle then starts from that point.



(d) Cutting surface on a plane from second time onward



RMILO

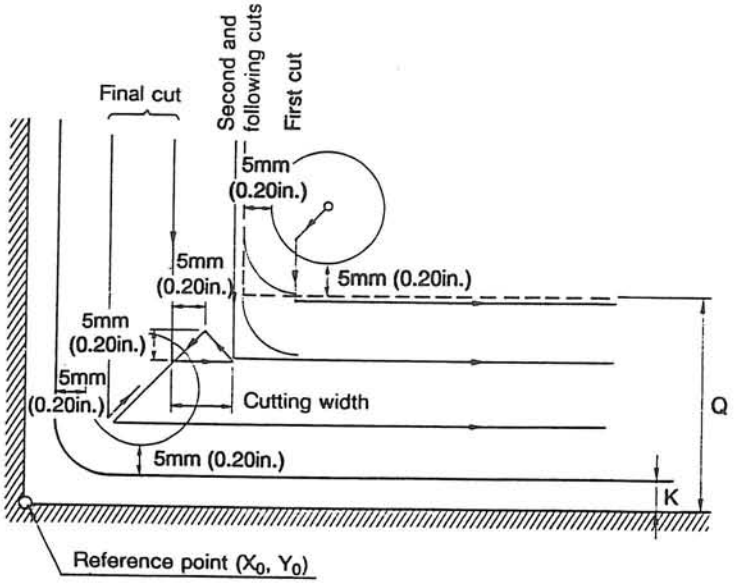


RMILI

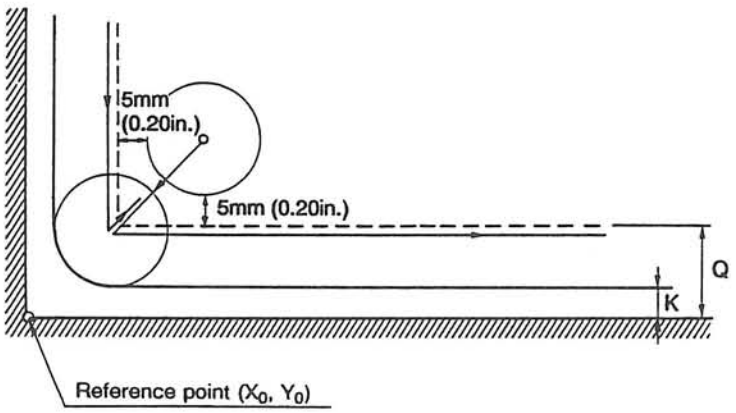
(e) Final infeed

The final infeed applies only to the RMILI function.

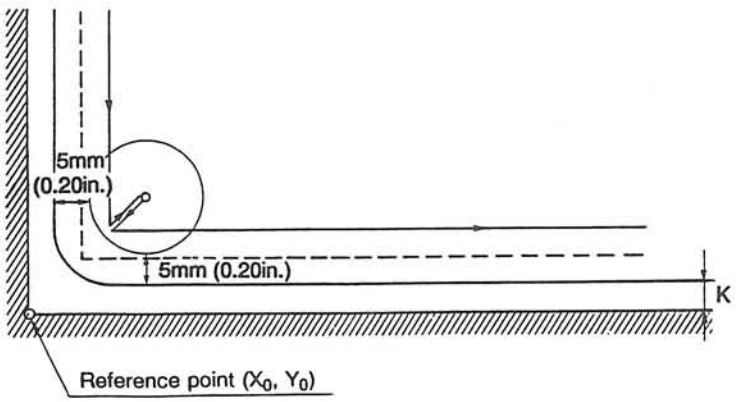
① Machining with more than one cut



② Machining with a single cut



③ Without Q (stock) command



(f) Retraction from workpiece

During the RMILI (internal cutting) mode, the cutter retracts inwardly from the workpiece as it is in contact with the workpiece at the end of the cycle. Retraction amount is 5 mm (0.20 in.) along both axes.
During the RMILO (external cutting) mode, the cutter does not retract.

SECTION 9 S, T, AND M FUNCTIONS

The S, T, and M functions specify the necessary machine operations other than axis movement commands.

S: Spindle speed

T: Tool number for tool change cycle

M: Turning solenoids and other similar devices on and off

For each one of these types of codes, only one code may be given in one block.

As many of the following M codes as needed may be specified in one block.

M15, M16	Rotary index table CW/CCW
M52, M53, M54	Fixed cycle; return level designating upper limit/specified point level/point R level
M115, M116	Second rotary index table CW/CCW
M130, M131	Cutting feed; spindle rotating condition OFF/ON
M132, M133	Single block ineffective/effective
M136, M137	Feedrate override ineffective/effective
M138, M139	Dry run ineffective/effective
M140, M141	Slide hold ineffective/effective

1. S Functions

(1) The S function specifies the spindle speed.

(a) For a variable-speed motor:

S and a 4-digit number, which together directly specify the rotation speed (rpm).

(b) For a constant speed AC motor:

S and a 2-digit number, which together specify the gear selection

(2) If there is an S command and an axis move command in the same block, the S function will become effective at the same time the axis starts to move.

(3) The S command will not be canceled when the NC is reset, however, it will be set to 0 when the power supply is turned off.

(4) The S command should be given before the spindle rotation command (M03 or M04) is given.

2. T Functions

(1) The T function specifies the number of the tool to be replaced.

(2) The programmed T code command prepares the tool to be set in the spindle. The next tool is indexed or taken from the magazine post and set in the ready station.

(3) The M06 code programmed after a T code initiates a tool exchange cycle in which the selected tool is directly exchanged with the tool presently in the spindle (active tool).

(4) If a T code is programmed in a block which also contains axis motion commands, the T code will be executed simultaneously with the axis motion commands.

3. M Functions

(1) The M function specifies that solenoids and other devices be turned on and off.

(2) The following table lists all of the M codes and their functions.
If an M code and axis movement commands are programmed in the same block, the M function either is executed at the same time as the axis movement or after the axis movement has been completed. This characteristic is defined for each M code in the fourth column of the following table.

(3) There are two types of M functions: modal and one-shot. A modal M code remains active until changed by another M code of the same group. A one-shot M code is active only in the programmed block. This characteristic is defined for each M code in the fifth column of the following table.

Note: Refer to Section 14, 4 for the M code list.

SECTION 10 SUBPROGRAM FUNCTIONS

(1) Programming sometimes repeats similar patterns or uses patterns from other programs. Instead of continually rewriting any repeatedly used programs, correct and rapid programming may be implemented by the subprogram function. In this way, the frequent patterns are stored as a subprogram and called as necessary.

(2) A subprogram can be called not only from the main program but also from another subprogram, up to eight-subprogram nesting is possible.

(3) Three different subprogram modes are available, as listed below. Note that a call is an independent command and it would be ineffective to give the same block an axis move command.

(a) Simple Call (CALL)

(b) Call After Axis Move (MODIN, MODOUT)

(c) G/M Code Macros

① Simple calls (G111 - G120, M201 - M210)

② Call after axis move (G100 - G110)

③ Maker macro calls (for simple calls only)

(4) Subprograms to be called may be grouped into three types:

Subprogram	File Device	File Extension	Usable Subprogram
User subprogram (system subprogram)	–	· SUB	O*** ↑ Other than O
	BB1:	· SSB	
Maker subprogram	BB0:	· MSB	OO***

***: numerals

(5) When a subprogram call command is issued, when the DIR function key is pressed in the EDIT/AUX mode, the search for the specified subprogram is made among the subprogram files displayed. However, if there is a subprogram connected to the main program, then the search is made from that subprogram.

(a) User Subprograms

An attempt is made to search for the file specified by the PSELECT command and, if that subprogram is not found, to search all the files whose device name is BB1: with extension .SSB.

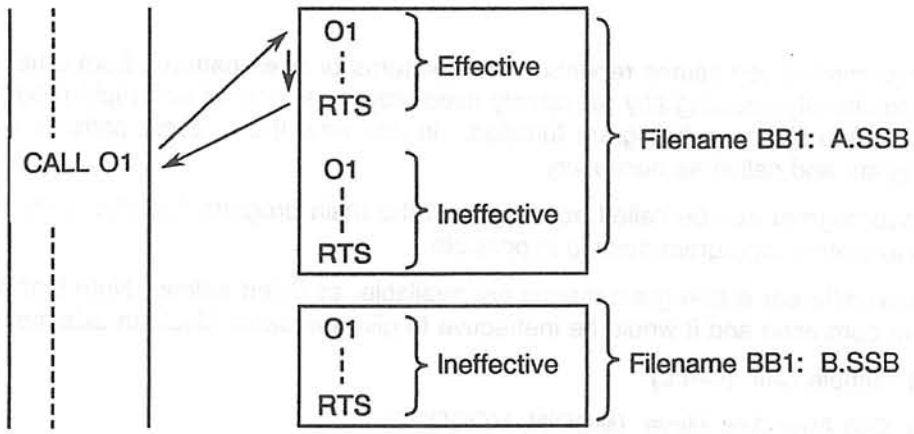
(b) Maker Subprogram

An attempt is made to search for all the files whose device name is BB1: with extension .MSB.

(6) In case the subprogram file includes more than one program of the same name, then only the first located is effective.

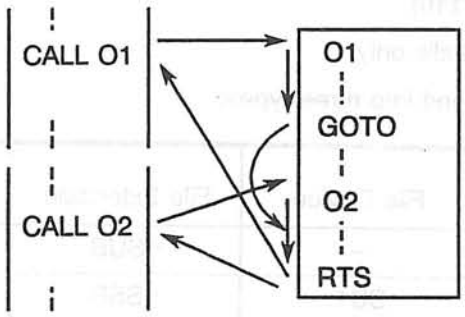
Example 1:

P-SELECT command where the subprogram file is not specified:

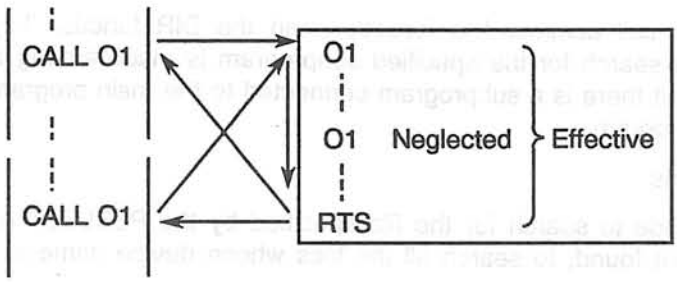


Example 2:

Program having a program name:



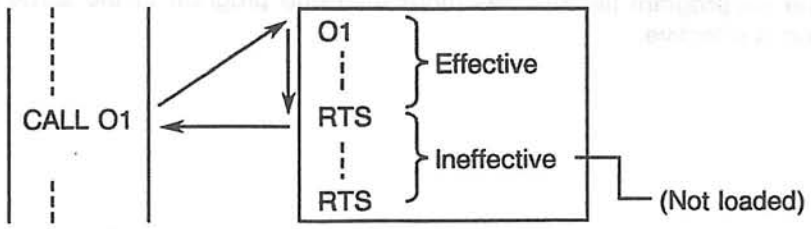
If a program name differing from the program called is present, use a GOTO statement to skip over it.



If the identical program name is used more than once, only the first program name is effective.

Example 3:

Program having RTS in it:



(7) Notes

(a) The total number of subprograms used or called in one program is 63.

(b) In the block which contains a CALL command, only a program name, (an optional block skip command) and/or a sequence name, may be programmed before the CALL command. If other commands are written before it, an alarm will occur.

N10×100Y200F300CALL O1
Alarm

Commands written after the program name, preceding the CALL command and up to the end of the block code in that block, are ignored.

N10CALLO1×100Y200F300
Ignored

(c) In operation method A (program selection operation with ordinary storage capacity memory), the total tape length for the schedule, main and subprograms are up to approximately 160 meters (48.78 ft.).

(d) In operation method B (program selection operation with large storage capacity memory), the total tape length for only the subprogram is approximately 100 meters (30.49 ft.) standard. This capacity can be increased when optional buffer expansion specifications are requested. With S specified in the PSELECT command option, an alarm will occur if a subprogram call (CALL; after axis move; G code macro) is issued. A library-registered subprogram is usable when option S is specified.

(e) Additionally, when a subprogram call command is issued by MDI, an alarm will occur if a call after the axis move command (MODIN, MODOUT or G100 - G110) is issued. However, simple calls may be carried out if the subroutine name, given by the current PSELECT command or by library registration, is specified.

1. Simple Call

This is function executes a subprogram in response to a call command.

Programming format:

【 CALL O__ Q__ Variable-setting 】

(The commands must be programmed in this order)

O: Program name to be called

Q: Number of repetitions (9999 or less)
Default is 1

Variable-setting: Variable = expression, variable = expression,..

(1) Here is the list of data and variable to be exchanged between programs.

(2) The left side members are the variable of the subprogram called, including the local variable, common variable, system variable, and the input/output variable.

(3) The right side members are the variables or data of the program to be called.

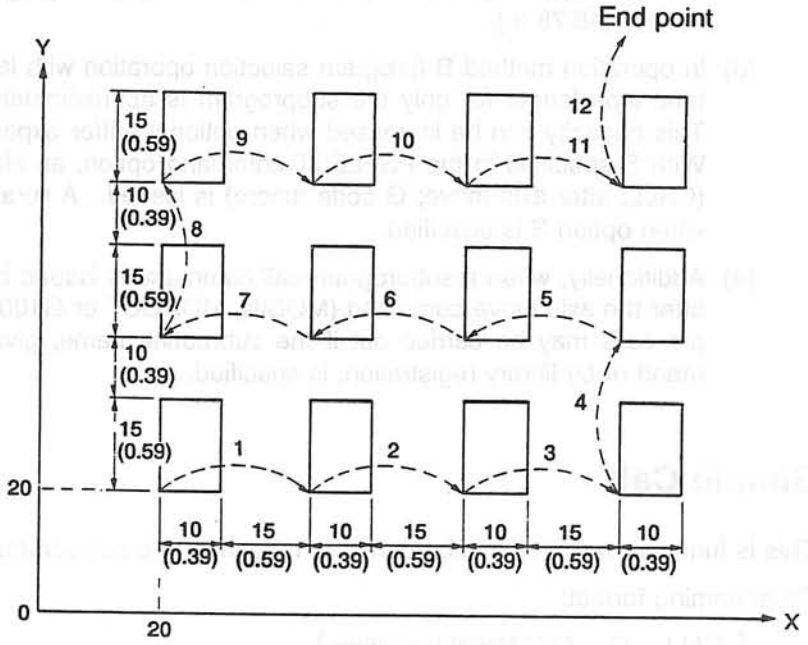
Example:

In the following case, numerical values of LB, LC, and LD are:

```
LB = 10, LC = 10, LD = 20
O1
N1  LA=10
N2  CALL O2  LA=20  LB=LA
N3  LC-LA
M02
O2                                    —— : Local variable of O1
N1  LD=LA                             ═══ : Local variable of O2
RTS
```

When the number of subprogram repetitions, specified by a Q word, is two or more, the present argument is not reset and is executed as is.

Example:



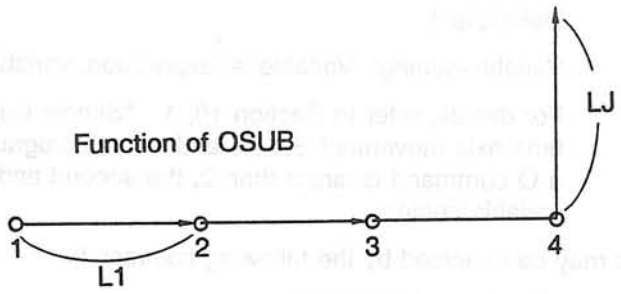
Main program:

```
O1
N1  G90 G00 X20 Y20
N2  CALL  OSUB  Q3 LX=10  LI=25  LP=4
&   LY=15  LJ=25  LZ=50
M02
```

Subprogram (Positioning):

```
OSUB
N11  LC=LP
N12  CALL  OSQR  LX=LX  LY=LY  LZ=LZ
N13  LC=LC-1
N14  IF  (LC LE 0)  N17
N15  G91G00X=LI .......  X-directional positioning
N16  GOTO  N12
N17  G91G00Y=LJ .......  Y-directional positioning
N18  LI=-LI ..........  Direction reversed on X-axis
RTS
```

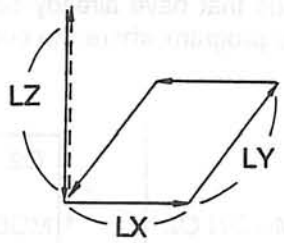
Function of OSUB



Subprogram (Cutting):

```
OSQR
N21  G91G01Z=-LZ
N22  X=LX
N23  Y=LY
N24  X=-LX
N25  Y=-LY
N26  G00Z=LZ
RTS
```

Function of OSQR



Programmers must keep the following in mind:

① Program name : OSUB

② Number of repetitions : No. of elements in the Y-axis direction

③ Variable setting

LX : Cutting distance of one pattern (X direction)

LY : Cutting distance of one pattern (Y direction)

LZ : Depth of cut

LI : Distance to the next pattern (X direction)

LJ : Distance to the next pattern (Y direction)

LP : No. of elements in the X-axis direction

2. Subprogram Call After Axis Movement (MODIN)

(1) A call after an axis move command establishes the call after an axis move mode. The execution of axis movement is followed by a subprogram run each time an axis move command is issued until it is canceled.

(2) In case the call after the axis move command, the call after the axis mode is established but the move command is neglected.

(3) Programming format:

【 MODIN O__ Q__ Variable-setting 】

O: Program name

Q: Number of repetitions (9999 or Less)

Default is 1

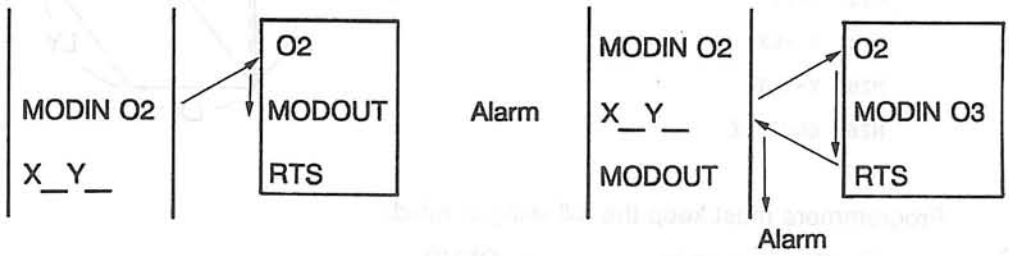
Variable-setting: Variable = expression, variable = expression,

For details, refer to Section 10, 1. "Simple Call". The variable setting is executed each time axis movement occurs and the subprogram is called. However, when the value of a Q command is larger than 2, the second and following subprogram call will not involve variable setting.

(4) This may be canceled by the following command:

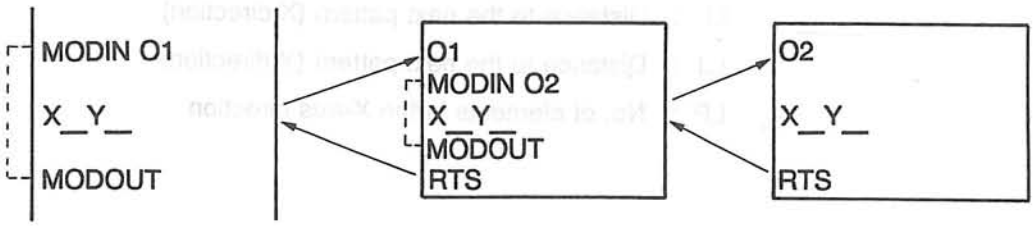
Programming format: MODOUT

This command cancels the last MODIN command that was specified, except for any MODIN commands that have already been canceled. It is important to place a MODOUT command in the same program where the corresponding MODIN command exists.



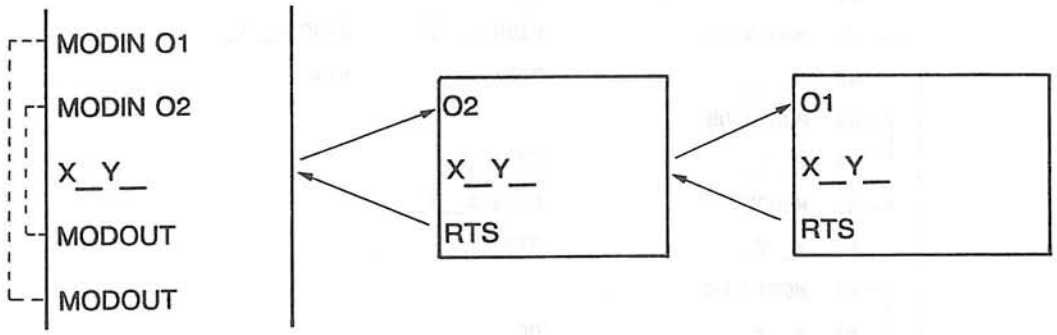
(5) It is possible to specify an 8-subprogram nesting without canceling the call after axis move command. This nesting , also called multiple call after axis move, is available in two manners.

(a) When a call after axis move command is programmed in the subprogram called by a call after axis move command given in a main program:



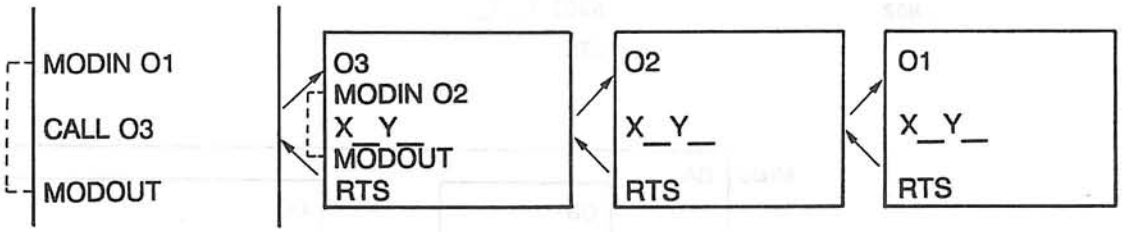
(b) Within a program, when one call after axis move command is followed by another, or when a call after axis move command is programmed in the subprogram called by a call command other than a call after axis move command, then the called subprograms are executed in the following sequence.

1) Within a program, one MODIN command is followed by another:

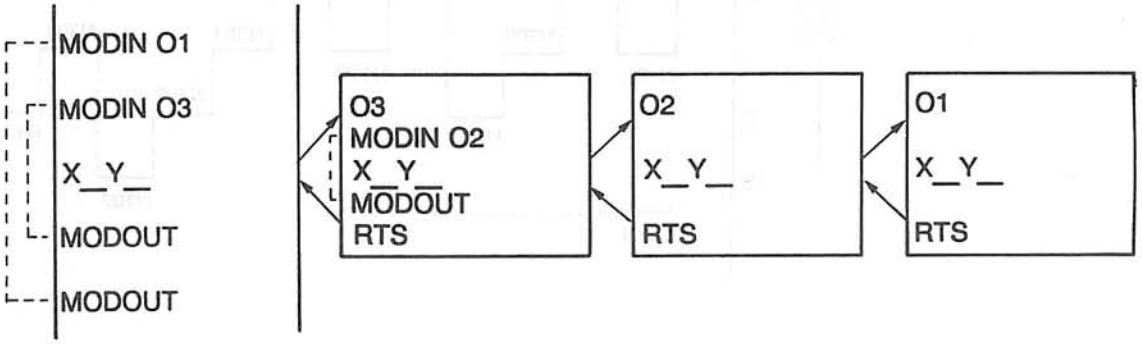


2) After a MODIN command is given in the main program, another MODIN command is given in the subprogram called by a call command other than the MODIN command:

a) For a program called by a CALL command



b) For a program called by a different MODIN command

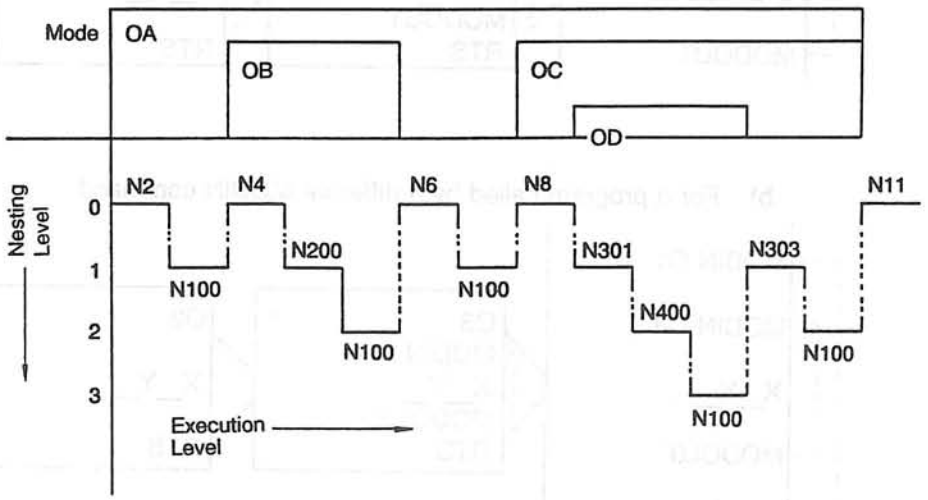


In b), the nesting between O3 and O2 is equivalent to (a); and the nesting between O1 and O2 is equivalent to (b).

Example:

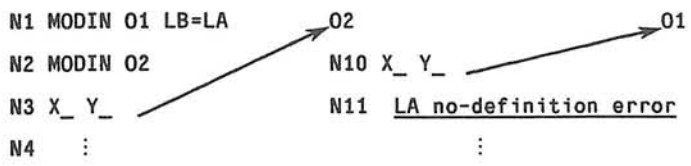
Multiple call after axis movement commands; execution sequence and call nesting level

Main Program	Subprogram	
O1	OA	OD
N1 MODIN OA	N100 X_ Y_	N400 X_ Y_
N2 X_ Y_	RTS	RTS
N3 MODIN OB		
N4 X_ Y_	OB	
N5 MODOUT	N200 X__Y__	
N6 X_ Y_	RTS	
N7 MODIN OC		
N8 X_ Y_	OC	
N9 MODOUT	N300 MODIN OD	
N10 MODOUT	N301 X_ Y_	
N11 X_ Y_	N302 MODOUT	
N02	N303 X_ Y_	
	RTS	



(6) Suppose that in the variable setting of an axis move call command, the right side members use local variables. In any program other than the program involving the call after the axis move command, the following must be noted if a call is made by an axis move command:

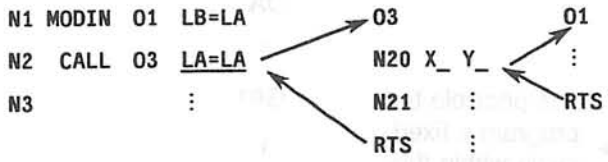
Example:



(7) Upon the axis movement of N3, subprogram O2 is called. When O1 is called with the axis movement command in N10, variables are set: but this results in an alarm because no L;A is defined in the subprogram O2. Therefore, the N2 block must be specified as follows:

N2 MODIN O2 LA=LA

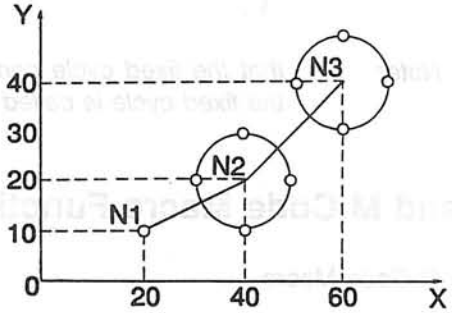
The same applies to the CALL command.



(8) Difference between Subprogram Call after Axis Move Function and Fixed Cycle

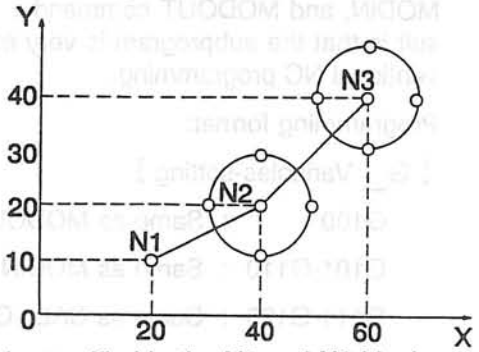
(a) When the drilling cycle is executed by using the subprogram call after and axis move function, drilling is not carried out on points N2 and N3.

```
MODIN OFXC
N1 G90 G00 X20 Y10
MODIN OCYC
N2 G90 G00 X40 Y20
N3 G90 G00 X60 Y40
MODOUT
MODOUT
```



(b) When the drilling cycle is executed by the fixed cycle function, drilling is performed on points N2 and N3.

```
G81 -------
N1 G90 X20 Y10
MODIN OCYC
N2 G90 X40 Y20
N3 G90 X60 Y40
MODOUT
G80
```

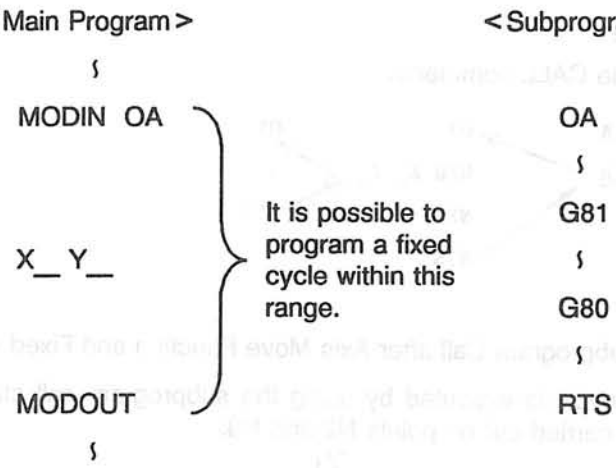


(c) It is possible to make no holes if NCYL is specified in the N2 and N3 blocks. In this case, the ICYC and the IFXC should be a subprogram, as indicated below:

```
OCYC              OFXC
N10 G91 G00 X10   G81  Z_ R_ F_
N11 X-10 Y10      G80
N12 X-10 Y-10     RTS
N13 X10 Y-10
RTS
```

(9) It is possible to program a fixed cycle within a subprogram called after axis movement:

Example:



Note: that the fixed cycle command (G80) must be given in the subprogram from which the fixed cycle is called before the RTS instruction is given.

3. G and M Code Macro Functions

(1) G Code Macro

Commands for calling a subprogram may be issued using ordinary G codes in place of CALL, MODIN, and MODOUT commands. Variable settings are based on address characters. The result is that the subprogram is very easy to use even for operators who are familiar with only conventional NC programming.

Programming format:

【 G__ Variables-setting 】

G100 : Same as MODOUT; variables setting is ineffective.

G101-G110 : Same as MODIN O__.

G111-G120 : Same as CALL O__.

Note: However, that the subprograms that may be called belong to the user subprogram file.

G300 : Same as MODOUT.

G301-G349 : Same as MODIN O__.

G350-G399 : Same as CALL O__.

Note: However, that the subprograms that may be called belong to the maker subprogram file.

Variable-settings:

<address><expression>,<address><expression>...

Addresses may be expressed by address characters and extended address characters, excluding G, M, N and O. Any command value is checked when that variable is actually specified.
Referencing/updating of variables uses local variable names in which addresses are preceded by 'P'. In case the command is not specified, reference does not result in an error, but is handled as EMPTY (undefined).

(2) M Code Macro

(a) A subprogram may be called using an M code in place of a CALL command. Note that variables-setting is not performed. This is also programmed in a block without other commands.

Programming format:

[M__]

M201-M210 : Same in meaning as CALL O__.

Note: however, that the subprograms that may be called belong to the user subprogram file.

(b) The program names corresponding to G101 - G120 and M201 - M210 are set by parameters, where it is not possible to use OO000 to OO999.

(c) G300 to G399 are system-determined and cannot be altered or set by the user.

(d) An alarm will occur if the parameter setting does not include the program name that corresponds to the G or M code macro specified, or if that name is not defined by the system.
An alarm will also occur if that program is not included in the subprogram file. The alarm takes place when the PSELECT command is issued. However, if the program is not selected by the PSELECT command, because, for example, the program name is altered by the program setting after the execution of the PSELECT command, then the alarm takes place at the time of G/M code macro execution.

(e) Variables cannot be used for "G" and "M" used as macro codes.

(f) For G and M, only numerical values are loaded when the PSELECT command is executed and other data, if specified, will result in an alarm.

Example:

G=111
G=VC1 } These are not loaded and an alarm occurs (G code error) when executed.

G112 A program named G112 is loaded.

Example: Line at angle.

Programming format:

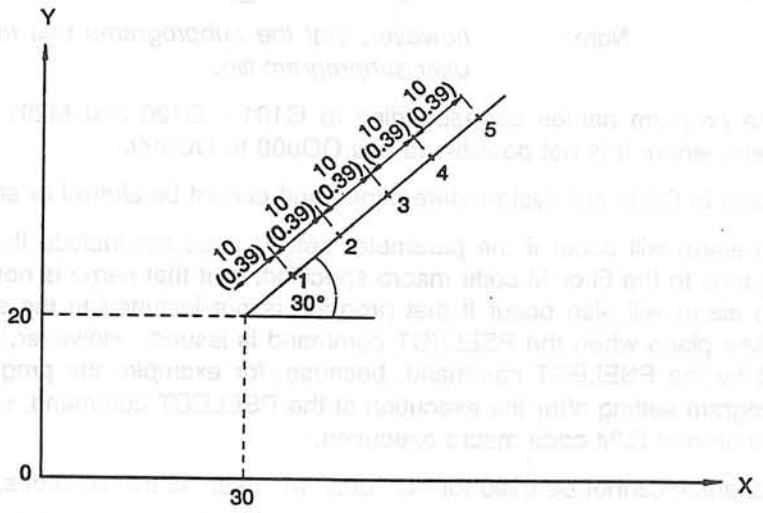
G111 X__ Y__ I__ J__ K__

X,Y : Reference point (Absolute)
I : Interval
J : Angle
K : Number of points

This example may be expressed by the following subprogram, which assumes that OLAA is set in G111 by parameter setting.

```
OLAA
N1  LEN=0
N2  PK=PK-1
N3  IF [PK LT 0] N7
N4  LEN=LEN+PI
N5  G90 G00  X=LEN*COS[PJ]+PX
&            Y=LEN*SIN[PJ]+PY
N6  GOTO N2
N7  RTS
```

A command of "G111X30Y20I10J30K5" gives the result shown below:



SECTION 11 USER TASK

1. User Task 1

The user task 1 allows user access to the high speed processing function, one of the many important features provided by the OSP5020M/OSP500M-G.

The user task 1 consists of the following three functions:

- Branch function
- Variable function
- Math function

1-1. Branch Functions

(1) The branch function controls the execution order of the sequences within programs. It consists of a GOTO statement, which causes an unconditional jump, and an IF statement for a conditional jump.

(2) An alarm will occur if the destination of a jump does not belong to the program, as the source of the branch command. In case there is more than one sequence name as the destination of a jump, the jump take place at the sequence name closer to the beginning of the program.

(3) A block which includes a branch command makes an attempt to find the destination of the jump with the beginning of the program as the starting point, so it takes longer the closer the destination is to the end of the program.

(4) A block which includes a branch command cannot accommodate anything other than the program name, (optional block skip) and sequence name.

N10 X100 Y200 F300 GOTO N20

Alarm

(5) The command programmed after the destination of the jump up to the end of the block code in that block are all ignored.

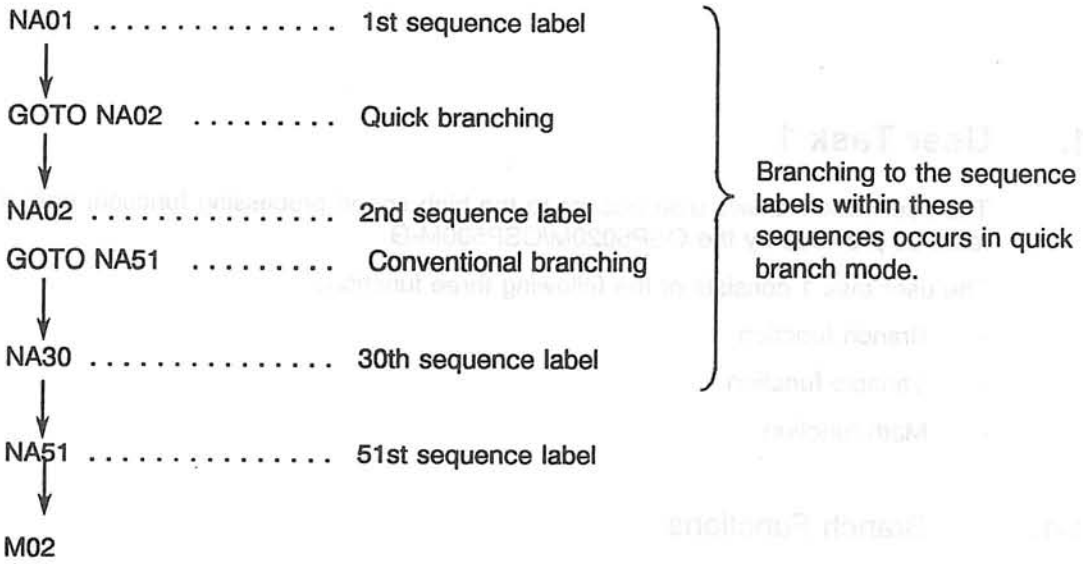
N10 GOTO N20 X100 Y200 F300

Ignored

(6) This function cannot be used in the MDI mode.

(7) If operation method A is selected by the setting of a parameter, branching can be done quickly by designating sequence labels as the destination of a branch command. However, this quick branching is possible only up to 30 sequence labels from the beginning of a program. For sequence labels after these 30 sequence labels, destination of a branch command is searched from the beginning of a program, as done conventionally.

Example: O1000



Sequence label: A sequence name which contains an alphabetic.

(8) For operation method B (program selection operation with large storage capacity memory), which is selected by appropriate parameter setting, the main program involves the following restrictions:

(a) When the S option is specified in the PSELECT command, the execution of a branch command results in an alarm.

(b) The sequence name of the destination of a jump must be a label, which is an alphanumeric string. A number, which is a numeric string, will result in an "undefined" error.

(c) Up to 30 sequence labels may be used, which include ones not specified as the destination of a jump. Failure to follow this limit will result in an error. It is recommended to use sequence labels only for blocks which will be the destination of a jump.

(9) For operation method B, the execution time required for a branch command is not affected by the location of the destination of the jump in the program.

1-1-1. GOTO Statement

This is a branch function which causes a jump unconditionally.

Programming format: [GOTO_N__]

(1) Notes:

(a) Destination sequence for jump should be specified by N__.

(b) A space or spaces must always be provided between the GOTO and the sequence name.

1-1-2. IF Statement

This is a branch function which causes a conditional jump. If the condition is satisfied, the execution sequence jumps to the specified destination. If the condition is not satisfied, the execution sequence moves to the next sequence.

Programming format: [IF [qualification] N__]

[IF [qualification] GOTO_N__]

(1) There are six types of qualifications available.

Operator	Meaning	Example	Contents	Rule
LT	Less Than, <	IF [VC1—LT—5] N100	Jump to N100 when VC1 is less than 5.	Provide a space on either side of the operator.
LE	Less than of Equal to, ≦	IF [VC1—LE—5] N100	Jump to N100 when VC1 is less than or equal to 5.	
EQ	EQual to, =	IF [VC1—EQ—5] N100	Jump to N100 when VC1 is equal to 5.	
NE	Not Equal to, ≠	IF [VC1—NE-5] N100	Jump to N100 when VC1 is not equal to 5.	
GT	Greater Than, >	IF [VC1—GT—5] N100	Jump to N100 when VC1 is greater than 5.	
GE	Greater that or Equal to, ≧	IF [VC1—GE—5] N100	Jump to N100 when VC1 is greater than of equal to 5.	

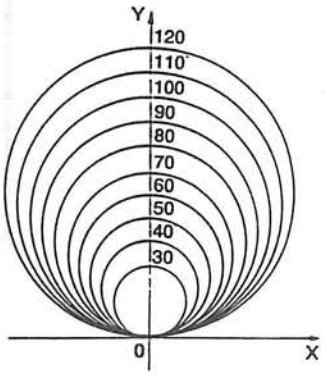
1-2. Variable Function

This function allows the use of variables in the data section of a word such as X=VC1 instead of numerical value such as X100. This gives programs more flexibility and versatility, since assigning numeric values to variables permits the same program to be used for machining similar types of workpieces.

Example:

Using the following program will permit the geometry, shown below, to be easily machined:

```
N1  G90 G00 X0 Y0
N2  G02 J=VC1
N3  VC1=VC1+VC2
N4  VC3=VC3-1
N5  IF [VC3 GT 0] N2
N6  M02
```



Before pressing CYCLE START,
set the variables using parameters as follows:

VC1 = 60 .... Radius of the first circle to be machined

VC2 = −5 .... Pitch of the radius

VC3 = 10 .... Number of circles to be machined

(1) Variables may be assigned to the addresses other than O and N and the extended addresses. '=', blank or HT should be used as a delimiter to be placed between an address and a variable.

(2) If the variable assigned is out of a specified range, determined by individual addresses, an alarm will occur. When only an integer is allowed as a numeric value, decimal fractions are truncated when assigned.

Example: G = VC1 (VC1 = 1 is interpreted as G1)

'=' must be used to assign variables to G and M codes and extended addresses.

(3) EMPTY

(a) The value of an undefined variable is represented by "EMPTY". Particularly, the undefined local variable beginning with "P" is assigned "EMPTY". Other local variables not beginning with "P" are not assigned with initial values unless defined.

(b) The following are interpretations of "EMPTY":

① In Case the Address is Assigned a Variable:

Assigning an undefined variable results in address omission.
The use of an undefined variable in the right member causes an alarm.

VC1 = EMPTY	VC1 = 0
G90X100Y=VC1 ↓ G90X100	G90X100Y=VC1 ↓ G90X100Y0

② In Case an Undefined Variable is Used in the Operational Expression:

"EMPTY" is interpreted as 0, except when no operational codes are used.

VC1 = EMPTY	VC1 = 0
VC2 = VC1 ↓ VC2 = EMPTY	VC2 = VC1 ↓ VC2 = 0
VC2 = +VC1 ↓ VC2 = 0	VC2 = +VC1 ↓ VC2 = 0
VC2 = VC1*VC1 ↓ VC2 = 0	VC2 = VC1*VC1 ↓ VC2 = 0

③ In Case an Undefined Variable is Used in the Qualification.

"EMPTY" is different from 0, only when EQ and NE are used.

VC1 = EMPTY	VC1 = 0
N1 IF [VC1 EQ EMPTY] N10 N2 ↓ Branching to N10	N1 IF [VC1 EQ EMPTY] N10 N2 ↓ To the next N2
N1 IF [VC1 NE 0] N10 N2 ↓ Branching to N10	N1 IF [VC1 NE 0] N10 N2 ↓ To the next N2
N1 IF [VC1 GE EMPTY] N10 N2 ↓ Branching to N10	N1 IF [VC1 GE EMPTY] N2 ↓ To the next N10
N1 IF [VC1 GT 0] N10 N2 ↓ To the next N2	N1 IF [VC1 GT 0] N10 N2 ↓ To the next N2

(4) Array Variables

(a) An array is the set of data having the same elements. The array name should be immediately followed by a subscript enclosed by [ ] to represent the specified elements.

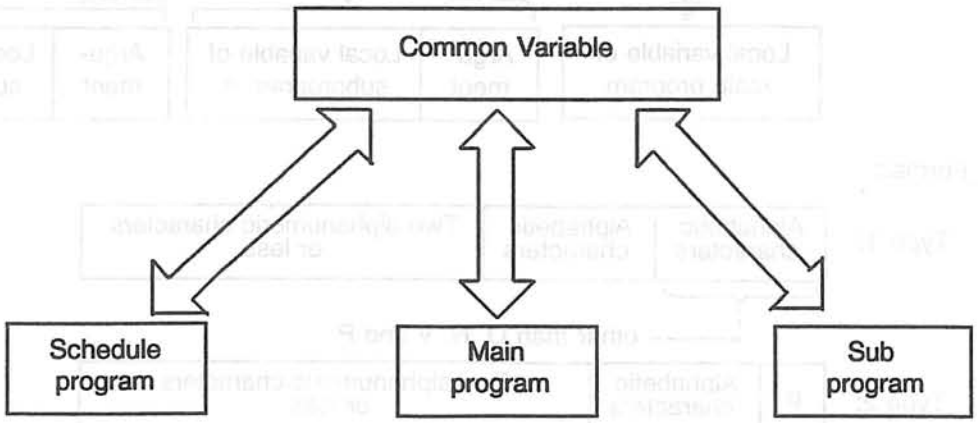
(b) Arrays may be used in the following variables other than local variables:

- Common variables . . . . . VC[n], where n is a subscript
- Some system variables . . . . Example: VZOFX △[n], etc. (user task 2)
- I/O variables . . . . . VDIN[n] or VDOUT [n] (user task 3)

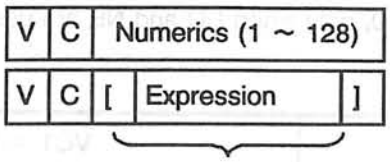
Assume that the arithmetic expression is used. The subscript used in the arithmetic expression is called a subscript expression. If an array variable is used in the subscript expression, an alarm occurs.

1-2-1. Common Variables

This refers to the variables, which are common to schedule programs and subprograms, and they may be referenced or updated in any of these programs.



(1) Format:



(The result of the expression is 1 - 128)

(2) The total number of the common variables is 128 (VC1 ~ VC128). VC001 and VC01 are interpreted as VC1.

(3) The common variable value may be set using parameters. Some variables (VC1 ~ VC32) are not affected by such conditions as power on/off, NC reset, etc. The others (VC33 ~ VC128) are cleared to "EMPTY" when the power is turned on.

(4) For the type 1, specify the common variable number by numerics directly. For the type 2, specify the common variable number by expressions according to the array format.

Example:

VC [1] : Interpreted as VC1

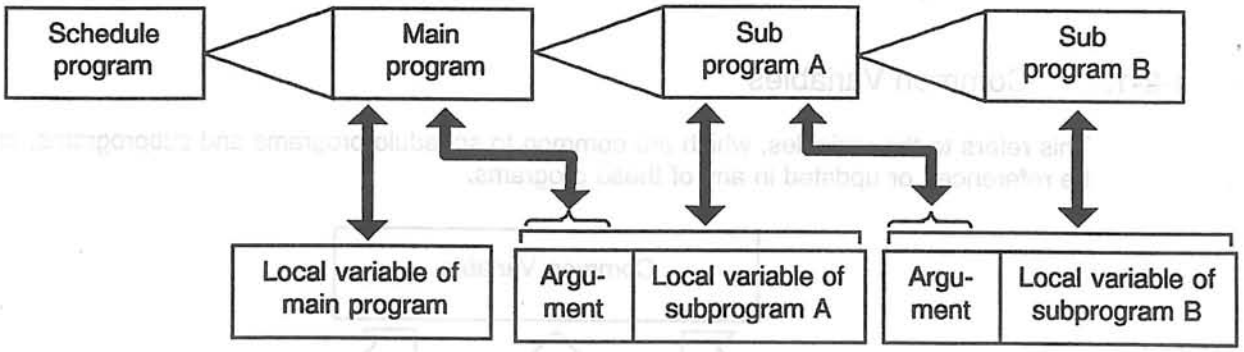
VC [VC1 + 1] : Interpreted as VC11, where VC1 = 10

(5) If the result of the expression falls out of the range (1 - 128) and alarm occurs.

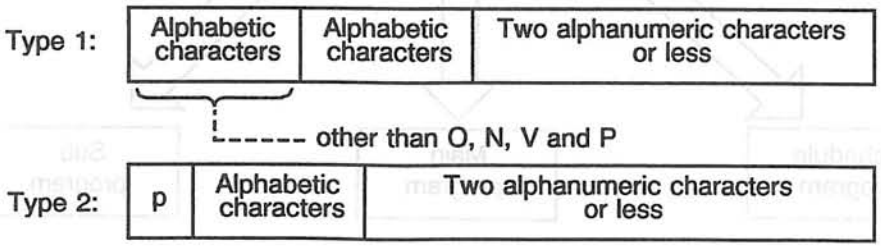
1-2-2. Local Variables

This variable may be set, referenced, or updated by a main program or subprogram. The value is valid only for a particular program. Normally the local variable set by one program cannot be referenced or updated by another program.

When the variable is set by an argument calling a subprogram, the left part local variable, which is a variable of a called subprogram, may be set by the calling subprogram and referenced or updated by the called subprogram. This variable may be used to pass arguments.



(1) Format:



(2) The local variable may be set by defining less than 5 characters except for reserved words to the left part of the '='. The local variables for the subroutine are erased after the subprogram has been executed as many time as commanded. For types 1 and 2, up to 255 variables may be set. All the local variables are erased by power ON/OFF or NC rest, etc.

(3) The type 2 local variable is set or updated in the same manner as the type 1 local variable. When referenced (defined at the right part of '='), the type 2 sets the local variable as "EMPTY", while the type 1 produces an alarm, if that local variable is not set.

(4) The addresses specified to assign arguments of the G code macro instruction is set using a variable name with "P" at its start and is regarded as a local variable for the type 2.

Example:

If G111X110Y200P5 is specified, local variables PX = 100, PY = 200 and PP = 5 are set. If special processing is required for the omitted address, the following programs may be used in macro:

```
N1 IF [PI NE EMPTY] N2
Special processing for omitted address I
N2 IF [PJ NE EMPTY] N3
Special processing for omitted address J
N3 .....
```

Subprogram called by G code macro

(5) List of Local Variable reserved Words

ABS	DGRDX	GRDY	NOT	RTMDI
AND	DGRDY	GRER	OR	RTI
AG	DIN	GROF	PCIR	RTS
ARC	DRAW	GRON	PMIL	SAVE
ATAN	DROUND	GRSK	PMILR	SIN
ATAN2	EIN	GRST	PRINT	SPRINT
BCD	EMPTY	GT	RCIRI	SQRT
BHC	EQ	HA	RCIRO	SQRX
BIN	EOR	HB	RMILI	SQRY
CALL	FA	HC	RMILO	TAN
CLEAR	FB	IF	RH	TLCO
COPY	FC	LAA	ROUND	TLFR
COPYE	FIX	LE	RP	TLFOFF
COS	FMILF	LPRINT	RQNFL	TLFON
DA	FMILR	LT	RS	TLOK
DB	FUP	MITCAN	RSQCO	TN
DC	GE	MOD	RSQRI	
DEF	GOTO	MODIN	RSQRO	
DELETE	GRCI	MODOUT	RSTRT	
DFIX	GRCV	MSG	RT	
DFUP	GRDX	NE	RTMCR	

1-3. Math Function

For specifying numerical values for variables and address characters (X, Y, Z, I, K,...), arithmetic expressions can be directly specified if algebraic calculations are required for obtaining the numerical values to be specified.

Programming Format: Address character or variables = expression

Example: Conventional Programming, ... X 135 X = 100 + XP2

Programming using Math Function, ... XP2 = 35

Expression

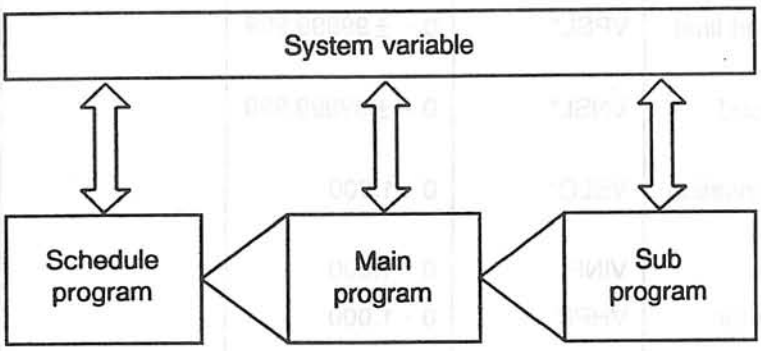
Operator	Meaning	Example
+	Positive sign	+ 1234
–	Negative sign	– 1234
+	Sum (addition)	X = 12.3 + VC1
–	Difference (subtraction)	X = 12.3 – VC1
*	Product (multiplication)	X = VC1 * 10
/	Quotient (division)	X = VC1 / 10

2. User Task 2 (Optional)

The user task 2 allows the use of system variables, logical and function operations, in addition to the functions available with the user task 1. Selection of I/O variable function is also possible.

2-1. System Variables

This variable which is determined by the system, may be referenced and updated by the schedule program, the main program and a subprogram. The system variable is referenced or updated after the previous sequence is executed.



(1) Format:

Type 1:

V	Alphabetical characters	Six alphanumeric characters or less

Type 2:

V	Alphabetical characters	Six alphanumeric characters or less	[	Expression	]

(The second character should not be C.)

(2) The type is determined depending on the system variable. If another type is specified, an alarm occurs.

(3) Some system variables may be set in ZERO SET, TOOL DATA and PARAMETER modes.

(4) The system variables are classified into the following three types:

(a) Read and Write System Variables

(b) Read and Write System Variables Requiring Care in Writing

(c) Read Only System Variables

(5) A List of System Variables

System Variable Name	Format	Range of Set Value	Subscript	Read/Write	Inch/metric Conversion
Zero offset	VZOF* [expression]	0 - ±99999.999	Work coordinate system number	Read/Write	*1
Tool length offset values	VTOFH [expression]	0 - ±999.999	Tool length offset number	Read/Write	*1
Cutter radius compensation values	VTOFD [expression]	0 - ±999.999	Cutter radius compensation number	Read/Write	*1
Positive travel end limit value	VPSL*	0 - ±99999.999		Read/Write	*1
Negative travel end limit value	VNSL*	0 - ±99999.999		Read/Write	*1
Backlash compensation value	VBLC*	0 - 1.000		Read/Write	*1
In-position width	VINP*	0 - 1.000		Read/Write	*1
In-position width for home position	VHPI*	0 - 1.000		Read/Write	*1
Origin of machine coordinate system	VMOF*	0 - +99999.999		Read/Write	*1
Positive programmable travel limit	VPPL*	0 - +99999.999		Read/Write	*1
Negative programmable travel limit	VNPL*	0 - +99999.999		Read/Write	*1
Home position location	VHPP* [expression]	0 - +99999.999	1 - 8 (double column series: 1 - 32)	Read/Write	*1
Sensor contact value	VSAP*	0 - +99999.999		Read/Write	*1
Communication for automation	VNCOM [expression]	Binary, 8 bits (1 byte)	1 - 4	Read/Write	*2
Calculated value	VRCO*			Read only	*1
Actual position	VAPA			Read only	*1
Coordinate system number	VACOD			Read only	*2
Active tool number	VATOL			Read only	*2
Next tool number	VNTOL			Read only	*2
Number of coordinate systems and tool data sets (NC specification code No. 2)	VSPCO			Read only	*2

* indicates an axis address, such as X, Y, and Z.
*1: Converted automatically
*2: No conversion

System Variable Name	Format	Range of Set Value	Subscript	Read/Write	Inch/metric Conversion
Number of tools for ATC	VSPTN			Read only	*2
Spec. code for subprogram (NC specification code No. 24)	VSPSB			Read only	*2
Machine lock	VMLOK			Read only	*2
PPC parameter	VPPCP			Read only	*2
PPC pallet number	VPLNO			Read only	*2
PPC parameter bit data	VPLDT			Read only	*2
3D graphic viewing angle (horizontal)	VGRH			Read only	*2
3D graphic viewing angle (vertical)	VGRV			Read only	*2
Accel/decel coefficient	VDMP*	0 - 65535		Read/Write	*1
DIF-DA coefficient	VDDA*	0 - 65535		Read/Write	*1
Velocity-DA coefficient	VVDA*	0 - 65535		Read/Write	*1
ODIF check value	VDIM*	0.001 - 30.000		Read/Write	*1
DA check value	VDAM*	0 - 16383		Read/Write	*2
APA check value	VALA*	0.001 - 30.000		Read/Write	*1
Unit amount check value	VODM*	0.001 - 30.000		Read/Write	*1
MOP control 1	VMPC1	Binary, 8 bits (1 byte)		Read/Write	*2
MOP control 2	VMPC2	Binary, 8 bits (1 byte)		Read/Write	*2
MOP tool number	VMPT	0 - 9		Read/Write	*2
Printer control	VPCNT	Binary, 8 bits (1 byte)		Read/Write	*2
Automating spec. judgment result 1	VOK1	Binary, 8 bits (1 byte)		Read/Write	*2
Automating spec. judgment result 2	VOK2	Binary, 8 bits (1 byte)		Read/Write	*2
Measuring number	VNUM	0 - 9999		Read/Write	*2
Printer control	VINTG	0 - ±99999.999		Read/Write	*2
Printer control	VPRT	Binary, 8 bits (1 byte)		Read/Write	*2
Active tool number	VTLCN	0 - 65535		Read/Write	*2

* indicates an axis address, such as X, Y, and Z.
*1: Converted automatically
*2: No conversion

System Variable Name	Format	Range of Set Value	Subscript	Read/Write	Inch/metric Conversion
Next tool number	VTLNN	0 - 65535		Read/Write	*2
Setting unit system	VINS	Binary, 8 bits (1 byte)		Read only	*2
Unit system switching flag	VINF	Binary, 8 bits (1 byte)		Read/Write	*2
Tool length/ breakage switching flag	VFST	Binary, 8 bits (1 byte)		Read/Write	*2
Feedrate clamp value	VFDMX	0 - 20000		Read only	*2
Graphic color specification	VGCLR	0 - 9		Read/write	*2
Program unit system	VINCH	Binary, 8 bits (1 byte)		Read/write	*2
Automatic crossrail positioning M code	VECWM	81 - 85, 190 - 199		Read only	*2
Sequence restart flag	VRSTT	Binary, 8 bits (1 byte)		Read only	*2
Machine input data	VMCI [expression]	Binary, 8 bits (1 byte)	Check data No.	Read only	*2
Machine output data	VMCO [expression]	Binary, 8 bits (1 byte)	Check data No.	Read only	*2
Machine extended input data	VMEI [expression]	Binary, 8 bits (1 byte)	Check data No.	Read only	*2
Machine extended output data	VMEO [expression]	Binary, 8 bits (1 byte)	Check data No.	Read only	*2
Panel input data	VPNI [expression]	Binary, 8 bits (1 byte)	Check data No.	Read only	*2
Panel output data	VPNO [expression]	Binary, 8 bits (1 byte)	Check data No.	Read only	*2
Operating time counter	VDTIM[α, β]	0 - 9999.9 (0 - 99999.9)	α: 1 - 5 β: 1 - 2	Read only	*2
Work counter	VWRKC [α, β]	0 - 99999999	α: 1 - 4 β: 1 - 2	Read only	*2

* indicates an axis address, such as X, Y, and Z.
*1: Converted automatically
*2: No conversion

Operating time counter:

α = 1: Power ON time
2: NC operating time
3: Spindle rotating time
4: Cutting time
5: External input time

β = 1: Count value
2: Set value

Work counter:

α = 1: Work counter A
2: Work counter B
3: Work counter C
4: Work counter D

β = 1: Count value
2: Set value

(6) Other system variables are indicated below. These variables are not displayed on the check data screen.

System variable	Format
G code	VGCOD[1] - VGCOD[64]
M code	VMCOD[1] - VMCOD[64]
S code	VSCOD
F code	VFCOD
D code	VDCOD
H code	VHCOD
Mirror image	VMRI
Internal clock	VPMNT 0 - 59 (minute counter; cleared every hour)
	VRMNT (minute counter; not cleared)
	VPHUR 0 - 23 (hour counter; cleared daily)
	VRHUR (hour counter; not cleared)
	VQDAT 0 - number of days
Tool data	VTLD1[m] ⋮ VTLD8[m] — Various tool management data, "m" indicating the maximum value of tool management data
Manual intervention shift amount	VMSF*
	* indicates an axis address (X, Y, Z...)

2-1-1. Read/Write System Variables

(1) Zero Offset

VZOF*[expression]

* ; Axis name ....X - Z, U - W, A - C

Expression: Work coordinate system number

Range ...1 - No. of work coordinate system sets

Read and write operations of the zero offset values for the work coordinate system indicated by the expression are possible. For this setting, unit system is as set by NC optional parameter (bit) No. 3. The axis is determined by the designated axis name.

Example 1: Writing X-axis zero offset No. 10

VZOFX[10] = 20

① μm unit system

Zero offset No. 10 = 20 μm

② mm unit system

Zero offset No. 10 = 20 mm

Example 2: Reading X-axis zero offset No. 10

VC1 = VZOFX[10]

If zero offset No. 10 is 20 mm,

① μm unit system

VC1 = 20000

② mm unit system

VC1 = 20

For details, refer to 2-1-4. "General Rule for Conversion between Inches and Milimeters". (Note: "inch system" refers to the English measurement system.)

(2) Tool Length Offset Value

VTOFH[expression]

Expression: Tool length offset number

Allowable range: 1 - No. of tool data sites

It is possible for the expression to indicate read and write operations of the tool length offset value for the tool length offset number. For this setting, unit system is a set by NC optional parameter (bit) No. 3.

Example 1: Writing tool length offset value at No. 10

VTOFH[10] = 20

① μm unit system

Tool length offset at No. 10 = 20 μm

② mm unit system

Tool length offset at No. 10 = 20 mm

Example 2: Reading tool length offset value at No. 10

VC1 = VTOFH[10]

If tool length offset value at No. 10 is 20 mm,

① μm unit system

VC1 = 20000

② mm unit system

VC1 = 20

For details, refer to 2-1-4. "General Rule for Conversion between Inches and Milimeters". (Note: "inch system" refers to the English measurement system.)

(3) Cutter Radius Compensation Values

VTOFD[expression]

Expression: Cutter radius compensation number

Allowable range: 1 - No. of tool data sets

It is possible for the expression to indicate read and write operations of the cutter radius compensation number. For this setting, unit system is a set by NC optional parameter (bit) No. 3.

Example 1: Writing cutting radius compensation value at No.10

VTOFD[10] = 20

① μm unit system

Cutter radius compensation value at No. 10 = 20 μm

② mm unit system

Cutter radius compensation value at No. 10 = 20 mm

Example 2: Reading cutter radius compensation value at No. 10

VC1 = VTOFD[10]

If cutter radius compensation value at No. 10 is 20 mm,

① μm unit system

VC1 = 20000

② mm unit system

VC1 = 20

For details, refer to 2-1-4. "General Rule for Conversion. between Inches and Milimeters" (Note: "inch system" refers to the English measurement system.)

(4) Positive Programmable Travel Limit

VPPL*

*: Axis name.....X - Z, U - W, A - C

It is possible for the axis name to indicate read and write operations of the programmable travel end limit (+) value. For this setting, unit system is a set by NC optional parameter (bit) No. 3. The value to be set is determined on the work coordinate system currently selected.

Note: *A value exceeding the travel end limit value cannot be set.*

Example 1: Writing programmable travel end limit (+) of the X-axis

VPPLX = 500

① μm unit system

Programmable travel end limit (+) = 500 μm

② mm unit system

Programmable travel end limit (+) = 500 mm

Example 2: Writing programmable travel end limit (+) of X-axis

VC1 = 1VPPLX

If programmable travel end limit (+) is 500 mm,

① μm unit system

VC1 = 500000

② mm unit system

VC1 = 500

For details, refer to 2-1-4. "General Rule for Conversion between Inches and Millimeters". (Note: "inch system" refers to the English measurement system.)

Note: *The travel end limit (+) is set on the machine coordinate system and the programmable travel end limit (+) is set on the work coordinate system.*

(5) Negative Programmable Travel Limit

VNPL*

*: Axis name....X - Z, U - W, A - C

It is possible for the axis name to indicate read and write operations of the programmable travel end limit (-) value. For this setting, unit system is set by NC optional parameter (bit) No. 3. The value to be set is determined on the work coordinate system currently selected.

Note: *A value exceeding the travel end limit value cannot be set.*

Example 1: Writing programmable travel end limit (−) of the X-axis

VNPLX = 500

① μm unit system

Programmable travel end limit (−) = 500 μm

② mm unit system

Programmable travel end limit (−) = 500 mm

Example 2: Writing programmable travel end limit (−) of X-axis

VC1 = VNPLX

If programmable travel end limit (−) is 500 mm,

① μm unit system

VC1 = 500000

② mm unit system

VC1 = 500

For details, refer to 2-1-4. "General Rule for Conversion between Inches and Millimeters.

Note: *The travel end limit (−) is set on the machine coordinate system and the programmable travel end limit (−) is set on the work coordinate system.*

(6) MOP Control 1

(a) VMPC1

System variable VMPC1 designates the MOP control mode.

VMPC1 is one byte data and each bit designates the control mode.

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
Sampling of No-load Spindle Current		Spindle Adaptive Control ON	Spindle Overload Monitoring ON	Sampling of No-load Axis Drive Current	Z-axis Overload Monitoring ON	Y-axis Overload Monitoring ON	X-axis Overload Monitoring ON

Designation of the control mode at VMPC1 is made in a hexadecimal number. The numbers preceded and followed by "#" and "H" respectively are handled as hexadecimal numbers.

(b) VMPC1 combination chart:

① Command code in part program

	Spindle Adaptive Control	Spindle Overload Monitoring	Z-axis Overload Monitoring	Value of VMPC1 (VMPC1 =)
1	×	×	×	0
2	×	×	○	#04H
3	×	○	×	#10H
4	×	○	○	#14H
5	○	×	×	#20H
6	○	×	○	#24H
7	○	○	×	#30H
8	○	○	○	#34H

Note: *Overload monitoring on X-axis and Y-axis is also selectable depending on the machine models.*

② Command code in no-load current sampling

	Spindle No-load Current Sampling	Feed Axis No-load Current Sampling	Value of VMPC1 (VMPC1 =)
1	×	×	0
2	×	○	#08H
3	○	×	#80H
4	○	○	#88H

(7) MOP Control 2

VMPC2 = 0 : MOP OFF
1 : MOP ON

(8) MOP Tool Number

VMPT

(a) The MOP holds the MOP control reference data such as reference value and limit values by the tool number set in the spindle.

(b) In cases where more than one combination of tool data is to be used for a single tool, sub numbers are designated by system parameter VMPT.

(c) This feature is used if cutting is carried out in different spindle ranges although the same tool is used.

(d) When the control is reset, or the control power is turned on, VMPT is set at "0" (VMPT = 0). This means that the tool data number for the MOP function is the tool number of the tool set in the spindle.

(e) Range of VMPT value is from 0 to 9.

Example 1: No. 105 tool set in spindle with VMPT = 7
"TL105-7" is displayed on MOP controller.

Example 2: No. 105 tool set in spindle with VMPT = 3
"TL105-3" is displayed on MOP controller.

Example 3: No. 98 tool set in spindle with VMPT = 0
"TL98" is displayed on MOP controller.

TL105-7, TL105-3 and TL 98 are handled as different numbers in the MOP controller.

(9) Graphic Color Specification

VGCLR

Designation of color for color graphic display is possible, and system parameter VGCLR can be read and written.

Color designation:

VGCLR = n (modal)

n: Integer (0 ~ 9)

This system variable is used to designate the color for tool path display.

n = 0 : Cancel of designated color

n = 1 : Blue

n = 2 : Green

n = 3 : Light blue

n = 4 : Red

n = 5 : Purple

n = 6 : Yellow

n = 7 : White

n = 8 : Black

n = 9 : Not displayed

(10) Tool Management Data

VTLD*[expression]

*: 1 ~ 8

It is possible to indicate read and write of the tool management data with the expression. The objective to be read or written is designated by an asterisk (*).
The following explains how the data is handled:

(a) VTLD1: Tool group number

(b) VTLD2: Tool life management mode

	Criterion for Tool Life Judgment	Replacement to Spare Tool when Tool Life is Expired
1 2 3	Tool life is judged by accumulated cutting time in which cutting tool is fed at the cutting feedrate.	When the T command calling the tool life expired tool is specified Not used Replacement to spare tool does not occur and tool life expired tool is used continuously
4 5 6	Tool life is judged by the count data such as total machined number of holes.	When the T command calling the tool life expired tool is specified next Not used Replacement to spare tool does not occur and tool life expired tool is used continuously.
0	Tool life judgment is not conducted.	

(c) VTLD3: OK/NG flag

Bit 0 - 4 : Used for other functions (Never attempt to change)

5 : NG1

6 : NG2

7 : NG3

(d) VTLD4: Tool life flag

Bit 0 : Tool life

1 : Tool wear

2 : Overload

3 : Torque

4 :

5 : Breakage

6 :

7 : User

(e) VTLD5: Second tool offset number

(f) VTLD6: Third tool offset number

(g) VTLD7: Tool life management setting value

h1 : 2-byte data

① When tool life is judged by accumulated cutting time (tool life mode: 1 - 3):

$0 \leqq h1 \leqq 32767$ (unit: min.)

② When tool life is judged by count data (tool life mode: 4 - 6):

$0 \leqq h1 \leqq 32767$

③ When tool life management is not executed (tool life mode: 0):

No restriction

(h) VTLD8....Remaining tool life value

3-byte data: 2 bytes at the left....h2
1 byte at the right....h3

① When too life is judged by accumulated cutting time (tool life mode: 1 - 3):

$-32768 \leqq h2 \leqq 32767$ (unit: min.)

$0 \leqq h3 \leqq 59$ (unit: sec.)

② When tool life is judged by count data (tool life mode: 4 - 6):

$0 \leqq h2 \leqq 65535$

Always h3 = 0

③ When tool life management is not executed (tool life mode: 0)

No restriction

Example: Reading remaining tool life value of the tool whose tool management number is 10

Reading in units of minutes:

VC1 = VTLD8[10]/256 [Cutting off lower one byte]

VC1 = FIX[VC1]

Reading in units of seconds:

VC1 = VTLD8[10] [Masking of lower one byte]

VC1 = VC1 AND 255

2-1-2. Read/Write System Variables Important in Writing

The system variables explained below have critical influence over machine operations. After they have been written, the machined surface will deteriorate, or unexpected interference will result, if the written values are re-set to the original values. Therefore, do not attempt to write them unless absolutely necessary.

(1) Positive Travel End Limit Value

VPSL*

*: Axis name....X - Z, U - W, A - C

(a) It is possible to designate read and write operations of the travel end limit of the axis indicated by the axis name in the positive (+) direction. For this setting, unit system is as set by NC optional parameter (bit) No. 3. The value to be set is determined on the machine coordinate system currently selected.

(b) If this system variable is re-written, the programmable travel end limit (+) value is automatically re-written to the same value.

Note: The travel end limit (+) is set by the value on the machine coordinate system. At the same time, the value converted into the one on the work coordinate system is set as the programmable travel end limit (+).

Example 1: Reading X-axis travel end limit

Assume that:

- Travel end limit (+) is 500 mm (19.69 in.) on the work coordinate system, and
- Work zero of the currently selected work coordinate system No. 2 is X = 150 mm (7.09 in.)

VC1 = VPSLX ..... Reading travel end limit (+) on machine coordinate system

VC2 = VACOD ..... Reading current coordinate system number

VC3 = VC1 - VZOFX[VC2] ..... Conversion of read travel end limit (+) into the value on the work coordinate system

① μm unit system

VC1 = 650000 VC2 = 2 VC3 = 500000

② mm unit system

VC1 = 650 VC2 = 2 VC3 = 500

(2) Negative Travel End Limit Value

VNSL*

*: Axis name....X - Z, U - W, A - C

(a) It is possible to designate read and write operations of the travel end limit of the axis indicated by the axis name in the negative (−) direction. For this setting, unit system is as set by NC optional parameter (bit) No. 3. The value to be set is determined on the machine coordinate system currently selected.

(b) If this system variable is re-written, the programmable travel end limit (−) value is automatically re-written to the same value.

Note: *The travel end limit (−) is set by the value on the machine coordinate system. At the same time, the value converted into the one on the work coordinate system is set as the programmable travel end limit (−).*

Example 1: Reading X-axis travel end limit

Assume that:

- Travel end limit (−) is -500 mm (19.69 in.) on the work coordinate system, and
- Work zero of the currently selected work coordinate system No. 2 is X = 150 mm (7.09 in.)

VC1 = VNSLX ..... Reading travel end limit (−) on machine coordinate system

VC2 = VACOD ..... Reading current coordinate system number

VC3 = VC1 - VZOFX[VC2] ..... Conversion of read travel end limit (−) into the value on the work coordinate system

① μm unit system

VC1 = −350000 VC2 = 2 VC3 = −500000

② mm unit system

VC1 = −350 VC2 = 2 VC3 = −500

(3) Backlash Compensation Value

VBLC*

*: Axis name....X - Z, U - W, A - C

Read and write operations of the backlash compensation value of the axis indicated by the axis name are possible. For this setting, unit system is a set by NC optional parameter (bit) No. 3.

Example 1: Reading X-axis backlash compensation value

Assume that backlash compensation value of X-axis is 0.05 mm (0.002 in.).

VC1 = VBLCX

① μm unit system

VC1 = 50

② mm unit system

VC1 = 0.05

(4) In-position Width

VINP*

*: Axis name....X - Z, U - W, A - C

It is possible to execute read and write operations of the in-position width of the axis by indicating the axis name. For this setting, unit system is as set by NC optional parameter (bit) No. 3.

Example 1: Reading X-axis in-position width

Assume that the in-position width of the X-axis is 0.003 mm.

VC1 = VINPX

① μm unit system

VC1 = 3

② mm unit system

VC1 = 0.003

(5) Origin of Machine Coordinate System

VMOF*

*: Axis name.....X - Z, U - W, A - C

It is possible to execute read and write operations of the zero point of the machine coordinate system on the axis by indicating the axis name. For this setting, unit system is as set by NC optional parameter (bit) No. 3.

Example1: Reading X-axis origin of machine coordinate system

Assume that the X-axis zero point of the machine coordinate system is 2675.632 mm.

VC1 = VMOFX

① μm unit system

VC1 = 2675632

② mm unit system

VC1 = 2675.632

(6) Sensor Contact Value

VSAP*

*: Axis name....X - Z, U - W, A - C

It is possible to engage read and write operations of the contact positions of the sensor or the touch probe, but only after the execution of the maker subprogram (MSB) for the automatic gaging cycle. The contact position coordinate value is for the axis designated by the axis name and for this setting , unit system is as set by NC optional parameter (bit) No. 3. The value is referenced from the zero position of the position encoder.

Example 1: Reading sensor contact position value on the X-axis.

Assume that:

- Sensor contact point value from the zero position of the position encoder . . . 3500 mm
- Origin of machine coordinate system . . . . . 2000 mm
- Work coordinate system currently selected . . . . . No. 2
- Zero point (X) on the work coordinate system . . . . . 700 mm

VC1 = VSAPX : Reading sensor contact point

VC2 = VC1 - VMOFX : Read value is converted into the value on the machine coordinate system

VC3 = VACOD : Reading the number of the present work coordinate system

VC4 = VC2 - VZOFX[VC3] : Value converted into the value on the work coordinate system

① μm unit system

Example 1: Reading X-axis zero point of the machine coordinate system

VC1 = 3500000 VC2 = 1500000 VC3 = 2 VC4 = 800000

② mm unit system

VC1 = 3500 VC2 = 1500 VC3 = 2 VC4 = 800

(7) Active Tool Number

VTLCN

It is possible to execute read and write operations of the tool number of the tool presently set in the spindle.

Example 1: Reading of the active tool number

VC1 = VTLCN

Note: Read only for the machine with ATC specifications.

(8) VTLNN

It is possible to execute read and write operations of the next tool number.

Example1: Reading of the next tool number

VC1 = VTLNN

Note: Read only for the machine with ATC specifications.

(9) Program Unit System

VINCH

Reading and writing operations of the unit system (NC optional parameter (bit) No. 3) used for the program which is being executed is possible.

Example 1: If the NC optional parameter (bit) No. 3 bits 0 and 1 are ON ("1") with other bits OFF ("0").

VC1 = VINCH

VC1 = 3

Parameter No.	Bit No.	Contents	Setting "1"	Setting "0"
3	0	Unit for distance is either "mm" or "inches".	inch	mm
	1	Units are "1 mm", "1 inch", "1 degree", or "1 sec.".	Yes	Conforms to the settings of bits 2 - 5 and bit 7.
	2	Unit for distance is either "0.01 mm", or "0.001 mm".	0.01 mm	0.001 mm
	3	Unit for feedrate is either "0.1 mm/min, 0.01 in/min", or "1 mm/min, 0.1 in/min".	0.1 mm/mim 0.01 in/mim	1 mm/mim 0.1 in/mim
	4	Unit for feedrate is either "0.001 mm/min, 0.0001 in/min", or "0.01 mm/min, 0.001 in/min".	0.001 mm/rev 0.0001 in/rev	0.01 mm/rev 0.001 in/rev
	5	Unit for time is either 0.01 sec or 0.1 sec.	0.01 sec	0.1 sec
	6	For decimal point data, decimal point position indicates 1mm, 1 inch, 1 deg or 1 sec.	Yes	Conform as to the setting of bits 1 - 5 and bit 7.
	7	Unit for time is either 0.001 sec or 0.1 sec.	0.001 sec	0.1 sec

Note: Read and write operations of the unit system, is enabled by setting the data of the NC optional parameter (bit) No. 34, bit 4.

2-1-3. Read Only System Variables

(1) Calculated Value

VRCO*

*: Axis name....X - Z, U - W, A - C

The calculated value (CON) of the axis designated by the axis name can be read. For this setting, unit system is as set by the NC optional parameter (bit) No. 3. The value is referenced to the zero position of the position encoder.

Example 1: Reading of the calculated X-axis position

Assume that:

- Calculated value ..... 3750 mm
- Zero on the machine coordinate system ..... 2500 mm
- Work coordinate system ..... No. 2
- Work zero (X) ..... 800 mm

VC1 = VRCOX : Calculated value read from position encoder (referenced to zero position of position encoder)

VC2 = VC1 - VMOFX : Read value is converted into the value on the machine coordinate system

VC3 = VACOD : Reading the number of present work coordinate system

VC4 = VC2 - VZOFX[VC3] : Value converted into the value on the work coordinate system

① μm unit system

VC1 = 3750000 VC2 = 1250000 VC3 = 2 VC4 = 450000

② mm unit system

VC1 = 3750 VC2 = 1250 VC3 = 2 VC4 = 450

(2) Actual Position Data

VAPA*

*: Axis name....X - Z, U - W, A - C

The actual value (APA) of the axis designated by the axis name can be read. For this setting, unit system is as set by NC optional parameter (Bit) No. 3. The value is referenced to the zero position of the position encoder.

Example 1: Reading calculated X-axis position

Assume that:

- Actual value ..... 3750 mm
- Zero in the machine coordinate system ..... 2500 mm
- Work coordinate system ..... No. 2
- Work zero (X) ..... 800 mm

VC1 = VAPAX : Actual value read from position encoder (referenced to the zero position of the position encoder)

VC2 = VC2 - VMOFX : Read value is converted into the value on the machine coordinate system

VC3 = VACOD : Reading the number of the present work coordinate system

VC4 = VC2 - VZOFX[VC3] : Value converted into the value on the work coordinate system

① μm unit system

VC1 = 3750000 VC2 = 1250000 VC3 = 2 VC4 = 450000

② mm unit system

VC1 = 3750 VC2 = 1250 VC3 = 2 VC4 = 450

(3) Active Work Coordinate System Number

VACOD

The work coordinate system number of the work coordinate system presently selected can be read.

Example 1: Present work coordinate system number 2

VC1 = VRCOX

VC2 = VC1 - VMOFX

VC3 = VACOD

VC4 = VC2 - VZOFX[VC3]

Thus, VC = 3.

(4) Active Tool Number

VATOL

The tool management number (tool kind + tool number) of the tool presently set in the spindle can be read. The data consists of two bytes; the upper six bits show the tool kind and the lower ten bits represent the tool number.

Tool kind (Some tool kinds cannot be set depending on the machine specifications.)

bit	15	14	13	12	11	10
	*	*	*	*	*	0

*: 0 or 1

bit15

0: Normal tool

1: Large diameter tool (L)

bit 14 to bit 11

0: Normal tool

1: Heavy tool (M)

2: Planer tool (P)

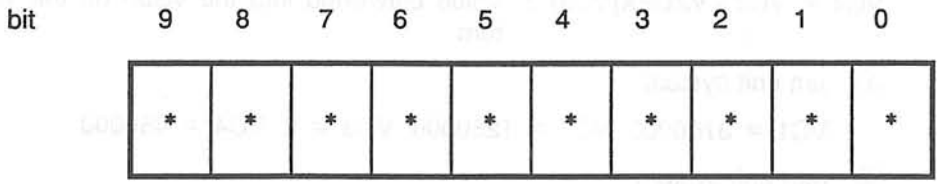
3: Planer heavy tool (PM)

4: Attachment tool (A)

5: Attachment heavy tool (AM)

6: U-axis tool (U)

Tool number



Example 1: Reading the management number of the active tool (Normal tool with the tool number 10)

VC1 = VATOL

VC1 = 10 (#0000000000001010)

Example 2: Reading the management number of the active tool (Normal tool with the tool number 10)

VC1 = VATOL

VC1 = 32778 (#1000000000001010)

Example 3: Reading the active tool number (Heavy tool with the tool number 1)

VC1 = VATOL

VC2 = VATOL AND #03FFH

VC1 = 2049 (#0000100000000001)

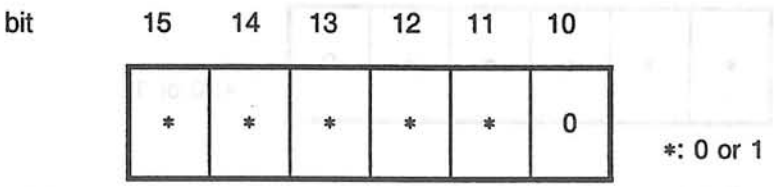
VC2 = 1 (#0000000000000001)

(5) Next Tool Number

VNTOL

The tool management number (tool kind + tool number) of the tool to be used next can be read. The data consists of two bytes; the upper six bits show the tool kind and the lower ten bits represent the tool number.

Tool kind (Some tool kinds cannot be set depending on the machine specifications.)



bit15

0: Normal tool

1: Large diameter tool (L)

bit 14 to bit 11

0: Normal tool

1: Heavy tool (M)

2: Planer tool (P)

3: Planer heavy tool (PM)

4: Attachment tool (A)

5: Attachment heavy tool (AM)

6: U-axis tool (U)

Tool number

bit	9	8	7	6	5	4	3	2	1	0
	*	*	*	*	*	*	*	*	*	*

*: 0 or 1

Example 1: Reading the management number of the next tool (Normal tool with the tool number 10)

VC1 = VNTOL

VC1 = 10

Example 2: Reading the management number of the next tool (Normal tool with the tool number 10)

VC1 = VNTOL

VC1 = 32778

Example 3: Reading only the next tool number (Heavy tool with the tool number 1)

VC1 = VNTOL

VC2 = VNTOL AND #03FFH

VC1 = 2049

VC2 = 1

(6) Number of Tools for ATC

VSPTN

The number of tool pots in the ATC magazine can be read.

Example 1: Reading of tool pot number in the Act magazine

Assume that the number of tool pots in the ATC magazine is "50".

VC1 = VSPTN

Thus, VC1 = 50.

(7) Machine Lock

VMLOK

It is possible to read whether or not the NC is presently in the machine lock status.

Example 1: When the control is in the machine lock status

VC1 = VMLOK

Thus, VC1 = 128.

Example 2: When the control is not in the machine lock status

VC1 = VMLOK

Thus, VC1 = 0.

(8) PPC Pallet Number (for PPC specification)

VPLNO

This is the pallet number of the pallet currently set on the machine table.
In cases when identical workpieces are set on several pallets and if the zero offset values differ slightly on individual pallets, this system variable VPLNO can be used to implement different work coordinate systems for individual pallets.

Example 1:

Assume that #1000 workpieces are set on the pallets No. 1 and No. 2, and that the work coordinate systems used for these pallets are as indicated below:

Work coordinate system No. 11 for pallet No. 1

Work coordinate system No.12 for pallet No. 2

```
O1000
N001  IF[VPLNO EQ 1]N010 ......   To N010 for pallet No. 1
      IF[VPLNO EQ 2]N020 ......   To N020 for pallet No. 2
      MSG(PALLET NO NG) .......   Display of message "PALLET NO NG", cutting cycle not executed
      GOTO NEND                   for this pallet and the cycle proceeds to the next pallet.
N010  G15 H11  ..............     Selection of H11 coordinate system for pallet No. 1
      GOTO N100
N020  G15 H12  ..............     Selection of H12 coordinate system for pallet No. 2
      GOTO N100
N100  T1  ..................      Part program for workpiece #1000
:
NEND  M02
```

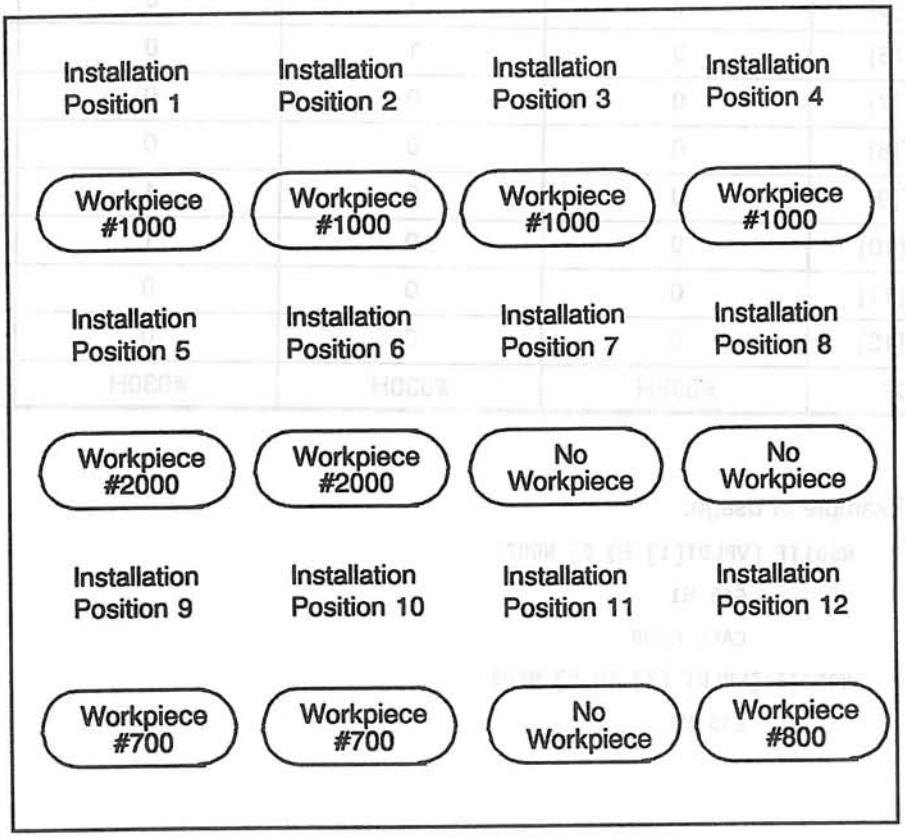
(9) PPC Parameters (for PPC specification)

VPPCP PPC parameter (word) data

VPLDT[expression] PPC parameter (bit) data

When multiple number of workpieces is set on a pallet with PPC set ON, this system is used to indicate the positions where the workpieces are set. The parameter must be set from the PPC panel beforehand.
VPPCP is used to read the parameters comprehensively.
VPLDT is used to read the data of the position indicated by the designated bit.
Up to 12 positions (12 bits) can be used to designate the installation position (PPC parameter) on a single pallet.

Example 1:



Assume that the workpieces are set as indicated above.

System variables VPLDT[1] through VPLDT[12] and VPPCP are set as below where individual programs are executed. These settings can be referenced from part programs.

	O1000 executed	O2000 executed	O0700 executed	O0800 executed
VPLDT[1]	1	0	0	0
VPLDT[2]	1	0	0	0
VPLDT[3]	1	0	0	0
VPLDT[4]	1	0	0	0
VPLDT[5]	0	1	0	0
VPLDT[6]	0	1	0	0
VPLDT[7]	0	0	0	0
VPLDT[8]	0	0	0	0
VPLDT[9]	0	0	1	0
VPLDT[10]	0	0	1	0
VPLDT[11]	0	0	0	0
VPLDT[12]	0	0	0	1
VPPCP	#00FH	#030H	#030H	#800H

Example of usage:

```
N001IF [VPLDT[1] EQ 0] N002
     G15 H1
     CALL OSUB
N002IF [VPLDT [2] EQ 0] N003
     G15 H2
```

(10) G Code

VGCOD[expression]

The mode of the present G code groups can be read. The expression is used to designate the group number of the G code.
The range of selectable G codes is from 1 to 64. The value to be read is the numerical value of a G code. However, "254" is read for G00 mode.

Group	G Code	Group	G Code
1	G0 G1 G2 G3 G60	12	G90 G91
2	G4	13	G94 G95
3	G10 G11	14	G61 G64
4	G15 G16	15	G20 G21
5	G17 G18 G19	16	G30
6	G22 G23	17	G31
7	G40 G41 G42	18	
8	G43 G44	19	G62
9	G50 G51	20	G92
10	G53 G54 G55 G56 G57 G58 G59		
11	G71 G73 G74 G76 G80 G81 G82 G83 G84 G85 G86 G87 G89		

Example 1: In G00 mode

VC1 = VGCOD[1]

Thus, VC1 = 254

Example 2: In the G1 mode in incremental mode

VC1 = VGCOD[1]

VC2 = VGCOD[12]

Thus, VC1 = 1, VC2 = 91

(11) M Code

VMCOD[expression]

The mode of the present M code groups can be read. The expression is used to designate the group number of the M code.
The selectable range of the M code is from 1 to 64. The value to be read is the numerical value of an M code. However, "254" is read for MO

Group	M Code	Group	M Code
1	M0 M1	21	M63 - 65
2	M3 M4 M5 M19	22	M12
3	M6 M70 M77 M170 - 173	23	M50 M51
	M177	24	M9
4	M60 M101 - 112 M178	25	M59
	M179 M160 M161	26	
5	M15 M16	27	M81 - 85
6	M115 M116	28	M150 - M153
7	M17	29	
8	M7	30	M10 M11
9	M130 M131	31	M20 M21
10	M8	32	M22 M23
11	M40 - 43	33	M24 M25
12	M52	34	M57 M58
13	M53 M54	35	M26 M27
14	M66 M67	36	M138 M139
15	M73 - M76	37	M132 M133
16	M134 M135	38	M13 M14 M18 M32 - 37 M48 M49 M181 - 185
17	M136 M137		
18	M2 M30		
19	M144 M145	39	M140 M141
20	M154 M155		

Example 1: Reading rotation direction of the 4th axis rotary table

Assume M15.

VC1 = VMCOD[5]

Thus, VC1 = 15.

(12) S Code

VSCOD

The command value of the present spindle speed command (S) can de read as programmed. Setting of the spindle speed override dial is ignored and the programmed value is read as programmed.

Example : S1200

VC1 = VSCOD

VC1 = 1200

(13) F Code

VFCOD

The command value of the present feedrate (F) can be read.
Setting of the feedrate override dial is ignored and the unit of the data to be read is 0.1 mm/min (for feed per minute mode) or 1 μ /rev (for feed per revolution mode).

Note that conversion to the English system (inches) is not carried out.

Example 1:

Reading the feedrate in units of inch/rev to variable VFCOD/10 when the programming unit system "mm" and G94 mode is active.

VC1 = VFCOD/10

Example 2:

Reading the feedrate in units of inch/rev to variable VC1 when the programming unit system "inch" and G95 mode is active.

VC1 = VFCOD/[1000*25.4]

Note that VFCOD reads only 4-digit F commands and will not be changed by 1-digit F commands.

(14) D Code

VDCODE

The present cutter radius compensation number can be read.

Example :

Assume that the present cutter radius compensation number is "5".

VC1 = VDCOD

Thus, VC1 = 5

(15) H Code

VHCOD

The present tool length offset number can be read.

Example :

Assume that the present tool length offset number is "5".

VC1 = VHCOD

Thus, VC1 = 5

(16) Mirror Image

VMRI

It is possible to read the present status of the mirror image function. The data is of one-byte length and each axis corresponds to each bit.

Mirror image ON : 1

Mirror image OFF : 0

bit	7	6	5	4	3	2	1	0
	0	0	*	*	*	*	*	*
			6th axis	5th axis	4th axis	Z- axis	Y- axis	X- axis

*shows 0 or 1

Example :

Checking procedure for whether or not the mirror image function is active for the X and Z-axes.

VC1 = 1 + 4 X-axis mask (2) + Z-axis mask (2)

VC2 = VMRI AND VC1

IF[VC2 EQ 0] N1

(17) Internal Clock

VPMNT: Minute counter cleared hourly
The range of values read is from 0 to 59

VRMNT: 4-byte free-running minute counter

VPHUR: Hour counter cleared daily
The range of values read is from 0 to 23

VRHUR: 4-byte free-running hour counter

VQDAT: 2-byte free-running date counter
The free-running counter is cleared by turning on/off the power supply.

Example :

To turn on the 60 minute timer

The example below assumes that the duration after turning on the power supply is considerably small compared with two minutes.

```
VC1 = VRMNT
NA1  VC2 = VRMNT - VC1
IF [VC2 GE 60] NA2
GOTO NA1
VC1 = 150000
```

(18) Manual Intervention Shift Amount

VMSF*

*: Axis name....X - Z, U - W, A - C

Manual shift amount of the axis indicated by the axis name can be read. For the setting, the unit system is as set by the NC optional parameter (bit) No. 3.

Example : Reading manual shift amount

Assume that manual shift amount (X) id 150 mm (5.91 in.).

VC1 = VMSFX

① μm unit

VC1 = 150000

② mm unit

VC1 = 150

(19) 3D Graphic Viewing Angle (Horizontal)

VGRH

The angle from the horizontal plane in the 3D graphic display can be read. The unit of the value read is degrees (°).

Example 1:

Reading 3D graphic view angle H

VC1 = VGRH

(20) 3D Graphic Viewing Angle (Vertical)

VGRV

The viewing angle from the vertical plane in the 3D graphic display can be read. The unit of the value read is degrees (°).

Example :

Reading 3D graphic view angle V

VC1 = VGRV

(21) Automatic Crossrail Positioning M Code

VECWM

For the current crossrail position where it has been positioned automatically using an M code, the corresponding M code number can be read.

5 crossrail positioning level specification....81 - 85

10 crossrail positioning level specification....190 - 199

Example : To check the crossrail position using the commands at the head of a program

O100

```
IF [VECWM NE 85] NALM  .............  Checking crossrail position
  :
  :                    }    ......    Machining with the crossrail positioned at M85 level
  :
GOTO NEND
NALM VDOUT[992] = 10  .............  Alarm processing
```

(22) Sequence Restart Flag

VRSTT

The flag is turned on when the restart search command (RS) is executed in the automatic mode; it is turned off after the designated sequence is located. The flag state can now be read.

Setting range: Binary 8 bits (1 byte)

(23) Machine Input Data

VMCI[*]

*: Machine input check data No. (1 to 128)

Data designated on the machine input check data screen is read.

Example:

The sequence jumps to N010 when the machine input check data No. 11 is 64 (bits 2, 5, and 6 are ON).

```
N001 IF[VMCI[11] EQ 64]N010
N002
 :
N010
```

(24) Machine Output Data

VMCO[*]

*: Machine output check data No. (1 to 128)

Data designated on the machine output check data screen is read.

Example: Refer to the example in (23) above.

(25) Machine Extended Input Data

VMEI[*]

*: Machine extended input check data No. (1 to 5)

Data designated on the machine extended input check data screen is read.

Example: Refer to the example in (23) above.

(26) Machine Extended Output Data

VMEO[*]

*: Machine extended output check data No. (1 to 5)

Data designated on the machine extended output check data screen is read.

Example: Refer to the example in (23) above.

(27) Panel Input Data

VPNI[*]

*: Panel input check data No. (1 to 37)

Data designated on the panel input check data screen is read.

Example: Refer to the example in (23) above.

(28) Panel Output Data

VPNO[*]

*: Panel output check data No. (1 to 38)

Data designated on the panel output check data screen is read.

Example: Refer to the example in (23) above.

(29) Operating Time Counter

VDTIM[α, β]

α = 1: Power ON time
= 2: NC operating time
= 3: Spindle rotating time
= 4: Cutting time
= 5: External input time

β = 1: Count value
= 2: Set value

The time counted by counters and their set value are read.

Example:

The sequence jumps to N010 when the cutting time reaches 10 hours.

```
N001IF[VDTIM[4,1] EQ 10]N010
N002
:
N010
```

(30) Work Counter:

VWRKC[α, β]

α = 1: Work counter A
= 2: Work counter B
= 3: Work counter C
= 4: Work counter D

β = 1: Count value
= 2: Set value

The time counted by counters and their set value are read.

Example:

The sequence jumps to N010 when the count value at work counter A reaches 5.

```
N001IF[VWRKC[1,1] EQ 5]N010
N002
:
N010
```

2-1-4. General Rule for Automatic Conversion between Inches and Millimeters

Automatic conversion follows the setting of the NC optional parameter (Bit) No. 3. Bit 0

(1) NC Optional Parameter (Bit) No. 3 Bit 0

Conversion is automatically made into inches when the setting is "1" and into millimeters when the setting is "0", and read/write is accomplished after the conversion.

Example:

VC1 = VTOFH[1]

Tool length offset value set at tool offset number 1 is set at VC1 in inches.

(2) NC Optional Parameter (Bit) No. 3 Bit 1

(a) The unit of length is either millimeters or inches when the setting is "1".

(b) Conversion between millimeters and inches is made according to the setting at bit 0.

(c) If the setting is 0, then the automatic conversion is made in accordance with the setting of other bits.

Example 1:

VC1 = VTOFH[1] (Bit 1 = 1)

VC1 = 2.5 when VTOFH[1] is 2.5 mm (0.098 in.).

Example 2:

VTOFH[1] = VC1 (Bit 1 = 1)

VTOFH[1] = 5.5 mm (0.22 in.) when VC1 is 5.5

(3) NC Optional Parameter (Bit) No. 3 Bit 2

(a) The unit of length is 1/100 mm when the setting is 1 with the millimeter unit (bit 0 = 0) selected.

(b) If the setting is 0, then the automatic conversion is made in accordance with the setting of other bits.

Example 1:

VTOFH[1] = 100 (Only bit 2 = 1)

Thus, VTOFH[1] = 1.0 mm (0.04 in.)

Example 2:

VC1 = VTOFH[1] (Only bit 2 = 1)

VC1 = 520 when VTOFH[1] = 5.2 mm (0.20 in.)

(4) NC Optional Parameter (Bit) No. 3 Bit 6

(a) Millimeters or inches are selected for system variables disregarding the use of a decimal point when the setting is "1".

(b) If the setting is 0, then the automatic conversion is made in accordance with the setting of other bits.

Example 1:

VTOFH[1] = 4 (bit 6 = 1)

Thus, VTOFH[1] = 4 mm (0.16 in.)

Example 2:

VTOFH[1] = 4.0 (bit 6 = 1)

Thus, VTOFH[1] = 4 mm (0.16 in.)

Example 3:

VC1 = VTOFH[1] (bit 6 = 1)

Thus, VC1 = 5.2 when VTOFH[1] = 5.2 mm (0.20 in.)

Priority of bits indicated above is as indicated below:

Priority high	↕	NC Optional parameter (bit) No. 3 bit 0
		NC Optional parameter (bit) No. 3 bit 1
		NC Optional parameter (bit) No. 3 bit 6
Priority low		NC Optional parameter (bit) No. 3 bit 2

(5) Setting of optional parameter (bit) No. 3 is summarized in the chart on the following pages.

Variables (local variables, common variables, system variables) in the right member of the expression are handled in the same manner as the decimal point data.

No.	Bit No.								Example
	7	6	5	4	3	2	1	0	
①	*	0	*	*	*	0	0	0	∘ Assume VTOFH[1] = 4.2 mm VC1 = VTOFH[1] VC1 = 4200 ∘ VTOFH[1] = 5 (5.0) VTOFH[1] = 0.005 mm
②	*	0	*	*	*	0	0	1	∘ Assume VTOFH[1] = 1.5 in. VC1 = VTOFH[1] VC1 = 15000 ∘ VTOFH[1] = 1 (1.0) VTOFH[1] = 0.0001 in.
③	*	0	*	*	*	0	1	0	∘ Assume VTOFH[1] = 4.2 mm VC1 = VTOFH[1] VC1 = 4.2 ∘ VTOFH[1] = 2 (2.0) VTOFH[1] = 2.0 mm
④	*	0	*	*	*	0	1	1	∘ Assume VTOFH[1] = 1.5 in. VC1 = VTOFH[1] VC1 = 420 ∘ VTOFH[1] = 2 (2.0) VC1 = 2 in.
⑤	*	0	*	*	*	1	0	0	∘ Assume VTOFH[1] = 4.2 mm VC1 = VTOFH[1] VC1 = 420 ∘ VTOFH[1] = 5 (5.0) VC1 = 0.05 mm
⑥	*	0	*	*	*	1	1	1	Same as ②
⑦	*	0	*	*	*	1	1	1	Same as ④
⑧	*	1	*	*	*	0	0	0	Same as ③
⑨	*	1	*	*	*	0	0	1	Same as ④
⑩	*	1	*	*	*	0	1	0	Same as ③
⑪	*	1	*	*	*	0	1	1	Same as ④
⑫	*	1	*	*	*	1	0	0	∘ Assume VTOFH[1] = 4.2 mm VC1 = VTOFH[1] VC1 = 4.20 ∘ VTOFH[1] = 5 (5.0) VTOFH[1] = 5 mm
⑬	*	1	*	*	*	1	0	1	Same as ④
⑭	*	1	*	*	*	1	1	0	Same as ③
⑮	*	1	*	*	*	1	1	1	Same as ④

2-1-5. Precautions

(1) Designation of a system variable, which can be read, at the left side will cause an alarm.

(2) Setting of EMPTY for system variables will cause the value 0 to be set.

(3) System variables can be read and written in the machine lock mode.

(4) Do not use system variable while the cutter radius compensation mode is active.

(a) Cutter radius compensation is executed based on the point data of the three points -- actual position, commanded position, and the next commanded position. Therefore, the next block to the one to be executed is read in advance.

(b) The variable function is executed when the block of commands is read and thus, variable function is executed before the execution of positioning at the commanded point. However, this is not always applicable because the buffer reading may be canceled by the parameter setting.

2-2. Math Functions

Various types of operations using variables are possible. Programming for this function can be made in the same manner as with general calculations.

Programming Format: Address character, Variables = Expression

Since the math functions available with the user task 1 are all supported, refer to 1, "User Task 1" in this Section. With the user task 2, logical and functional operations can be used in addition to the available math functions with the user task 1.

2-2-1. Logical Operations

Operation	Symbol	Operation Example	VDOUT[17]
OR	Logical OR	VDOUT[17] = VDIN[17]␣OR␣VDIN[18]	01111110
AND	Logical AND	VDOUT[17] = VDIN[17]␣AND␣VDIN[18]	00001000
EOR	Exclusive OR	VDOUT[17] = VDIN[17]␣EOR␣VDIN[18]	01110110
NOT	Negation	VDOUT[17] = NOT␣VDIN[17]	10010101

Note 1: Examples above are for input and output variables.

Note 2: Place blanks before and after the logical operation symbols (EOR, OR, AND, NOT).

Note 3: The values in the VDOUT column indicate those of VDOUT[17] for VDIN[17] = 01101010 and VDIN[18] = 00011100.

2-2-2. Functions

Function Symbol	Operation	Operation Example	VC1	Remark
SIN	Sine	VC1 = SIN[30]	0.5	
COS	Cosine	VC1 = COS[VC2]	0.5	
TAN	Tangent	VC1 = TAN[45]	1	
ATAN	Arctangent (1)	VC1 = ATAN[1]	45	Value range: −90° - 90° (Note 4)
ATAN2	Arctangent (2)	VC2 = ATAN2[1,[−SQRT[3]]]	150	(Note 5)
SQRT	Square root	VC1 = SQRT[VC2 + 4]	8	
ABS	Absolute value	VC1 = ABS[20−VC2]	40	
BIN	Decimal to binary conversion	VC1 = BIN[VDIN[17]]	(Note 6)	4 bytes
BCD	Binary to decimal conversion	VDOUT[17] = BCD[VC1]	(Note 7)	4 bytes
ROUND	Integer implementation (rounding)	VC1 = ROUND[27.6348]	28	
FIX	Integer implementation (truncation)	VC1 = FIX[27.6348]	27	
FUP	Integer implementation (raising)	VC1 = FUP[27.6348]	28	
DROUND	Unit integer implementation (rounding)	VC1 = RDOUND[13.26462]	13.265	(Note 8)
DFIX	Unit integer implementation (truncation)	VC1 = DFIX[13.26462]	13.264	(Note 8)
DFUP	Unit integer implementation (raising)	VC1 = DFUP[13.26462]	13.265	(Note 8)
MOD	Remainder	VC1 = MOD[VC2,7]	4	60 ÷ 7 = 8...4

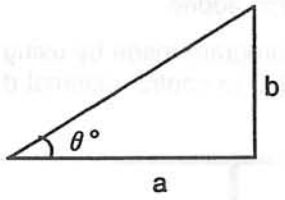
2-2-3. Notes

(1) The variables and numbers after the function operation symbols must be enclosed using '[' and ']'. In addition, they are used for specifying the priority of operation execution order.

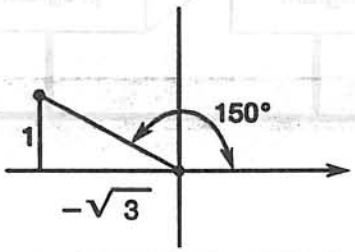
(2) When two elements are specified within the brackets, they must be separated with a comma.

(3) The VC1 column, below, indicates the value obtained from the operation to its left, when VC2 equals 60.

(4) Arc tangent (1) (ATAN)
θ = ATAN [b/a]
Arc tangent (2) (ATAN2)
θ = ATAN2[b,a]



(5) The value of ATAN2[b,a] represents the angle of the point defined by the coordinate values (a,b). Its range is from −180° to 180°.



(6) If the value of VDIN[17] is "01011001" (BCD), the result of operation is VC1 = 59.

(7) If the value of VC1 is 59, the result is VDOUT[17] = 01011001.

(8) In this example the setting unit is mm.

If inches are selected for the setting unit, values will be truncated, rounded up or rounded off to the fourth place to the right of the decimal point.

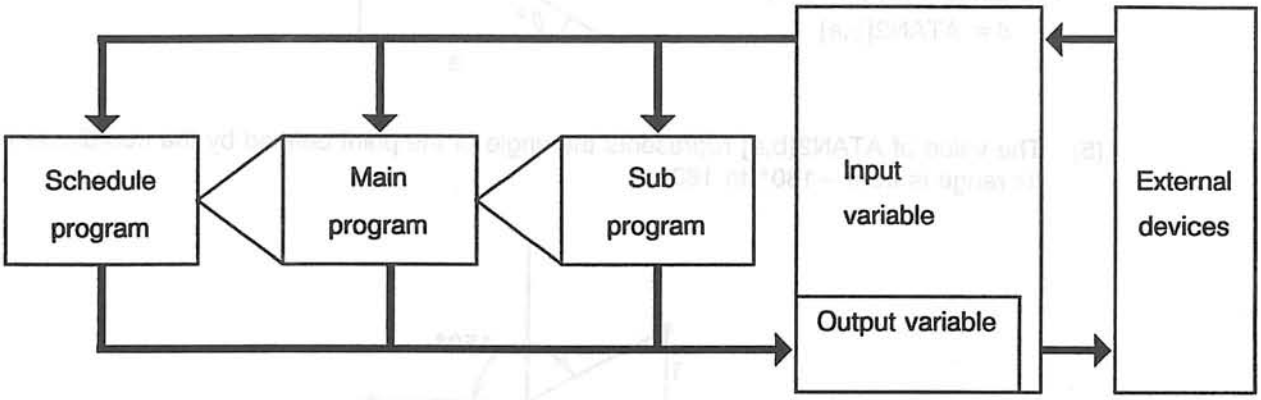
2-3. I/O Variable

The I/O variable can reference or update an I/O signal external to the system, while the system variable is used as internal data.

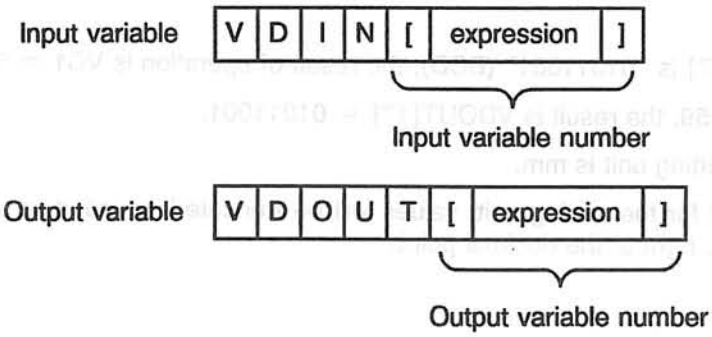
(1) The I/O variable is also be determined by the system and may be referenced (input variable) or updated (output variable) in a schedule program, main program, or subprogram.

(2) The system variables function (timer control, user alarm generation), which is effective for using the I/O variables, is also added.

(3) The use of a control program made by using the I/O variables and the logical and functional operations makes it possible to control external devices.



(4) Programming format:



(5) The I/O variable is referenced or updated after the previous sequence is executed.

(6) Updating (defined at the left of the equal sign "=") the input variable or referencing (defined at the right of the equal sign) the output variable results in an alarm.

2-3-1. Input Variables (VDIN)

Input Variable No. n	Data Contents	Input Device
1 ~ 16	Bit data read at 16 points of input signals 1 (ON), 0 (OFF) bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 n = 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1	EC board
17	1 byte data; n = 9 - 16 corresponds to bits 0 through 7	EC board
18	1 byte data; n = 1 - 8 corresponds to bits 0 through 1	EC board
20	1 word data; n = 1 - 16 crresponds to bits 0 through 15	EC board
1000	Free running time counter; 0.001 sec.	
1001	Free running time counter; 1 sec.	
1002	Free running time counter; 1 min.	
1003	Free running time counter; 1 hour	
1004	Free running time counter; 1 day	

Note: Data at 1000 through 1004 is cleared to zero (0) when the power supply is turned on; it is not cleared by NC reset operation.

2-3-2. Output Variable (VDOUT)

Output Variable No. n	Data Contents	Input Device
1 - 16	Bit data output of 16 output signals 1 (ON), 0 (OFF) bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 n = 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1	EC board
17	1 byte data; n = 9 - 16 corresponds to bits 0 through 7	EC board
18	1 byte data; n = 1 - 8 corresponds to bits 0 through 7	EC board
20	1 word data; n = 1 - 16 corresponds to bits 0 through 15	EC board
990 - 993	Used to generate the following alarm: Alarm No. 990: Alarm D W11 991: Alarm C 929 992: Alarm B 619 993: Alarm A 151 Displays user alarm codes in the following format: VDOUT[n] = XXXX Here, "n" represents an output variable number for alarm level (A - D). "XXXX" represents user reserve code. Here, XXXX is a decimal number of 0 through 9999, with leading zeros suppressed. Example: VDOUT[992] = 2 represents 619 Alarm B. User reserve code 2	

2-3-3. Alarm Message

User designated sub messages for user defined a:arms can be displayed on the screen.

Designation of sub messages can be set at system variable VUACM.

A brief explanation if the system variable VUACM is provided below:

(1) VUACM[n] ..... "n" is a subscript expression.

(2) Format:

VUACM[1] = 'character-string (within 16 characters)'

Describe a character-string between apostrophes.

(3) Once a sub message is set at VUACM, it is displayed at the alarm display line of the screen when a user designated alarm occurs.

(4) VUACM is cleared by NC reset operation. If character data is set at VUACM assigned with a halfway subscript number while VUACM is cleared, corresponding display is not available.

Just after resetting NC,
VUACM[3] = 'ABC' . . . . Subscript number must begin with "1".

(5) Allowable subscript range is from "1" to "16".

(6) Characters exceeding 16 characters are ignored.

(7) Data at the end: The last data set signals the end of all data.

VUACM[1] = 'ABCDEFG'

VUACM[3] = '123'

With the setting as indicated above, the display will be 'AB123'.

(8) VUACM[1] = '^character-string'

The symbol "^" (accidental conflux) at the beginning of the character-string between apostrophes converts upper-case characters into lower-case characters.

VUACM[1] = '^ABCD'
↓
abcd

4th and 5th line characters are converted into characters on the 6th and 7th lines.

(9) To insert an apostrophe within a character-string, place two apostrophes in succession. One of them is discarded and the other is recognized as data.

VUACM[1] = '123''AB'
↓
123'AB

(10) The symbol "^" and one of the two apostrophes placed in succession are not counted as a character.

2-3-4. Precautions

(1) VDIN variables can be designed only at the right part of an operation command. If they are specified at the left part, an alarm occurs.

(2) VDOUT variables can be designated only at the left part of an operation command. If they are specified at the right part, an alarm will occur.
Therefore, if reference to a previous output condition is required, output it via other types of variables such as local variables and common variables, and access that variable.
Numerical values right of the decimal point are rounded.

(3) If a value greater than the allowable size of individual variables is used in an output variable, and alarm will occur.

Bit : 0 or 1

Byte : 0 - 255

Word : 0 - 65535

(4) EMPTY is regarded as zero (0).

(5) Input/output operations are executed with the input/output variables even during machine lock mode.

(6) Input and output status is not influenced by NC resetting operation. Output signals are by turning the power off and then back on again.

(7) Output of output variables is executed with 25.6 msec. to external interface after the command.

This means that two output signals may be output simultaneously if they are designated continuously.

```
VDOUT[1] = 1
VDOUT[2] = 1
```

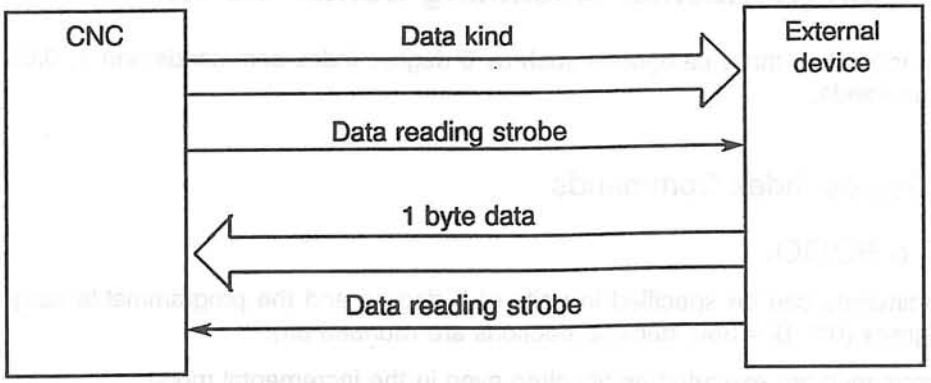
(8) Do not use the system variables, including any input/output variables, during the cutter radius compensation mode. The cutter radius compensation function is executed with the data of three points; active, commanded, and the following point. Therefore, the commands in the next block are read in advance while this function is effective. On the contrary, variables functions are executed as soon as they are read.
Therefore, if system variables are used within the cutter radius compensation function, the following problem occurs:

- The variable function command in the next block is executed before the axis movement command specified in the current block.

The problem stated above is a general problem and other patterns may occur depending on the programmed commands.

2-3-5. Application Example of Input/output Variables

(1) Assume that the information concerning the kind of data is output from the CNC to an external device and the corresponding one byte data is input to the CNC from the external device. This input and output process will require the procedure diagrammed below:



(2) The procedure above can be executed by writing sub programs.

Data kind ..... VDOUT[17]

Data reading strobe ..... VDOUT[1]
(CNC to external device)

One byte data from an external device ..... VDIN [17]

Data reading strobe ..... VDIN [1]
(External device to CNC)

The sub program may be written as below, using the system variables as indicated above.

Main Program

```
VC1=3
CALL  O100
VC3=VC2
```

Sub program

```
O 100
VDOUT[17]=VC1
G4 F1                          Timer: 1 sec.
VDOUT[1]=1
G4 F5                          Timer: 5 sec.
IF [VDIN[1] EQ 0] NALM.......(Note)
VC2=VDIN[17]
VDOUT[1]=0
GOTO NEND
NALM VDOUT[992]=2
NEND RTS
```

Note: *This block of command generates an alarm if the data reading strobe from an external device is not turned on within 5 seconds.*

SECTION 12 OTHER FUNCTIONS

1. B Function (Horizontal Machining Center MC-H)

The B function has three categories such as 5 degree index commands and 1, 0.001 (0.0001) degree index commands.

1-1. 5 Degree Index Commands

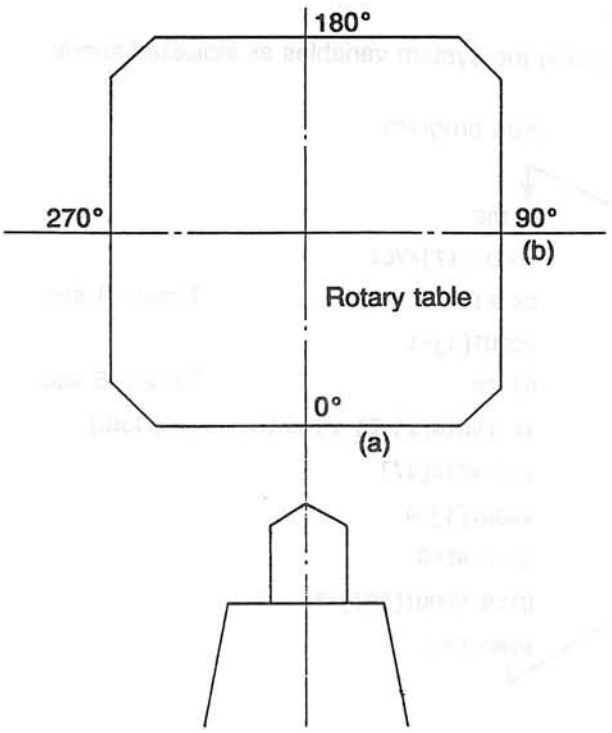
BOOO or BOOO.

(1) Commands can be specified in units of 5 degree and the programmable range is from 0 to 360 degrees (0≦ B ≦360, decimal fractions are rounded off).

(2) Commands are executed as absolute even in the incremental mode.

(3) The direction of rotary table rotation is specified by M15 and M16.

(4) In the G01 mode, a B command should be programmed in blocks not containing axis movements commands. In the G00 mode, it can be programmed with other axis movement commands in the same block.



The rotary index table is provided with angular graduations for 0 to 360 and its angular position is fixed (absolute values).
The rotary index table is provided with angular graduations for 0 to 360 and its angular position is fixed (absolute values).
An example of B-axis rotation within a fixed cycle mode is shown left:

```
N..G81G56 X..Y..R..Z..F..H..M52
N..B60 X50 M15
N..X0 M52
N..B90
N..X0 M52
```

(5) M15 .... Rotary table forward (CW)

M16 .... Rotary table reverse (CCW)

Example:

The commands for indexing position (b) from position (a) and back to position (a) are as follows:

M15

B90

M16

B0

1-2. 1 Degree Index Commands (Optional)

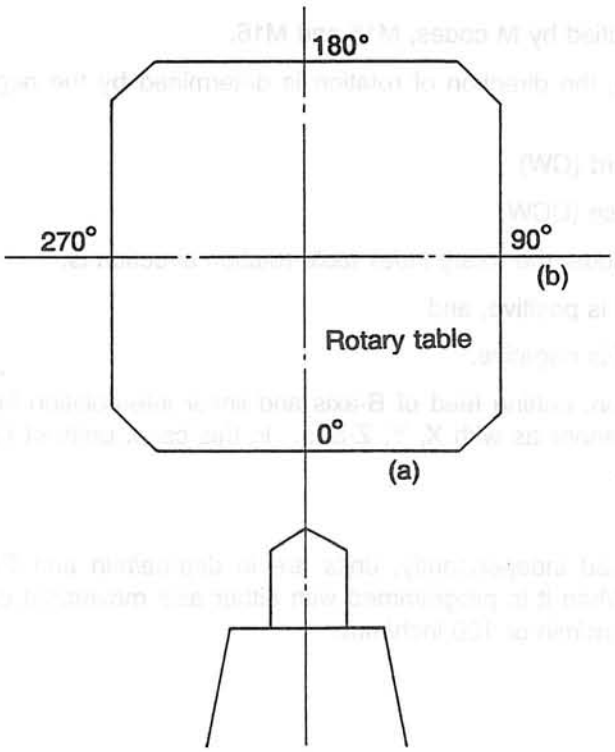
B○○○ or B○○○.

(1) Commands can be specified in units of 1 degree and the programming range is from 0 to 360 degrees (0≦ B ≦360, decimal fractions are rounded off).

(2) Commands are executed as an absolute command even in the incremental mode.

(3) The direction of rotary table rotation is specified by M15 and M16.

(4) In the G01 mode, a B command should be programmed in blocks not containing axis movement commands. In the G00 mode, it can be programmed with other axis movement commands in the same block.



The rotary index table is provided with angular graduations for 0 to 360 and its angular position is fixed (absolute values).
An example of B-axis rotation within a fixed cycle mode is shown left:

N..G81G56 X..Y..R..Z..F..H..M52

N..B60 X50 M15

N..X0 M52

N..B90

N..X0 M52

(5) M15 .... Rotary table forward (CW)

M16 .... Rotary table reverse (CCW)

Example:

The commands for indexing position (b) from position (a) and back to position (a) are as follows:

M15

B90

M16

B0

1-3. 0.001 Degree Commands (Optional)

BOOO. OOOor BOOOOOO

It is possible to select the command type according to the setting for NC optional parameter (bit) No. 39, bit 4. The selectable type is the rotary axis type in which a command should be designated within 360 degrees, and the multi-turn type command which can be designated in the same manner as a linear axis command.

(1) Rotary axis command

(a) Commands should be given in units of 0.001 degree (0.0001 degree) and programmable range is from −360.000 to 360.000 degrees (−360.0000≦ B ≦360.0000).

In units of 0.001, 0.0001 or 1 degree (depending on the setting of the NC optional parameter (bit) No. 3 or bit 0 of No. 4).

(b) The direction of rotation is specified by M codes, M15 and M16.

In incremental command mode, the direction of rotation is determined by the negative sign preceding the angle value.

- M15 ..... Rotary table forward (CW)
- M16 ..... Rotary table reverse (CCW)

In incremental command mode, the rotary index table rotation direction is:

CW when the command is positive, and

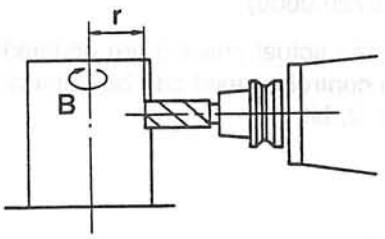
CCW when the command is negative.

(c) With the 0.001 deg. specification, cutting feed of B-axis and linear interpolation including B-axis is possible in the same manner as with X, Y, Z-axis. In this case, units of feedrate for the B-axis must be attended to.

B-axis feedrate:

When B-axis is programmed independently, units are in degree/min and F100 corresponds to 100 deg/min. When it is programmed with either axis movement commands, F100 is executed as 100 mm/min or 100 inch/mm.

Example 1:



Assume "r" in the figure at the left to be 100 mm, B-axis rotates at a rate of 100 deg/min when the commands G91 G01 B100 F100 are programmed.

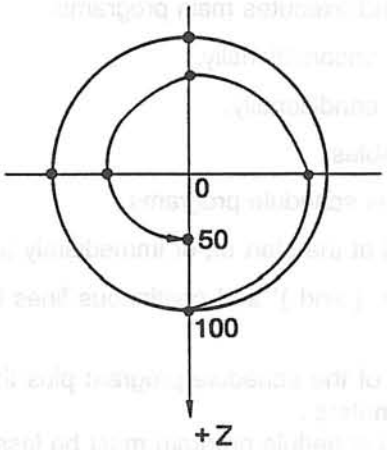
The actual cutting speed (surface speed on a workpiece at a cutting point) is:

$100 \times 2 \times 3.14 \times (100/360) = 174$ mm/min

Example 2:

The axis movement distance is calculated as:

G91 G01 B360 Z-50 F100



$\sqrt{(360)^2 + (-50)^2} = 363.456$

The feedrate is 100 mm/min.

The time required for the axes to move the programmed distance is:

363.456/100 = 3.6

(2) Multi-turn command

(a) Commands are designated in units of 0.001 deg. (0.0001 deg.) in the range of $-9999.999 \leqq B \leqq +9999.999$ ($-9999.9999 \leqq B \leqq +9999.9999$).

The units of command conform to the setting for bit 0 of NC optional parameter (bit) No. 3 or No. 4.
Designation is possible in units of 0.001 degrees, 0.0001 degrees or 1 degree.

(b) Rotating direction is determined according to the relative position of the target position from the actual position regardless of the dimensioning mode (absolute or incremental).
When the target position is located in the increasing coordinate value direction in respect to the actual position, the axis rotates in the forward direction. If it is in the decreasing coordinate value direction, the axis rotates in the reverse direction.

Designation of M15/M16 is ignored.

(c) Travel limits (P/N) are fixed.

Travel limit (P) 9720.000 (9720.0000)

Travel limit (N) .... −9720.000 (−9720.0000)

(d) Whether the coordinate values of the actual position are updated to a value within the range of 0 to 359.999 degrees when the control is reset can be determined according to the setting for NC optional parameter (bit) No. 2, bit 1.

2. Schedule Programs (Optional)

The schedule program permits different types of workpieces to be machined continuously without any operator intervention by using a pallet changer, or other automated loading and unloading units.

(1) Several main programs can be selected and executed in the specified order by the schedule program.

(2) A schedule program is a set of the following five blocks:
If other blocks are specified, and alarm will occur.
The program must be terminated with the END block.

(a) PSELECT block . . Selects and executes main programs.

(b) GOTO block . . . . Branches unconditionally.

(c) IF block . . . . . Branches conditionally.

(d) VSET block . . . . . Sets variables.

(e) END block . . . . . Terminates schedule programs.

(3) These commands must be specified at the start of, or immediately after, the sequence name.

(4) Although comments given between '(and)' and continuous lines identified by '$' are valid, the optional block skip (/) is invalid.

(5) For method A, the total tape length of the schedule program plus the main program and subprogram must be less than about 160 meters .
For method B, the tape length of the schedule program must be less than about 5 meters .

2-1. PSELECT Block

This block selects and executes main programs for a workpiece to be machined.

(1) This function searches a specified main program file for a specified main program to be selected as a machining program. This function also searches a specified subprogram file, or system subprogram file, and manufacturer subprogram file for the required subprograms and selects them automatically.

(2) After the completion of selection, it starts execution if the single block function is "off", and waits execution until the cycle start signal is given if the single block function is "on". It repeats selected programs as specified.

(3) If the system is not in the AUTO mode, the selection and execution of a main program is put off until the system is placed in the AUTO mode.

(4) Programming format:

The command must be specified in the following order:

[ ] may be omitted.

{PSELECT}␣[fm],[pm],[fs][;OP␣],[n] (CR) /(LF)

(a) fm: Main program file name

[ ] may be omitted.

[3 characters :][16 characters or less][. 3 characters]

device name — file name — extension

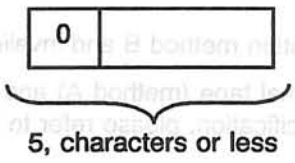
① The defaults of the device, file, and extension (A comma (,) can be omitted) are 'BB1', 'A' and 'MIN', respectively.

② The default of fm is BB1:A.MIN.

③ If '*' or '?' is used in a main program file name, an alarm will occur.

④ If the specified file does not exist, an alarm will occur.

(b) Pm: Main program name



① The default of Pm is the name of the top program of the file specified by fm.

② If the specified file odes not exist in fm, an alarm occurs.

③ The absence of M02 or M30 indicating the end of a program cause an alarm.

(c) fs: Subprogram file name

[ ] may be omitted.

[3 characters :][16 characters or less][. 3 characters]

device name — file name — extension

① fs may be omitted when:

- no subprogram is called in the main program,
- the subprogram called from a main program or subprogram exists in BB1:*.SSB (system subprogram) or in BB1:*.MSB (manufacturer subprogram), and
- required subprograms other that SSB and MSB are contained in the main program file.

If fs is specified, the device name and extended name may be omitted. Their defaults are 'BB1' and 'SUB' respectively. If all of them are omitted, it is assumed that no file has been specified.

② If the total number of used subprograms exceeds 62, an alarm will occur.

③ If 'RTS', which means the end of the subprogram, does not exist, an alarm will also occur.

④ Also, an alarm will be triggered if the required subprogram name that is OO000 to OO999 does not exist in a manufacturer subprogram file.

⑤ If the required subprogram name other than OO000 to OO999 does not exist in the fs specified file or the system subprogram file, and alarm occurs.

Note: If the file specified by fs does not exist, an alarm occurs.

(d) n: Repetition count

Q	=	Expression

① The repetition count (the number of the called program to be repeated), which must range from 1 through 9999, may be specified using the address 'Q'. Its default is 1.

② If other that 1 - 9999 is specified, and alarm will occur. '␣' substitutes for '='. If '=' is followed by a numeric value, it may be omitted.

(e) OP: pecification of options

① Specification of S option

;S

This is the command not to search for subprograms.

An S option significantly reduces the run time of the PSELECT command. This option is effective only for main programs and if a subprograms function or a branch function is used, and alarm occurs.

The S option is valid for the operation method B and invalid for method A.

For the difference between a normal tape (method A) and a large capacity tape (method B), with and without S option specification, please refer to table 12-1.

② Specification of A option

;A

The specification of an A option always selects operation method A independent of the setting of optional parameter (word) No. 11.

③ Specification of B option

;B

The specification of a B option always selects operation method B independent of the setting of optional parameter (word) No. 11.

B option is effective only in operation method B. If this option is selected in operation method A, it is ignored.

Table 12-1. Normal/ Large Storoge Capacity Tapes

Item		Selection and Operation for Normal Storage Capacity	Selection and Operation of Large Storage Capacity		
Parameter setting		Method A	Method B		Method S
Specification of S option in PSELECT command mode		Invalid	Valid		—
			S option not specified	S option specified	
Program size limitation	Main program	Total tape length is less than 80 meters (optionally 160 meters *1)	Up to total length of the stored main program		Same as Method B
	Subprogram		Total tape length is less than 100 meters *1	—	—
	Library program			Total tape length is less than 100 meters. *1	Same as Method B
	Schedule program				
Subprogram function		Available	Available	Unavailable (alarm)	Same as Method B
Branch function		Available	Available	Unavailable (alarm)	Same as Method B
Instruction for jump destination of branch instruction	Main program	Sequence number, sequence label	Sequence label only	—	—
	Subprogram		Sequence label or sequence number		
	Library program				
	Schedule program				
Main program label limit		No limit	Fewer that 30 pcs.	No limit	Same as Method B
Execution time for PSELECT command		Several minutes	Several minutes	Ends at once	Same as Method B

* This capacity can be extended by selecting the operation buffer expansion specification.

2-2. Branch Block

The branching functrion of the schedule program, which is identical to Section 11, 11-1. "Branch Function", falls into GOTO and IF blocks, which provide unconditional branching and conditional branching, respectively.

(1) GOTO Block

The GOTO block unconditionally changes program sequences. The destination of a jump is to be specified using a sequence name immediately after the GOTO command.

Programming format:

Commands must be specified in the following order:

GOTO␣N _____

Specifies the destination of a jump

(2) IF Block

The IF block conditionally changes program sequences. If the condition is 'true', the sequence branches to the destination of a jump. If the condition is 'false', it proceeds to the next stop.

Programming format:

Commands must be specified in the following order:

IF [Expression ␣ Comparison operator ␣ Expression] N_____

Specifies the destination of a jump

The copmparison operators include LT(<), LE(≦), EQ(=), NE(≠), GE(≧), GT(>).

2-3. Variables Setting Block

'VSET' must be specified to set variables using the schedule program.

Programming format:

Commands must be specified in the following order:

VSET ␣ [Variable] = [Expression] ␣ [Variable] = [Expression]

If any variable other than a common variable, system variable, or I/O variable, is specified at the left part, an alarm will occur.

Addtionally, if an I/O variable is specified at the right part, an alrm will also occur.

2-4. Schedule Program Termination Block

The schedule program must be terminated with an 'END' block. All the blocks form the 'END' block on are invalid.

Programming format:

END

3. Library Program

This is the function necessary for executing subprograms and G code macros with MDI, executing a program containing a subprogram call with an S option specified in the large capacity operation mode, etc.
In other words, although the subprograms that are necessary for operating a main program are stored in the operation area by selecting the PSELECT mode, subprogram registration is also possible with this library program registration function.

For details, please refer to Section 9-23, "Library Program Registration" in Section 2 of the Operational Manual for OSP5020M/OSP500M-G.

4. File Management

4-1. Files

(1) Programs are executed after they have been stored in the OSP5020M/OSP500M-G memory.

(2) As a standard feature, the memory in the control has a capability to store up to 40 meters (131.2 ft.) of tape, and a maximum of 5,120 meters (16,797.9 ft.) can also be stored optionally. Therefore, several programs are stored in the memory at the same time.

(3) To facilitate the handling of stored programs, each is stored in respective files with different names. This is similar to keeping documents in lockers in files.
See Fig. 12-1 on the following page.

(4) The file management system is composed of the following three kinds:

(a) Bubble Memory (data bank)

This functions as a locker where data is stored in units of files.

(b) Various Files

A file functions as a document or an account book.

(c) Program Name or Number

Program numbers or names will be the same as individual documents. (process sheet)

4-2. Various Files

(1) Files may be equivalent to document files or account books, and each file for the same workpiece type is assigned a name (file name), which consists of the main file name and an extension.

(2) A file name should consist of up to 16 alphanumeric characters including a minus code, beginning with an alphabetical character. An extension should consist of 3 characters beginning with an alphabetical character, separated from a main file name by a period (.) between them.

Main file name	.	Extension

(Up to 16 characters beginning with an alphabetical character)

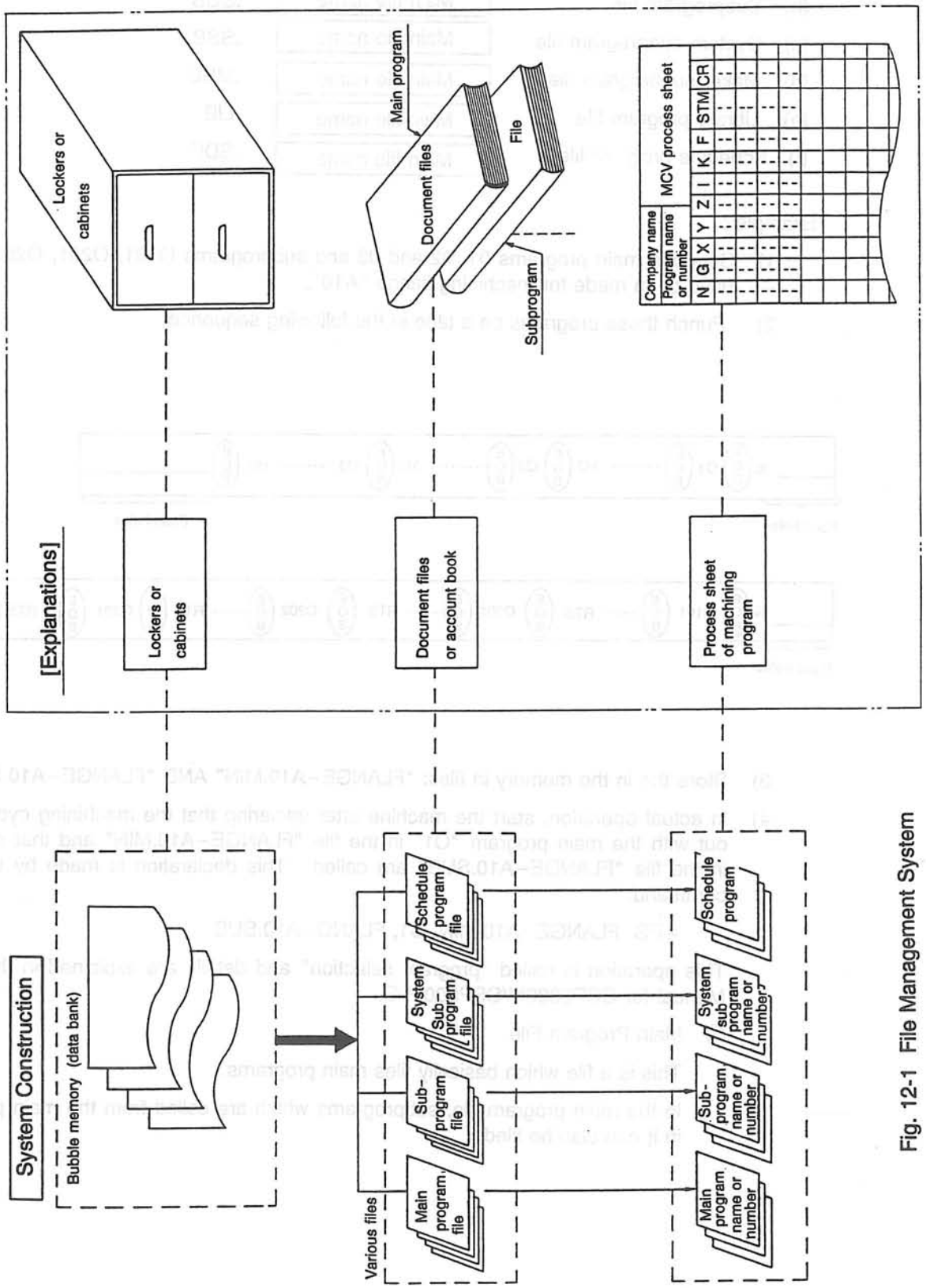


Fig. 12-1 File Management System

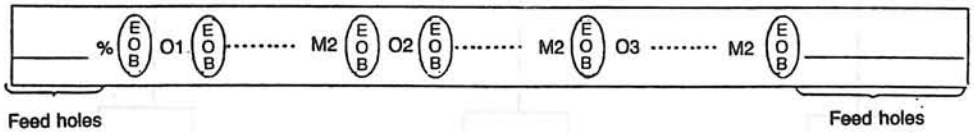
(3) Types of files:

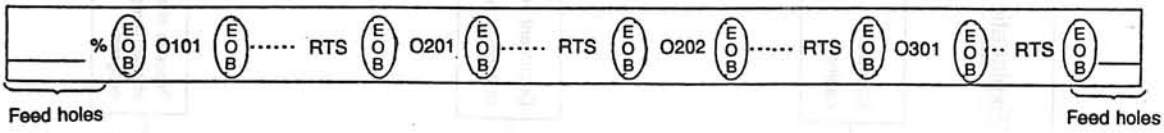
(a)	Main program file	Main file name	.MIN
(b)	Subprogram file	Main file name	.SUB
(c)	System subprogram file	Main file name	.SSB
(d)	Maker subprogram file	Main file name	.MSB
(e)	Library program file	Main file name	.LIB
(f)	Schedule program file	Main file name	.SDF

Example :

1) Suppose main programs 01, 02 and 03 and subprograms O101, O201, O202 and O301 have been made for machining flange "A10".

2) Punch these programs on a tape in the following sequence:





3) Store the in the memory in files: "FLANGE−A10.MIN" AND "FLANGE−A10.SUB".

4) In actual operation, start the machine after declaring that the machining cycle is carried out with the main program "O1" in the file "FLANGE−A10.MIN" and that subprograms in the file "FLANGE−A10.SUB" are called. This declaration is made by the following command:

=PS FLANGE−A10.MIN, O1, FLANG−A10.SUB

This operation is called "program selection" and details are explained in the Operation Manual for OSP5020M/OSP500M-G.

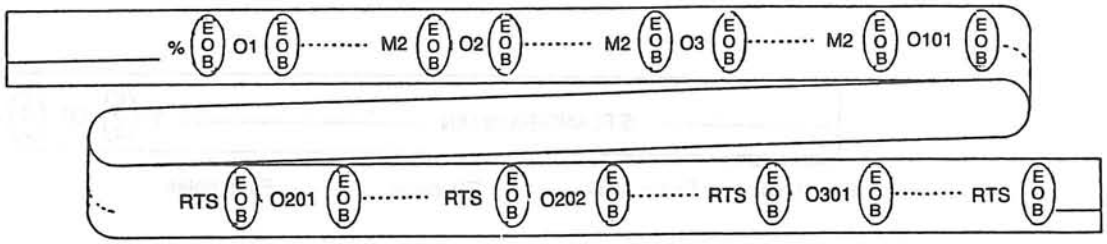
① Main Program File

This is a file which basically files main programs.

In the main program file, subprograms which are called from the main program filed in it can also be filed.

Example :

When filing a subprogram named FLANGE−A10.SUB in Example 1 with the main program FLANGE−A10.MIN, make an NC program as shown below:



② Subprogram File

This is a file of subprograms.

When a machining cycle is to be carried out with the programs in this file, it is necessary to specify the subprogram name in selecting programs.

③ System Subprogram File

a) Subprograms, when they are called and used, must always be specified in the program selection.

b) Assume that a subprogram OPAN which can be used in common to several main programs has been made and that the main programs which call this subprogram be O1, O2, and O3, which call subprograms O101, O201, and O301, respectively. Then the subprogram OPAN must be filed in each of the subprogram files which files the subprograms O101, O201, O301.

c) The system subprogram file is used to file such general-purpose subprograms and when a subprogram to be called from the main program is not in the main program file or the subprogram file specified at the same time, the subprogram in this system subprogram file is called and used.

④ Maker Subprogram File

This file is handled in the same manner as a system subprogram file. In this file, subprograms made by us at Okuma for such automating are filed.

⑤ Library Program File

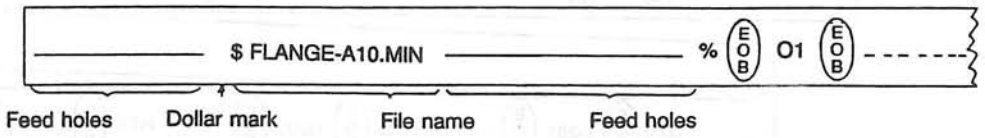
Generally, for the use of a subprogram in the MDI mode operation entering the desired subprogram number such as CALO100, the main program which called this subprogram must be selected in advance.
By filing the subprograms which are often called in the MDI mode operation in the library program file, they can be easily called.

⑥ Schedule Program File

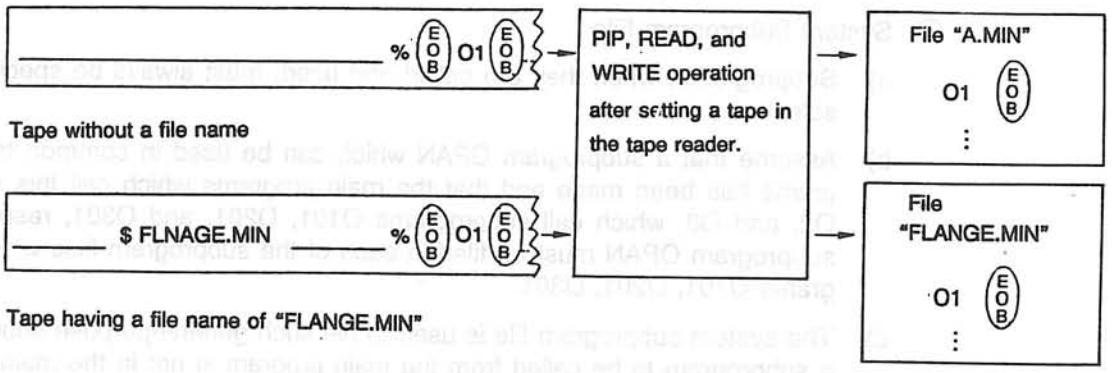
Refer to 2. "Schedule Programs" in this Section.

4-3. File Name of Paper Tape

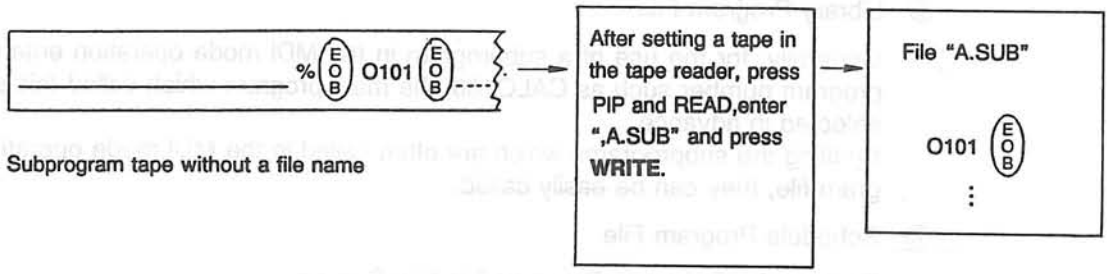
On the OSP5020M/OSP500M-G, a paper tape containing programs is regarded as a file. This means that a file name can be given to paper tapes. The manner for giving a file name to a paper tape is shown below:



The omission of a file name indicates the file name of "A.MIN", Therefore, when a tape without a file name , as the one shown below, is read, the program is filed in the file, "A.MIN".



A subprogram tape without an assigned file name is also filed in the main program file, "A.MIN", if directly read, making it impossible to call it later. Therefore, to store such a subprogram, it is necessary to follow the operations below:



4-4. Operating Files

(1) The read and program selection are examples of file operations. In addition, editing, punching, copying (duplication), etc. are included in the file operation.

(a) For instance, the read operation is carried out by the command shown below:

>R <input file name>, <output file name>

① Here, the input file name and output file name refer to the names Flange-A10.MIN and A.MIN. In addition to the file names, the device name from which the file is to be read and the one where the file is to be stored must be given. To read the file, FLANGE-A10.MIN, from the tape reader and store it in the control memory with the same file name, the command should be as follows:

>R TR:FLANGE−A10.MIN, BB1:FLANGE−A10.MIN

The abbreviations TR and BB1 refer to the following devices:

TR: Tape reader
BB1: CNC memory (bubble cassette memory)

② If a device name is not given, a specific device name is assumed depending on the type of operation intended. For example, in the read operation, the device name of the input file is "TR:" and the output file name is "BB1".

③ The omission of the output device name indicates the same device name as the input device.

That is, the command ">R TR:FLANGE-A10.MIN, BB1:FLANGE−A10.MIN" is simply written as ">R FLANGE−A10.MIN".

④ From the explanation above, the popular operation like reading from the tape reader and storing in the bubble memory is instructed with the command not containing full information.

⑤ For details of the operation, refer to Section 2, 7-4. "Transfer" on the Operation Manual for OSP5020M/OSP500M-G.

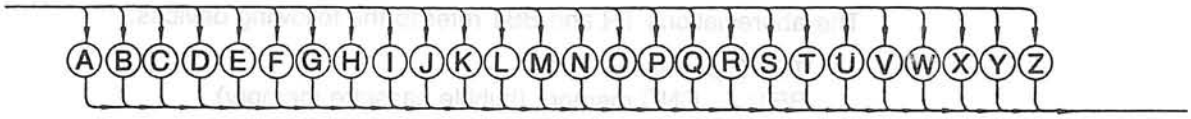
SECTION 13 NC PROGRAM GRAMMAR

General Rule of Legend:

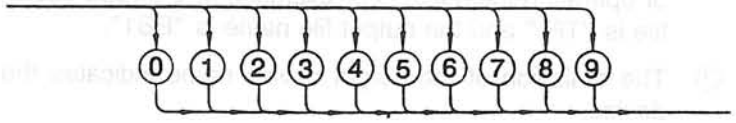
Those enclosed by ○ or ⬭ indicates a character itself

Those enclosed by ▭ indicates the defined item

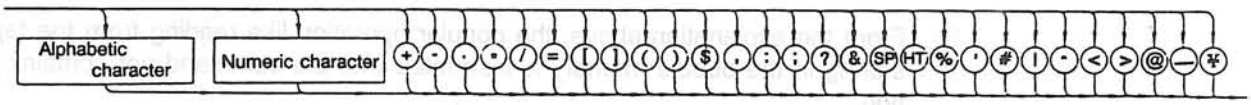
(1) Alphabetic characters



(2) Numeric characters

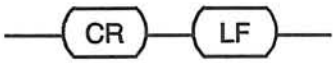


(3) Basic symbols



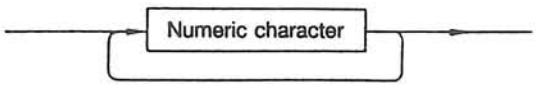
Note: HT ... tabulation, SP ... space

(4) EOB (End of Block)



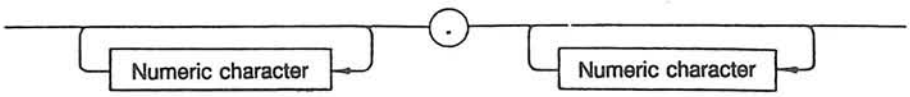
Note: For the EIA code, only (CR) *is used.*
CR .... carriage return
LF ..... line feed

(5) Integers without a sign



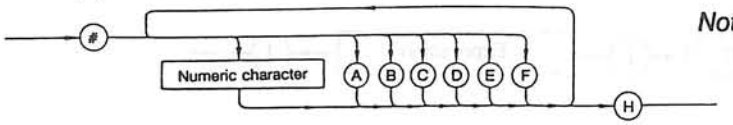
Note: Numeric characters should be written in up to eight digits.

(6) Real numbers without a sign



Note: A numeric character should be written in up to eight digits. The numeric character which has digits only at the right to a decimal point, allowable maximum number of digits is seven.

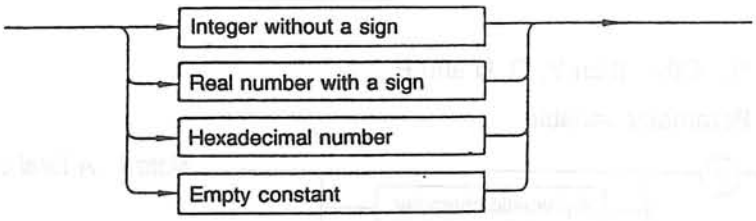
(7) Hexadecimal number



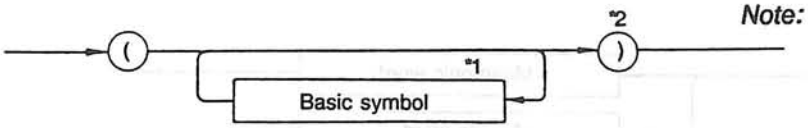
Note: The allowable maximum hexadecimal number is 99999999 when converted into a decimal number.

(8) Empty constants

(9) Constants without a sign

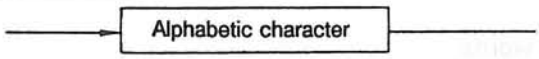


(10) Statement

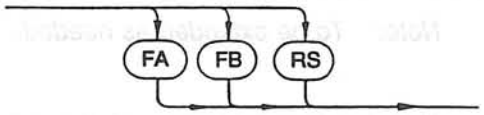


*Note: *1: ")" code should be excluded.*
**2: ")" code at the end of a block is omissible.*

(11) Address character

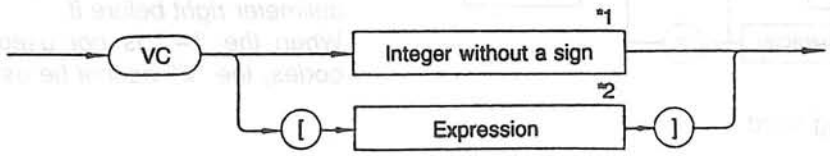


(12) Extended character



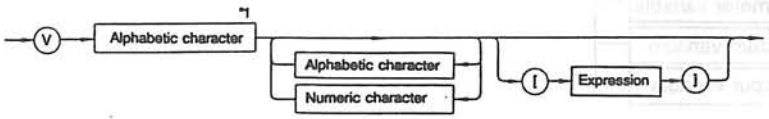
Note: Extended as needed

(13) Common variable



*Note: *1: From 1 to 128*
**2: From 1 to 128 after truncation*

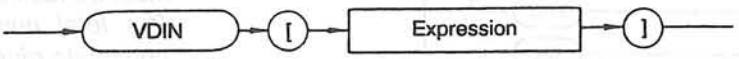
(14) System variable



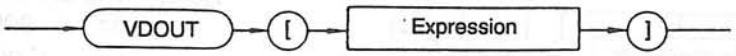
Note: The value of an expression in ( ) is truncated.
The range starts from "1". The maximum value will vary depending on the type of system variables.
The character-string length should be up to 8.

*1: Other than C

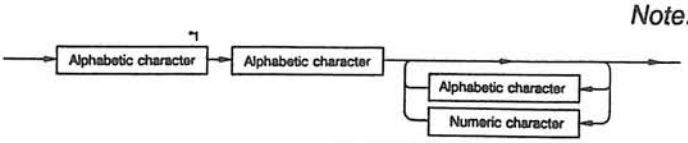
(15) Input variable



(16) Output variable



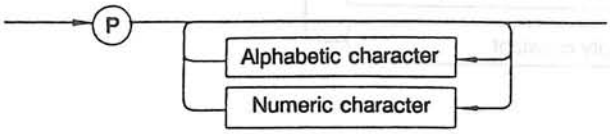
(17) Local variable



Note: A total of four characters may be used. Extended adress characters reserved words (function names mnemonics, etc.) should be excluded.

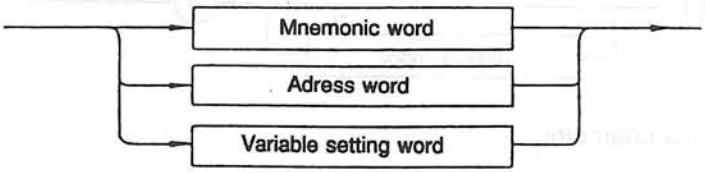
*1: Other than V, O, N and P

(18) Parameter variable



Note: A total of four characters may be used.

(19) Words

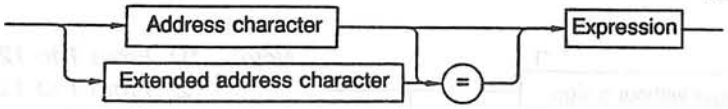


(20) Mnemonic words

LAA I ARC I GRDX I GRDY I DGRDX I DGRDY I SQRX I SQRY I BHC I OMIT I RSTRT I FMILR I FMILF I PMIL I PMILR I RMILI I RMILO I MSG I NMSG I NCYL I COPY I COPYE I

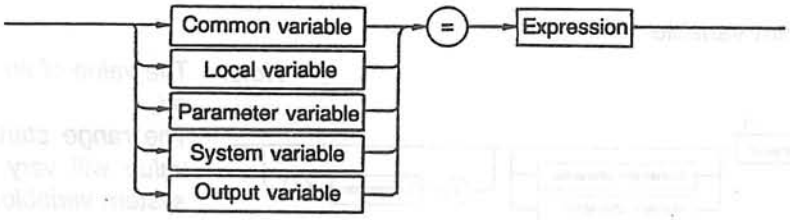
Note: To be extended as needed.

(21) Address word

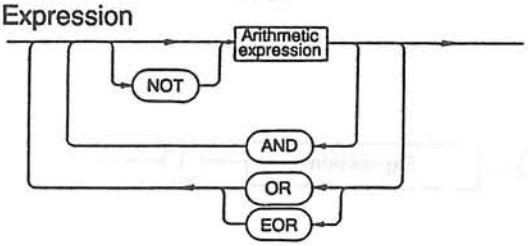


Note: When an expression begins with an alphabetic character, place either the "=" or a delimeter right before it.
When the "=" is not used in G and M codes, the "#" cannot be used.

(22) Variable setting word

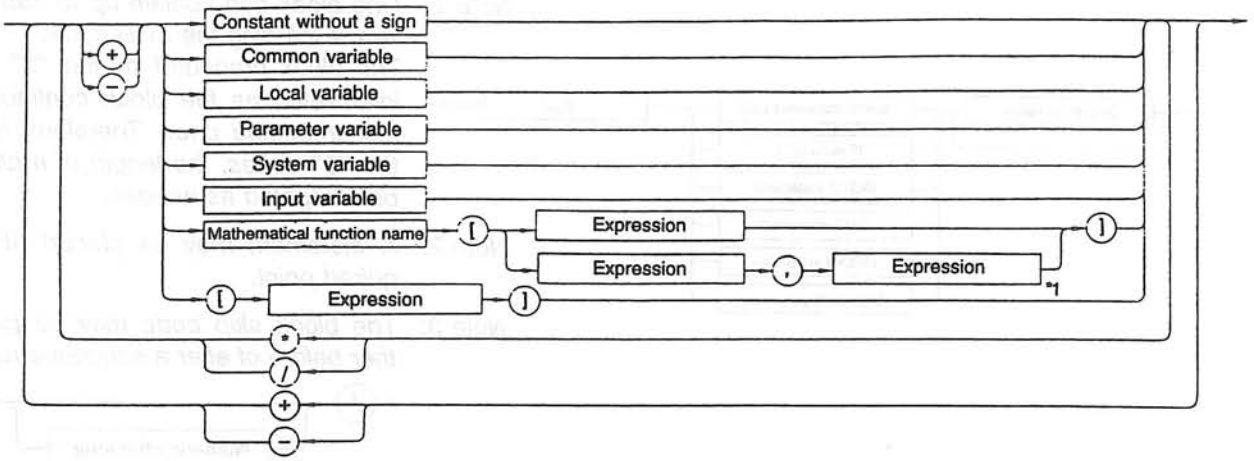


(23) Expression



Note: The total number of elements in a block, including data and operation symbols, must be less than 126.
The total number of data variables and constants must be less than 63.

(24) Arithmetic expression



*1: For ATAN2 and MOD

Note: Each of the elements may be separated by a milliondelimiter.

The maximum number of unsolved elements are as follows, when calculation of an expression is made sequentially from the left:

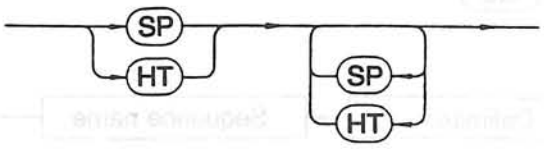
Operation symbol 15

Data 8

(25) Mathematical function names

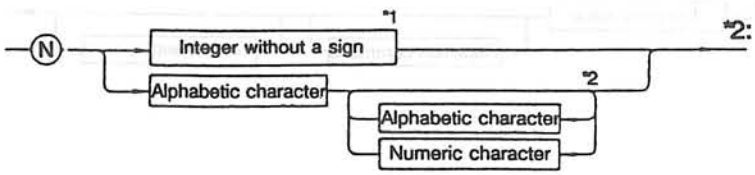
SIN I COS I TAN I ATAN I ATAN2 I SQRT I ABS I BIN I BCD I ROUND I FIX I FUP I DROUND I DFIX I DRUP I MOD I

(26) Delimiter



Note: When several items which begin with an alphabetic character continue, place a delimiter.
A statement (..) may be used as a delimiter.

(27) Sequence names

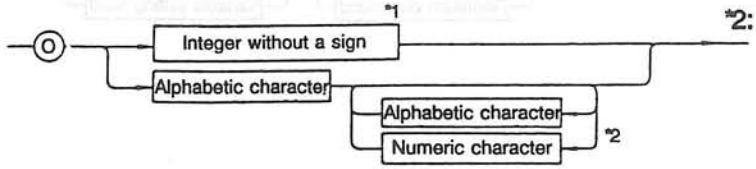


*1: This is called a sequence number

*2: This is called a sequence label.

Note: Sequence names consist of up to four digits following "N". "NOEX", "NCYL" AND "NOT are exceptions.

(28) Program names

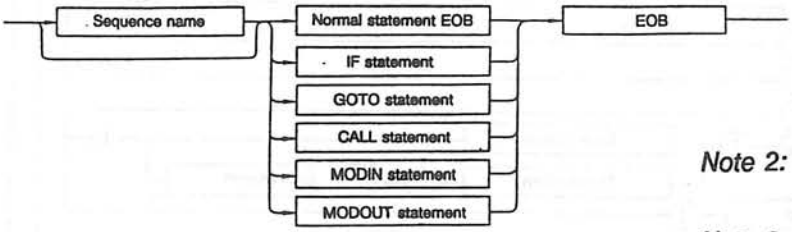


*1: This is called a program number.

*2: This is called a program label.

Note: Sequence names consist of up to four digits following "O". "OR" is an exception.

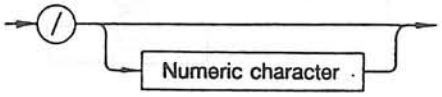
(29) Blocks



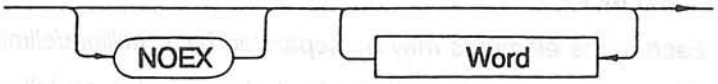
Note 1: One block can contain up to 158 characters, excluding the EOB code. The block preceded by the "$" code is interpreted as the block continuing from the preceding one. Therefore, by giving the "$" codes, the length of a block may be elongated as needed.

Note 2: A statement may be placed at any required point.

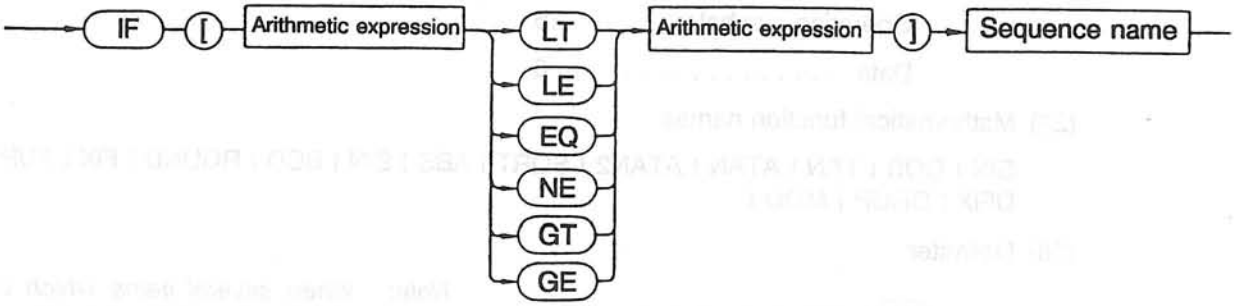
Note 3: The block skip code may be placed either before of after a sequence name.



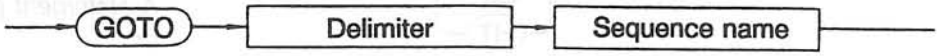
(30) Normal statements



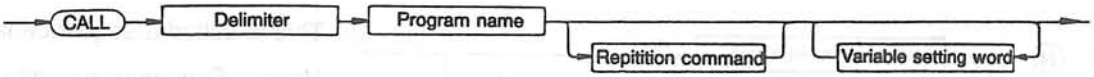
(31) IF statement



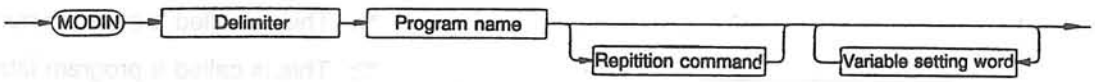
(32) GOTO statement



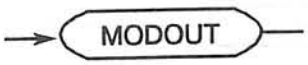
(33) CALL statement



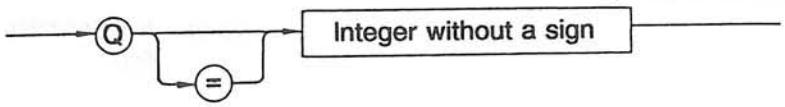
(34) MODIN statement



(35) MODOUT statement

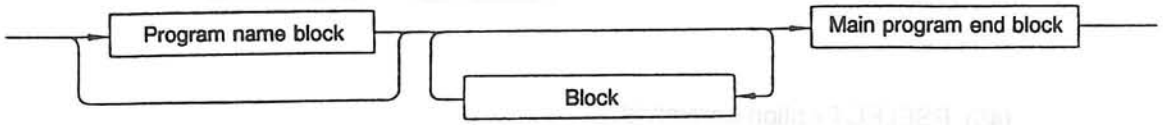


(36) Repetition command

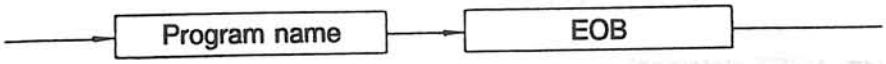


*1: From 1 to 9999

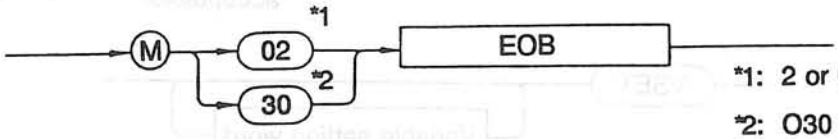
(37) Main program



(38) Program name block



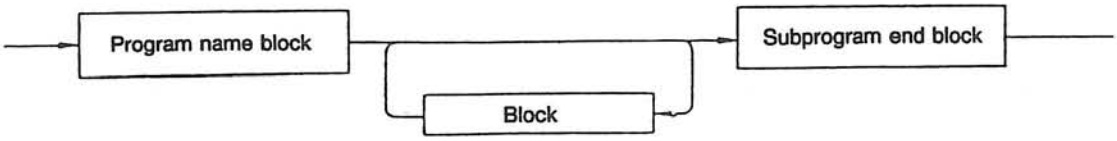
(39) Main program end block



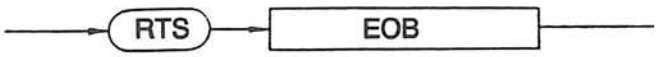
*1: 2 or 002 is also acceptable.

*2: O30 is also acceptable.

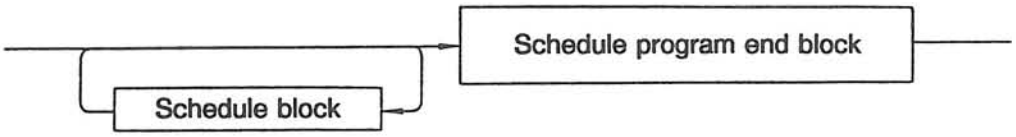
(40) Subprogram



(41) Subprogram end block

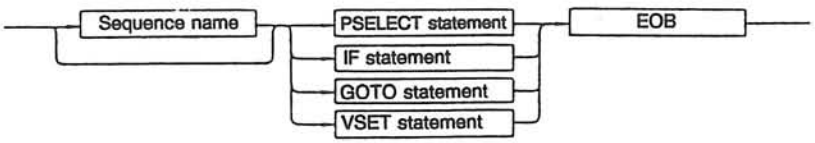


(42) Schedule program



(43) Schedule block

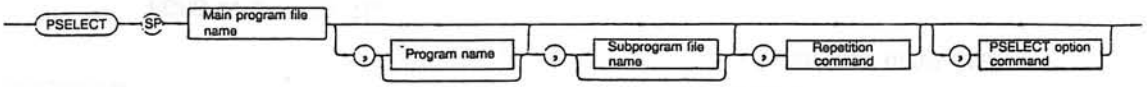
Note: A statement (..) may be given. However, a block containing only a statement, an empty block is not allowed.



(44) Schedule program end block

(45) PSELECT statement

Note: No delimiter is allowed where not specified.

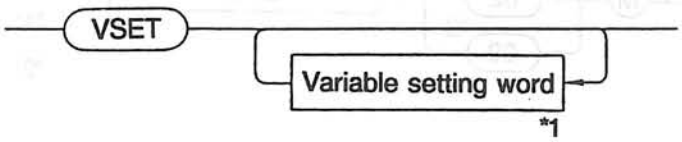


(46) PSELECT option command

Note: Expandable

(47) VSET statement

*1: Local variable and parameter variable are not acceptable.



SECTION 14 APPENDIX

1. EIA/ISO Code Table

EIA Code										ISO Code										Remarks	HEX
Channel / Character	8	7	6	5	4		3	2	1	Channel / Character	8	7	6	5	4		3	2	1		
0			○			∘				0			○	○		∘				Numerical character 0	30
1						∘			○	1	○		○	○		∘			○	Numerical character 1	B1
2						∘		○		2	○		○	○		∘		○		Numerical character 2	B2
3				○		∘		○	○	3			○	○		∘		○	○	Numerical character 3	33
4						∘	○			4	○		○	○		∘	○			Numerical character 4	B4
5				○		∘	○		○	5			○	○		∘	○		○	Numerical character 5	35
6				○		∘	○	○		6			○	○		∘	○	○		Numerical character 6	36
7						∘	○	○	○	7	○		○	○		∘	○	○	○	Numerical character 7	B7
8					○	∘				8	○		○	○	○	∘				Numerical character 8	B8
9				○	○	∘			○	9			○	○	○	∘			○	Numerical character 9	39
A		○	○			∘			○	A		○				∘			○	Alphabetical character A	41
B		○	○			∘		○		B		○				∘		○		Alphabetical character B	42
C		○	○	○		∘		○	○	C	○	○				∘		○	○	Alphabetical character C	C3
D		○	○			∘	○			D		○				∘	○			Alphabetical character D	44
E		○	○	○		∘	○		○	E	○	○				∘	○		○	Alphabetical character E	C5
F		○	○	○		∘	○	○		F	○	○				∘	○	○		Alphabetical character F	C6
G		○	○			∘	○	○	○	G		○				∘	○	○	○	Alphabetical character G	47
H		○	○		○	∘				H		○			○	∘				Alphabetical character H	48
I		○	○	○	○	∘			○	I	○	○			○	∘			○	Alphabetical character I	C9
J		○		○		∘			○	J	○	○			○	∘		○		Alphabetical character J	CA
K		○		○		∘		○		K		○			○	∘		○	○	Alphabetical character K	4B
L		○				∘		○	○	L	○	○			○	∘	○			Alphabetical character L	CC
M		○		○		∘	○			M		○			○	∘	○		○	Alphabetical character M	4D
N		○				∘	○		○	N		○			○	∘	○	○		Alphabetical character N	4E
O		○				∘	○	○		O	○	○			○	∘	○	○	○	Alphabetical character O (program number or program name)	CF
P		○		○		∘	○	○	○	P		○		○		∘				Alphabetical character P	50
Q		○		○	○	∘				Q	○	○		○		∘			○	Alphabetical character Q	D1
R		○			○	∘			○	R	○	○		○		∘		○		Alphabetical character R	D2
S			○	○		∘		○		S		○		○		∘		○	○	Alphabetical character S	53

EIA Code										ISO Code										Remarks	HEX
Channel / Character	8	7	6	5	4		3	2	1	Channel / Character	8	7	6	5	4		3	2	1		
T			○			∘		○	○	T	○	○		○		∘	○			Alphabetical character T	D4
U			○	○		∘	○			U		○		○		∘	○		○	Alphabetical character U	55
V			○			∘	○		○	V		○		○		∘	○	○		Alphabetical character V	56
W			○			∘	○	○		W	○	○		○		∘	○	○	○	Alphabetical character W	D7
X			○	○		∘	○	○	○	X	○	○		○	○	∘				Alphabetical character X	D8
Y			○	○	○	∘				Y		○		○	○	∘			○	Alphabetical character Y	59
Z			○		○	∘			○	Z		○		○	○	∘		○		Alphabetical character Z	5A
+		○	○	○		∘				+			○		○	∘		○	○	Plus Sign	2B
−		○				∘				−			○		○	∘	○		○	Minus Sign	2D
/			○	○		∘			○	/	○		○		○	∘	○	○	○	Block Delete or Division Symbol	5F
BLANK						∘				NULL						∘				Null	00
SPACE				○		∘				SPACE	○		○			∘				Space*	C0
TAB			○	○	○	∘	○	○		HT					○	∘			○	Tab	09
ER					○	∘		○	○	%	○		○			∘	○		○	Program Start	C5
CR/EOB	○					∘				NL/LF					○	∘		○		End of Block	0A
						∘				CR	○				○	∘	○		○	Carriage Return	8D
.		○	○		○	∘		○	○	.			○		○	∘	○	○		Period	2E
,			○	○	○	∘		○	○	,	○		○		○	∘	○			Comma	AC
DEL		○	○	○	○	∘	○	○	○	DEL	○	○	○	○	○	∘	○	○	○	Delete*	FF
BS			○		○	∘		○		BS	○				○	∘				Back Space*	88
&					○	∘	○	○		&	○		○			∘	○	○		Ampersnad	A6
						∘										∘					
(				○	○	∘		○		(			○		○	∘				Contol Out (comment start)	28
)		○			○	∘		○		)	○		○		○	∘			○	Control In (comment end)	A9
										$			○			∘	○			Dollar Sign (continuous block)	24
										*	○		○		○	∘		○		Multiplication Symbol	AA
										=	○		○	○	○	∘	○		○	Equal Sign	BD
										?			○	○	○	∘	○	○	○	Question Mark	3F
										[	○	○		○	○	∘		○	○	Bracket, Left	DB
										]	○	○		○	○	∘	○		○	Bracket, Right	DD
										:			○	○	○	∘		○		Colon (program number or program name)	3A
										;	○		○	○	○	∘		○	○	Semicolon	BB

* Ignored when used as cutting data.

2. List of G Codes

G Code	Group	Functions
G00○	1	Positioning
G01○		Linear interpolation
G02		Circular interpolation - Helical cutting (CW)
G03		Circular interpolation - Helicla cutiing (CCW)
G04◎	2	Dwell
G05		
G06		
G07		
G08		
G09◎	18	Exact stop
G10*	3	Cancel of G11
G11		Parallel and rotational shift of coordinate system
G12		
G13		
G14		Axis designation/cancel
G15	4	Selection of work coordinate system (Modal)
G16◎		Selection of work coordinate system (One-shot)
G17○	5	Plane selection: XY
G18○		Plane selection: ZX
G19○		Plane selection: YZ
G20◎	15	Inch input confirmation
G21◎		Metric input confirmation
G22○	6	Programmable stroke limit ON
G23○		Programmable stroke limit cancel
G24		
G25		

G Code	Group	Functions
G26		
G27		
G28		
G29		
G30◎	16	Positioning to home position
G31◎	17	Skip function
G32		
G33		
G34		
G35		
G36		
G37		
G38		
G39		
G40*	7	Cutter radius compensation cancel
G41		Cutter radius compensation, left
G42		Cutter radius compensation, right
G43*	8	Three dimensional compensation cancel
G44		Three dimensional compensation ON
G45		
G46		
G47		
G48		
G49		
G50*	9	Enlargement and reduction of geometry cancel
G51		Enlargement and reduction of geometry ON

* : Has already been set when power supply is turned on.
◎ : Effective in a specified block.
○ : May be set by an initial condition parameter.

G Code	Group	Functions
G52		
G53○	10	Tool length offset cancel
G54		Tool lenght offset, X-axis
G55		Tool lenght offset, Y-axis
G56○		Tool length offset, Z-axis
G57		Tool length offset, 4th-axis
G58		Tool lenght offset, 5th-axis
G59		Tool length offset, 6th-axis
G60	1	One-directional positioning
G61	14	Exact stop mode
G62	19	Programmable mirror image function
G63		
G64*	14	Cutting mode
G65		
G66		
G67		
G68		Cancel of G69
G69		Setting of 3-D coordinate system conversion
G70		
G71	11	Designation of return level for M53
G72		
G73	11	Fixed cycle, High speed deep hole drilling cycle
G74		Fixed cycle, Reverse tapping
G75		
G76	11	Fixed cycle, Fine boring
G77		
G78		
G79		
G80*	11	Fixed cycle, Model cancel
G81		Fixed cycle, Drill/spot boring
G82		Fixed cycle, Drill/counter boring
G83		Fixed cycle, Deep hole drilling cycle
G84		Fixed cycle, Tapping
G85		Fixed cycle, Boring
G86		Fixed cycle, Boring
G87		Fixed cycle, Back boring
G88		
G89	11	Fixed cycle, Boring
G90○	12	Absolute dimensioning
G91○		Incremental dimensioning
G92	20	Setting of work coordinate system
G93		
G94○	13	Feed per min.
G95○		Feed per rev.
G96		
G97		
G98		
G99		

* : Has already been set when power supply is turned on.
◎ : Effective in a specified block.
○ : May be set by an initial condition parameter.

3. List of Mnemonic Codes

Mnemonic Code	Group	Functions
NOEX	34	Specification of the sequence not executed
CALL	27	Subprogram, Simple call
RTS		Subprogram, End code
MODIN		Subprogram, Call after positioning mode ON
MODOUT		Subprogram, Call after positioning mode OFF
GOTO	28	Branch Command, Uncondtitional jump
IF		Branch Command, Conditional jump (6 kinds)
RTMCR		Macro processing end code (used only in system)
RTMDI		MDI processing end code (used only in system)
OMIT	29	Coordinate Calculation Function, Omit
RSTRT		Coordinate Claculation Function, Restart
LAA	30	Coordinate Claculation Function, Line at angle
ARC		Coordinate Calculation Function, Arc
GRDX		Coordinate Calculation Function, Grid X
GRDY		Coordinate Calculation Function, Grid Y
DGRDX		Coordinate Calculation Function, Double grid X
DGRDY		Coordinate Calculation Function, Douuble grid Y
SQRX		Coordinate Calculation Function, Square X
SQRY		Coordinate Calculation Function, Square Y
BHC		Coordinate Calculation Function, Bolt hole circle
FMILR	31	Area Machining, Face milling (Rough)
FMILF		Area Machining, Face milling (Finish)
PMIL		Area Machining, Pocket milling (Zigzag)
PMILR		Area Machining, Pocket milling (Spiral)
RMILO		Area Machining, Round milling (Out)
RMILI		Area Machining, Round milling (In)
MSG	35	Message display
NMSG		Restoring original display
NCYL	36	Fixed cycle, No cycle movement
COPY	39	Copy, Initial value of local coordinate system for parallel shift rotation
COPYE		Copy end, Incremental value of local coordinate system for parallel shift or rotation
CHFC		Arbitrary-angle chamfering (Chamfering)
CHFR		Arbitrary-angle chamfering (Rounding)

4. List of M Codes

M Code	Group	Function	Against Axis	Movement Modal /One-shot	Remarks
M00	1	Program stop	After	one-shot	Spindle coolant stops; selectable with parameter.
M01		Optional stop	After	one-shot	
M02	18	End of program	After	one-shot	NC reset
M03	2	Spindle CW	At the same time	modal	
M04		Spindle CCW	At the same time	modal	
M05		Spindle stop	After	modal	
M06	3	Vertical spindle tool change	After	one-shot	
M07	8	Oil mist coolant ON	At the same time	modal	
M08	10	Coolant pump ON	At the same time	modal	
M09	24	Coolant system OFF (M07, 08, 12, 50, 51, 59 OFF)	After	modal	
M10	30	A-axis clamp	At the same time	modal	
M11		A-axis unclamp	At the same time		
M12	22	Chip air blow ON	At the same time	modal	
M15	5	4th-axis - rotary index table CW	At the same time	modal	
M16		4th-axis - rotary index table CCW	At the same time	modal	
M17	7	Indexing head index CCW At the same time one-shot	At the same time	one-shot	command effective for only M73 - M76.
M19	2	Spindle orientation (forward)	After	modal	
M20	31	B-axis clamp	After	modal	
M21		B-axis unclamp	After	modal	
M22	32	Y-axis clamp	After	modal	
M23		Y-axis unclamp	After	modal	
M24	33	Z-axis clamp	After	modal	
M25		Z-axis unclamp	After	modal	
M26	35	C-axis clamp	After	modal	
M27		C-axis unclamp	After	modal	
M30	18	End of tape	After	one-shot	NC reset
M32	38	Splash guard door close	At the same time	modal	
M33		Splash guard door open	At the same time	modal	

M Code	Group	Function	Against Axis	Movement Modal /One-shot	Remarks
M40	11	High/middle-high/middle-low/low range	At the same time	modal	Spindle gears are automatically determined by RPM command
M41		High/middle-high/middle-low range	At the same time	modal	
M42		High/middle-high range	At the same time	modal	
M43		High range	At the same time	modal	
M44		AAC(F) next attachment clear	At the same time	one-shot	F: Floor type T: Table type
M45		AAC(F) change preparation	At the same time	one-shot	
M46		AAC(F) no next attachment	At the same time	one-shot	
M47		AAC(T) no next attachment	At the same time	one-shot	
M48	38	AAC(T) next attachment clear	At the same time	one-shot	
M49		AAC(T) change preparation	At the same time	one-shot	
M50	23	Through-the-tool coolant, low pressure ON	At the same time	modal	
M51		Through-the-tool coolant, high pressure ON	At the same time	modal	
M52	12	Return level in fixed cycle: upper limit	At the same time	one-shot	
M53	13	Return level in fixed cycle: upper limit	At the same time	modal	
M54		Return level in fixed cycle: point R level	At the same time	modal	
M57	34	W-axis clamp	After	modal	
M58		W-axis unclamp	After	modal	
M59	25	Chip air blow ON	At the same time	modal	
M60	4	Dallet change command	After	one-shot	
M62		Vertical spindle tool change preparation	After	modal	
M63	21	ATC no next tool	At the same time	one-shot	
M64		Next tool return cycle	At the same time	one-shot	
M65		ATC ready	At the same time	one-shot	
M66	14	Continuous tool change between the vertical and horizontal spindles (same tool)	At the same time	one-shot	
M67		Continuous tool change between the vertical and horizontal spindles (different tool)	At the same time	one-shot	
M68		Vertical spindle tool clamp	After	modal	
M69		Vertical spindle tool unclamp	After	modal	

M Code	Group	Function	Against Axis	Movement Modal /One-shot	Remarks
M70	3	Manual tool change	After	one-shot	
M71		Manual attachment tool change	After	one-shot	
M72		Horizontal spindle tool change preparation	After	modal	
M73	15	Swivel head, front position	After	one-shot	
M74		Swivel head, left position	After	one-shot	
M75		Swivel head, rear position	After	one-shot	
M76		Swivel head, right position	After	one-shot	
M77	3	Horizontal spindle tool change	After	one-shot	
M78		Horizontal spindle tool clamp	After	modal	
M79		Horizontal spindle tool unclamp	After	modal	
M81	27	Automatic W-axis positioning 1	After	one-shot	
M82		Automatic W-axis positioning 2	After	one-shot	
M83		Automatic W-axis positioning 3	After	one-shot	
M84		Automatic W-axis positioning 4	After	one-shot	
M85		Automatic W-axis positioning 5	After	one-shot	
M87		Oil mist/air blow ON	At the same time	one-shot	Effective for M90, 91 and 98
M88		Dust collector ON	At the same time	modal	
M89		Dust collector OFF	At the same time	modal	
M90		Vertical spindle oil mist cycle mode ON	At the same time	modal	
M91		Tap-drill hole chip air blow cycle mode ON	At the same time	modal	Turned OFF by M90
M98		Horizontal spindle oil mist cycle mode ON	At the same time	modal	

M Code	Group	Function	Against Axis	Movement Modal /One-shot	Remarks
M101	4	Pallet positioning 1	After	one-shot	
M102		Pallet positioning 2	After	one-shot	
M103		Pallet positioning 3	After	one-shot	
M104		Pallet positioning 4	After	one-shot	
M105		Pallet positioning 5	After	one-shot	
M106		Pallet positioning 6	After	one-shot	
M107		Pallet positioning 7	After	one-shot	
M108		Pallet positioning 8	After	one-shot	
M109		Pallet positioning 9	After	one-shot	
M110		Pallet positioning 10	After	one-shot	
M111		Pallet positioning 11	After	one-shot	
M112		Pallet positioning 12	After	one-shot	
M115	6	5th-axis - rotary table CW	At the same time	modal	
M116		5th-axis - rotary table CCW	At the same time	modal	
M118		Spindle orientation (reverse)	After	modal	
M119		Spindle orientation (forward/reverse)	After	modal	
M120		Work shower ON	At the same time	modal	
M130	9	Spindle rotation condition for cutting feed OFF	At the same time	modal	Cutting feed (G01, G02, G03) allowed without spindle rotation
M131		Spindle rotation condition for cutting feed ON	At the same time	modal	Cutting feed (G01, G02, G03) not allowed without spindle rotation
M132	37	Single block ineffective	At the same time	modal	
M133		Single block effective	At the same time	modal	
M134	16	Spindle speed override ineffective	At the same time	modal	
M135		Spindle speed override effective	At the same time	modal	
M136	17	Feedrate override ineffective	At the same time	modal	
M137		Feedrate overide effective	At the same time	modal	
M138	36	Dry run ineffective	At the same time	modal	
M139		Dry run effective	At the same time	modal	

M Code	Group	Function	Against Axis	Movement Modal /One-shot	Remarks
M140	39	Slide hold ineffective	At the same time	modal	
M141		Slide hold effective	At the same time	modal	
M142		Spindle overload detection ineffective	At the same time	modal	
M143		Spindle overload detection effective	At the same time	modal	
M144	19	Touch sensor advance	After	modal	
M145		Touch sensor retract	After	modal	
M150	28	Coolant group specification - vertical spindle	At the same time	modal	
M151		Coolant group specification - horizontal spindle	At the same time	modal	
M152		Coolant group specification - 3rd group	At the same time	modal	
M153		Coolant group specification - 4th group	At the same time	modal	
M154	20	Sensor air blow OFF	After	modal	
M155		Sensor air blow ON	At the same time	modal	
M157		AAC (2 st.) no next tool	After	one-shot	
M158		AAC (2 st.) next tool clear	After	one-shot	
M159		AAC (2 st.) tool change preparation	After	one-shot	
M160	4	PPC pallet loading	After	one-shot	
M161		PPC pallet unloading	After	one-shot	
M163		Long tool - no next tool	At the same time	one-shot	
M165		Long tool - tool change preparation	At the same time	one-shot	Safety cover specification of next tool
M166		ATC active tool return mode specification	At the same time	one-shot	
M170	3	AAC (F) attachment change	After	one-shot	F: Floor type
M171		AAC (T) attachment change	After	one-shot	T: Table type
M172		Long tool - tool change command	After	one-shot	
M173		AAC (2 st.) - attachemtn change command	After	one-shot	Column type
M176		Dust collection mode - air blow	At the same time	one-shot	
M177	3	Angualr attachment tool change	After	one-shot	AT-ATC
M178	4	Dust collection mode - ON	At the same time	modal	
M179		Dust collection mode - OFF	At the same time	modal	

M Code	Group	Function	Against Axis	Movement Modal /One-shot	Remarks
M181	38	External M signal	At the same time	one-shot	When bit 1 of machine user parameter (bit) No. 3 is set at "1", an external M signal is output after the completion of axis movement
M182		External M signal	At the same time	one-shot	
M183		External M signal	At the same time	one-shot	
M184		External M signal	At the same time	one-shot	
M185		External M signal	At the same time	one-shot	
M186		External M signal	At the same time	one-shot	
M187		External M signal	At the same time	one-shot	
M188		External M signal	At the same time	one-shot	
M190		Automatic W-axis positioning 1	After	one-shot	For automatic W-axis positioning 10 points specification
M191		Automatic W-axis positioning 2	After	one-shot	
M192		Automatic W-axis positioning 3	After	one-shot	
M193		Automatic W-axis positioning 4	After	one-shot	
M194		Automatic W-axis positioning 5	After	one-shot	
M195		Automatic W-axis positioning 6	After	one-shot	
M196		Automatic W-axis positioning 7	After	one-shot	
M197		Automatic W-axis positioning 8	After	one-shot	
M198		Automatic W-axis positioning 9	After	one-shot	
M199		Automatic W-axis positioning 10	After	one-shot	

The commanded state of the following M codes may be displayed in the M code dield (BLOCK).

- M03, 04, 05, 19
- M06, 77
- M07
- M10, 11
- M15, 16
- M17
- M20, 21
- M22, M23
- M24, M25
- M40 - 43
- M53, 54
- M57, 58
- M63, 64, 65
- M73 - 76
- M81 - M85
- M60, 101 - 112, 160, 161
- M115, 116
- M130 - 131
- M132, 133
- M134, 135
- M136, 137
- M138, 139
- M140, 141
- M150 - 153
- M08
- M12
- M13, 14, 18, 32 - 37, 48, 49, 181 - 185
- M26, 27
- M30
- M50, 51
- M52
- M59
- M66, 67
- M144, 145
- M154, 155
- M201 - 210
- M254

Note: The M code indication may include up to 26 modal states

LIST OF PUBLICATIONS

Publication No.	Date	Edition	
3336-E	July 1989	1st	A707 *
3336-E-R1	September 1991	2nd	A709 *
3336-E-R2	December 1991	3rd	
3336-E-R3	July 1993	4th	A709BM

This manual may be at variance with the actual product due to specification or design changes.
Please also note that specifications are subject to change without notice.
If you require clarification or further explanation of any point in this manual, please contact your OKUMA representative.