

FANUC SYSTEM 6T-MODEL B

OPERATOR'S MANUAL

CONTENTS

I. GENERAL	1
1. INTRODUCTION	3
2. NOTES ON READING THIS MANUAL	3
II. SPECIFICATIONS	5
1. BASIC SPECIFICATIONS	7
2. BASIC OPTIONS	12
3. ADDITIONAL OPTION	14
III. PROGRAMMING	21
1. WHAT IS PROGRAMMING	23
2. PROGRAM COMPOSITION	24
2.1 Block	25
2.2 Word	25
2.3 Input Format	27
2.4 Decimal Point Programming	28
2.5 Maximum Programmable Dimensions	29
2.6 Program Number	30
2.7 Sequence Number	30
2.8 Optional Block Skip	31
3. DIMENSION WORD	32
3.1 Controlled Axis	32
3.1.1 Coordinate axes and symbol of motion	32
3.2 Increment System	33
3.2.1 Least input increment (Input unit)	33
3.2.2 Least command increment (Output unit)	33
3.2.3 Multiplication of input unit by ten	33
3.2.4 Input unit 1/10	34
3.3 Maximum Stroke	34
3.4 Program Zero Point and Coordinate System	34
3.5 Coordinate System and Start Point	35
3.6 Reference Point	38

3.7	Absolute and Incremental Programming	39
3.8	Diameter and Radius Programming	39
4.	FEED FUNCTION	41
4.1	Rapid Traverse Rate	41
4.2	Feedrate	41
4.3	Lead of Thread	42
4.4	Feedrate Reduction 1/10	43
4.5	Automatic Acceleration and Deceleration	43
5.	PREPARATORY FUNCTION (G FUNCTION)	45
5.1	Positioning (G00)	46
5.2	Linear Interpolation (G01)	47
5.3	Chamfering and Corner R	49
5.4	Direct Input of Drawing Dimensions	50
5.5	Circular Interpolation (G02, G03)	55
5.6	Feedrate Sine Curve Control	58
5.7	Thread Cutting (G32, G34)	59
5.7.1	Straight, taper thread cutting (G32)	59
5.7.2	Variable-lead thread cutting (G34)	63
5.7.3	Continuous threading	63
5.8	Automatic Reference Point Return (G27 ~ G30)	64
5.8.1	G27-Reference point return check	64
5.8.2	G28 Automatic return to reference point	64
5.8.3	G29 Automatic return from reference point	65
5.8.4	G30 2nd reference point return	65
5.9	Dwell (G04)	66
5.10	Programming of Absolute Zero Point (G50)	66
5.10.1	Command used to set the work coordinate system	66
5.10.2	Shifting the work coordinate system	67
5.10.3	Automatic coordinate system setting	68
5.10.4	Work coordinate system shifting	69
5.10.5	Direct measured value input in work coordinate system shifting	71
5.10.6	Mirror image for double turrets (G68, G69)	72
5.11	Inch/Metric Conversion (G20, G21)	72
5.12	Feedrate Command Switching (G98, G99)	73
5.13	Constant Surface Speed Control (G96, G97)	73
5.14	Stored Stroke Limit (G22, G23)	73
5.15	Skip Function (G31)	77
6.	COMPENSATION FUNCTION	79
6.1	Tool Offset	79
6.1.1	Basic tool offset	79
6.1.2	T code for tool offset	80
6.1.3	Tool selection	80
6.1.4	Tool offset number	80
6.1.5	Offset	81
6.1.6	Offset vector	81
6.1.7	Offset cancel	81
6.1.8	Program example	81
6.1.9	Only T code	83
6.2	Tool Geometry Offset and Tool Wear Offset	83
6.2.1	T code for tool offset	83
6.2.2	Tool offset number	84
6.2.3	Offset	85
6.2.4	Offset vector	85
6.2.5	Offset cancel	85
6.3	Tool Nose Radius Compensation	87
6.3.1	Imaginary tool nose	87
6.3.2	Direction of imaginary tool nose	90
6.3.3	Setting the tool nose radius compensation value	91

6.3.4	Work position and move command	92
6.3.5	Notes in tool nose radius compensation	97
6.4	Detailed Description of Tool Nose Radius Compensation	102
6.4.1	Tool nose R center offset vector	102
6.4.2	G40, G41, G42	102
6.5	Input of Offset Value by Programming (G10)	133
6.6	Direct Input of Measured Offset Value	134
6.7	Automatic Tool Offset	135
6.8	External Tool Compensation	137
7.	MACHINING CYCLE FUNCTION	138
7.1	Canned Cycle (G90, G92, G94)	138
7.1.1	Cutting cycle A, G90	138
7.1.2	Thread cutting cycle, G92	139
7.1.3	Cutting cycle B, G94	142
7.1.4	Canned cycle usage	144
7.2	Multiple Repetitive Cycles (G70 ~ G76)	145
7.2.1	Stock removal in turning (G71)	145
7.2.2	Stock removal in facing (G72)	148
7.2.3	Pattern repeating (G73)	149
7.2.4	Finishing cycle (G70)	150
7.2.5	Peck drilling along the Z axis (G74)	155
7.2.6	Grooving in X axis (G75)	155
7.2.7	Thread cutting (G76)	156
7.2.8	Notes on multiple repetitive cycle (G70 ~ G76)	157
8.	SPINDLE SPEED FUNCTION (S FUNCTION), TOOL FUNCTION (T FUNCTION) MISCELLANEOUS FUNCTION (M FUNCTION)	159
8.1	Spindle Speed Function (S Function)	159
8.1.1	S 2-digit	159
8.1.2	S 4-digit	159
8.2	Constant Surface Speed Control	159
8.2.1	Command	160
8.2.2	Spindle speed override	160
8.2.3	Clamping of maximum spindle speed	160
8.2.4	Rapid traverse (G00)	160
8.2.5	Example	161
8.2.6	Relation between spindle speed and surface speed	163
8.3	Tool Function (T Function)	163
8.4	Miscellaneous Function (M Function)	164
8.4.1	M02, M30: End of program	164
8.4.2	M00: Program stop	164
8.4.3	M01: Optional stop	164
8.4.4	M98: Calling of subprogram	164
8.4.5	M99: End of subprogram	164
9.	SUBPROGRAMS	165
9.1	Preparation of Subprogram	165
9.2	Subprogram Execution	165
9.3	Notes on Subprogram Control	166
10.	CUSTOM MACRO A AND B	169
10.1	General	169
10.2	Variables	172
10.2.1	Variable expressions	172
10.2.2	Quotation variables	172
10.2.3	Undefined variables	172
10.2.4	Display and setting variable values	173
10.3	Types of Variables	173
10.3.1	Local variable #1 to #33	173
10.3.2	Common variable #100 to #149, #500 to #509	174
10.3.3	System variables (For custom macro B)	174

10.4	Arithmetic Commands	183
10.4.1	Definition and substitution of variables	183
10.4.2	Addition arithmetic	183
10.4.3	Multiplication arithmetic (User macro B option)	183
10.4.4	Functions (User macro B option)	183
10.4.5	Combination of arithmetic operations	184
10.4.6	Modification of arithmetic sequence using []	184
10.4.7	Precision	185
10.4.8	Notes on decreased precision	185
10.5	Control Command	186
10.5.1	Divergence (GOTO)	186
10.5.2	Iteration (Custom macro B option)	186
10.6	Creation and Registration of Custom Macro Body	190
10.6.1	Creation of custom macro body	190
10.6.2	Registration of custom macro body	190
10.7	Macro Calls	191
10.7.1	Simple calls	191
10.7.2	Modal calls	193
10.7.3	Multiplex calls	193
10.7.4	Multiplex modal calls	193
10.7.5	Macro call using G codes	194
10.7.6	Subprogram call with M code	194
10.7.7	Subprogram call with T codes	195
10.7.8	Location of decimal point in an argument	195
10.7.9	Difference between M98 (subprogram call) and G65 (custom macro body call)	195
10.8	Relation to Other Functions	196
10.9	Special Codes and Words Used in Custom Macro	197
10.10	Limitations	198
10.11	P/S Alarm Explanation	199
10.12	External Output Commands	199
10.12.1	Open command POPEN	199
10.12.2	Data output commands BPRNT, DPRNT	199
10.12.3	Close command PCLOS	202
10.12.4	Setting required for using this function	202
10.12.5	Cautions	202
10.13	Custom Macro Interrupt Function (Custom Macro B)	203
11.	TOOL LIFE MANAGEMENT	204
11.1	General	204
11.2	Setting The Tool Groups	204
11.3	Specifying Tool Groups in a Machining Program	205
11.4	Execution of Tool Life Management	206
11.4.1	Counting tool life	206
11.4.2	Tool change signal and tool change reset signal	207
11.4.3	Tool skip signal	207
11.4.4	New tool select signal	207
11.5	Display and Input of Tool Data	207
11.5.1	Display and modification of tool group data	207
11.5.2	Display of tool life data during program execution	207
11.5.3	Presetting of tool life counters	208
11.6	Setting of Parameters	208
11.7	Alarm	208
11.8	Notes	208
12.	PROGRAM TAPE COMPOSITION	209
12.1	Paper Tape Used for NC Control	209
12.2	Tape Code	210
12.3	Codes Used in Programming	211
12.4	Program Tape	213
12.4.1	Tape start	215

12.4.2	Leader section	215
12.4.3	Program start	215
12.4.4	Program section	215
12.4.5	Comment section	215
12.4.6	Program end	217
12.4.7	Tape end	217
12.5	Block Composition	217
IV.	OPERATION	219
1.	TURNING THE POWER ON AND OFF	221
1.1	Turning the Power On	221
1.2	Turning the Power Off	221
2.	LOCK KEY	222
3.	TAPE READER	223
3.1	Tape Reader Without Reels	223
3.2	Tape Reader With Reels	224
3.3	Tape Reader Handling	226
4.	OPERATION OF THE OPERATOR'S PANEL	230
4.1	Operator's Panel	230
4.2	Emergency Stop	231
4.3	Mode Select	231
4.4	Operation Relative to Manual Operations	232
4.4.1	JOG feed	232
4.4.2	HANDLE feed	234
4.4.3	INCREMENTAL feed	235
4.4.4	Manual absolute	236
4.5	Manual Reference Point Return	241
4.6	Automatic Operation	242
4.6.1	Starting automatic operation	242
4.6.2	Halting automatic operation	243
4.6.3	Single block	244
4.6.4	Restart after Feed Hold	246
4.6.5	Manual operation during automatic operation	247
4.6.6	MDI operation during automatic operation	247
4.6.7	Optional block skip	247
4.6.8	Feedrate override	248
4.6.9	Dry run	248
4.6.10	Machine lock	248
4.6.11	Display lock	249
4.6.12	Mirror image	249
4.6.13	Rapid traverse override	249
4.6.14	External workpiece number search function	249
4.7	Manual Handle Interruption Function	251
4.7.1	General	251
4.7.2	Handle interruption operation	251
4.7.3	Movement by handle interruption	251
5.	DISPLAY AND OPERATION ON THE CRT CHARACTER DISPLAY MDI & DPL PANEL	253
5.1	Status Indication	254
5.2	Keyed Data Display	255
5.3	Program Number and Sequence Number Display	255
5.4	Alarm Display	256
5.5	Operator Message	256
5.6	Display and Reset of a Current Position	257
5.7	Command Value Display	259
5.8	Setting	260
5.8.1	Display and setting input, output, etc.	260
5.8.2	Display and setting of custom macro variable values	263
5.9	Operation from the MDI	264

5.10	Starting MDI Operation	265
5.11	Reset	265
5.12	Setting and Display of Tool Offset and Tool Nose Radius Compensation Values	266
5.12.1	Absolute input	266
5.12.2	Incremental input	267
5.12.3	When tool offset is divided between tool geometry offset and tool wear offset	267
5.12.4	Setting work coordinate system shift amount	268
5.12.5	Direct measured value input for work coordinate system shift	270
5.12.6	Direct input of tool offset values	272
5.12.7	Input an offset value to the counter (option)	273
5.13	Program Display	274
5.14	Program Number Search	276
5.15	Registering a Program Tape in Memory	276
5.16	Registering One Tape with Programs into Memory	276
5.17	Adding a Program after an Already Registered Program	277
5.18	Collating Programs in Memory and Those on Tape (Option)	277
5.19	Inputting Programs by Keys	278
5.20	Deleting a Program	279
5.21	Deleting All Programs	279
5.22	Punching a Program (Option)	279
5.23	Punching All Programs (Option)	280
5.24	Sequence Number Search	281
5.25	Program Restart	283
5.26	Sequence Number Comparison and Stop	285
5.27	Entering Offset Values from Tape (option)	286
5.28	Punching Offset Values onto Tape (option)	286
5.29	Parameter Display	286
5.30	Program Editing	286
5.30.1	Words scan	287
5.30.2	Word search	288
5.30.3	Search for addresses only	288
5.30.4	Returning the cursor to the Head of a Program	288
5.30.5	Inserting a Word	289
5.30.6	Changing a word	290
5.30.7	Insertion or correction of words, blocks, and character strings	290
5.30.8	Deleting a word	291
5.30.9	Deleting up to an EOB	291
5.30.10	Deleting blocks	292
5.30.11	Memory arrangement	292
5.30.12	Displaying program numbers of all registered programs	292
5.30.13	Custom macro editing	292
5.31	Run Hour Display	294
5.32	Menu Switch Function	295
6.	POSITION DISPLAY BY POSITION DISPLAY UNIT	296
7.	OPERATION OF FANUC CASSETTE	297
7.1	General	297
7.2	Input/Output Operation	297
7.3	What is a File?	297
7.4	Heading of File	297
7.5	Deletion of File	298
7.6	Data Transfer to and from Cassette	299
7.6.1	NC program transfer to and from cassette	299
7.6.2	Offset data transfer to and from cassette	300
7.6.3	NC parameter transfer to and from cassette	300
7.7	Cassette Exchange Request	301
7.8	Cassette Adapter Lamp Conditions	301
7.9	Parameters and Setting on Data Transfer to and from Cassette	302

V. MAINTENANCE	303
1. DAILY MAINTENANCE	305
1.1 Tape Reader Cleaning	305
1.2 Tape Reader Lubrication	307
1.3 Air Filter Cleaning	310
2. FUSE CHECK AND REPLACEMENT	311
2.1 Input Unit	311
2.2 Power Supply for Control Unit (Stabilizer Unit)	312
3. CHECK AND REPLACEMENT OF DC MOTOR BRUSH	313
4. TROUBLESHOOTING	314
4.1 General	314
4.2 Checking Input Voltage, Peripheral Conditions, Operation, Programming, Drives, Machine and Interface Control	314
4.3 NC System Check (No tools required)	316
4.4 NC Status	317
APPENDIXES	319
APPENDIX 1 TAPE CODES USED IN PROGRAMMING	323
APPENDIX 2 G FUNCTION TABLE	325
APPENDIX 3 TABLE OF COMMAND VALUE RANGES	327
APPENDIX 4 NOMOGRAPHS	328
APPENDIX 5 TAPE JOINING	332
APPENDIX 6 PARAMETERS	333
APPENDIX 7 ALARM LIST	364
APPENDIX 8 STATUS WHEN TURNING THE POWER ON, WHEN CLEARED AND WHEN RESET	372
APPENDIX 9 STORED PITCH ERROR COMPENSATION FUNCTION	374
APPENDIX 10 LIST OF OPERATIONS	379
APPENDIX 11 PROGRAM KEY LOCK	382
APPENDIX 12 CUSTOM MACRO INTERRUPT FUNCTION	384

303	V. MAINTENANCE
305	1. DAILY MAINTENANCE
305	1.1 Tape Reader Cleaning
307	1.2 Tape Reader Lubrication
310	1.3 Air Filter Cleaning
311	2. FUSE CHECK AND REPLACEMENT
311	2.1 Input Unit
312	2.2 Power Supply for Control Unit (Stabilizer Unit)
313	3. CHECK AND REPLACEMENT OF DC MOTOR BRUSH
314	4. TROUBLESHOOTING
314	4.1 General
314	4.2 Checking Input Voltage, Peripheral Conditions, Operation, Programming, Driver, Machine and Interface Control
316	4.3 NC System Check (No tools required)
317	4.4 NC Status
318	APPENDICES
320	APPENDIX 1 TAPE CODES USED IN PROGRAMMING
320	APPENDIX 2 G FUNCTION TABLE
323	APPENDIX 3 TABLE OF COMMAND VALUE RANGES
328	APPENDIX 4 NOMOGRAMS
329	APPENDIX 5 TAPE JOINING
332	APPENDIX 6 PARAMETERS
334	APPENDIX 7 ALARM LIST
337	APPENDIX 8 STATUS WHEN TURNING THE POWER ON, WHEN CLEARED AND WHEN RESET
338	APPENDIX 9 STORED PITCH ERROR COMPENSATION FUNCTION
338	APPENDIX 10 LIST OF OPERATIONS
339	APPENDIX 11 PROGRAM KEY LOCK
340	APPENDIX 12 CUSTOM MACHO INTERRUPT FUNCTION

I. GENERAL

GENERAL

1. INTRODUCTION

The FANUC SYSTEM 6T, a high-accuracy, high-performance fixed-software CNC for lathes, has been developed by FANUC Limited to satisfy the needs of NC markets in the world. The control circuit uses high-speed microprocessors, custom LSIs, semiconductor memory, and the latest storage elements to provide high reliability and significantly improve the performance/cost ratio.

THE FANUC SYSTEM 6T is closed-loop CNC using FANUC DC servo motors developed by FANUC Limited itself, and a high-performance pulse encoder or a resolver as the detector.

This manual explains programming, operation, and routine maintenance. For convenience the basic and optional specifications are explained first. The following detailed descriptions can be referred to as required. This manual covers everything pertaining to the system 6T and, also gives the specifications for the machine operator's panel. Also refer to the manual issued by the machine tool builder.

2. NOTES ON READING THIS MANUAL

- 2.1** The function of an NC machine tool system depends not only on the NC, but on the combination of the machine tool, its magnetic cabinet, the servo system, the NC, the operator's panels, etc. It is too difficult to describe the function, programming, and operation relating to all combinations. This manual generally describes these from the stand-point of the NC. So, for details on a particular NC machine tool, refer to the manual issued by the machine tool builder, which should take precedence over this manual.
- 2.2** This manual addresses as many subjects as possible. But it would become too voluminous to point out everything that should not or cannot be done. Functions which are not specifically stated as possible are impossible.
- 2.3** Notes refer to detailed and specific items. So, when a note is encountered, terms used in it sometimes are not explained. In this case, first skip the note, then return to it after having read over the manual for details.

1. INTRODUCTION

The FANUC SYSTEM 6T is a high-performance fixed-soft-ware CNC for lathe, has been developed by FANUC Limited to satisfy the needs of NC markets in the world. The control circuit uses high-speed microprocessor, custom LSI, semiconductor memory, and the latest storage elements to provide high reliability and significantly improve the performance/cost ratio.

THE FANUC SYSTEM 6T is closed-loop CNC using FANUC DC servo motor developed by FANUC Limited itself, and a high-performance pulse encoder or a resolver as the detector.

This manual explains programming, operation, and routine maintenance. For convenience the basic and optional specifications are explained first. The following detailed descriptions can be referred to as required. This manual covers everything pertaining to the system 6T and, also gives the specifications for the machine operator's panel. Also refer to the manual issued by the machine tool builder.

2. NOTES ON READING THIS MANUAL

2.1 The function of an NC machine tool system depends not only on the NC, but on the combination of the machine tool, its magnetic cabinet, the servo system, the NC, the operator's panel, etc. It is too difficult to describe the function, programming, and operation relating to all combinations. This manual generally describes these from the stand-point of the NC 20, for details on a particular NC machine tool, refer to the manual issued by the machine tool builder, which should take precedence over this manual.

2.2 This manual addresses as many subjects as possible. But it would become too voluminous to point out everything that should not or cannot be done. Functions which are not specifically stated as possible are impossible.

2.3 Notes refer to detailed and specific items. So, when a note is encountered, terms used in it sometimes are not explained. In this case, first skip the note, then return to it after having read over the manual for details.

II. SPECIFICATIONS

II. SPECIFICATIONS

1. BASIC SPECIFICATIONS

No.	Item	Specification	Reference Item											
(1)	Controlled axes	Two axes (X and Z)	III 3.1											
(2)	Simultaneously controllable axes	Two axes for rapid traverse, feed, and manual operation.												
(3)	Increment system	<table border="1"> <tr> <td>Least input increment</td><td>X, Z axis</td><td>0.001 mm</td><td>0.0001 inches</td></tr> <tr> <td rowspan="2">Least command increment</td><td>X axis</td><td>0.0005 mm</td><td>0.00005 inches</td></tr> <tr> <td>Z axis</td><td>0.001 mm</td><td>0.0001 inches</td></tr> </table> X axis diameter programming is employed here. Radius programming is available as an option. The least input increment in metric input can be changed to 0.01 mm by a parameter setting. The least input increment can be changed to 0.0001 mm or 0.00001 inch by another parameter.	Least input increment	X, Z axis	0.001 mm	0.0001 inches	Least command increment	X axis	0.0005 mm	0.00005 inches	Z axis	0.001 mm	0.0001 inches	III 3.2
Least input increment	X, Z axis	0.001 mm	0.0001 inches											
Least command increment	X axis	0.0005 mm	0.00005 inches											
	Z axis	0.001 mm	0.0001 inches											
(4)	Maximum programmable dimensions	± 99999.999 mm ± 9999.9999 inches In inch input and metric output, the max. programmable dimension is ± 3937.0078 inches.	III 2.5											
(5)	Command tape	8-channel opaque paper tape or color tape (Color tape is used for tape readers without reels only.) (EIA RS-227, ISO 1154, JIS C6246)	III 12.1											
(6)	Tape code	EIA RS-244 A	III 12.2											
(7)	Tape format	Variable block, word and address format	III 2.3											
(8)	Decimal point programming	Data can be input with a decimal point. Addresses which can use a decimal point are X, Z, U, W, I, K, R, E and F.	III 2.4											
(9)	Rapid traverse rate	A rapid traverse rate can be set up to 24,000 mm/min., or 960 inches/min. for each axis by parameters. An optional override can be applied to the rapid traverse rate. (Fo, 25, 50 or 100%)	III 4.1											
(10)	Feedrate (G98, G99)	The feedrate can be set per revolution or per minute with G codes. G98: Feed per minute (mm/min., inches/min.) Range: 1~15,000mm/min. 0.01~600.00inches/min. G99: Feed per revolution (mm/rev, inches/rev) Range: 0.01 ~ 500.00 mm/rev 0.0001 ~ 50.0000 inches/rev An override from 0 to 200% in increments of 10% can be used.	III 4.2											
(11)	Automatic acceleration/deceleration	In rapid traverse, regardless of whether the control is in manual or automatic mode, the acceleration/deceleration ramps are linear therefore shortening the positioning time.	III 4.5											

No.	Item	Specification	Reference Item
(12)	Absolute/incremental programming	Address words X and Z specify absolute dimensions. U and W specify incremental dimensions. These address words may be specified in the same block.	III 3.7
(13)	Programming of absolute zero point (G50)	A G50 command with address X and Z establishes the coordinate system so that the coordinate values of the present tool position become the specified value.	III 5.10
(14)	Positioning (G00)	Execution of a G00 command moves the machine at the rapid traverse rate. The machine decelerates to a stop at the end point, and an inposition check (to check whether the machine has reached the specified position) is performed according to parameter setting.	III 5.1
(15)	Linear interpolation (G01)	Linear interpolation is performed at the feedrate indicated by the F code.	III 5.2
(16)	Circular interpolation (G02, G03)	Any arc between 0 and 360 degrees may be programmed with G02 or G03. G02: Clockwise (CW) G03: Counterclockwise (CCW)	III 5.5
(17)	Thread cutting (G32)	Straight thread, tapered screw, or scroll thread can be cut by specifying the lead directly with the F code.	III 5.7
(18)	Buffer storage	One block of tape information is read into buffer storage while the previous block is being performed. This is to avoid interruptions between blocks caused by the reading time of the tape reader. While the data is in buffer storage, BUF is displayed on the CRT.	
(19)	Dwell (G04)	Execution of a G04 command prevents control from proceeding to the next block of information for the period of time specified in one of following addresses: P, U, or X.	III 5.9
(20)	Miscellaneous function (M 2-digit)	Control (ON/OFF) of machine tool functions can be performed by specifying a 2-digit numerical command following the address "M". Only one M code can be specified in a block.	III 8.4
(21)	Spindle speed function (S 2-digit)	Spindle speed can be specified with a 2-digit numerical command following the address "S".	III 8.1
(22)	Tool function (T2-digit/T4-digit)	Tool selection and tool offset are effected by specifying either a 2-digit or a 4-digit numerical command following the address "T".	III 8.3
(23)	Tool offset	The least significant digit (in a 2-digit T code) or the two least significant digits (in a 4-digit T code) in the T code indicate the tool offset number. The offset value corresponding to the tool offset number must be input from the MDI keyboard and stored in memory. 16 offset pairs are standard, additional offset pairs are available as options for a total of 32 or 64 offset pairs.	III 6.1

No.	Item	Specification	Reference Item
(24)	Dry run	The feed rate becomes the Jog feed rate in dry run mode. The rapid traverse rate remains effective for the rapid traverse command (G00) and the optional rapid traverse override is also effective. It is also possible to execute a dry run for the rapid traverse command (G00) by parameter setting.	IV 4.6.9
(25)	Interlock	The feeds for both axes can be stopped at the same time. If the interlock is set while the machine tool is moving along the axes, it decelerates to a stop. If the interlock signal is cleared, the machine tool accelerates to start feeding.	
(26)	Single block	Commands may be executed block by block.	IV 4.6.3
(27)	Optional block skip	When the optional block skip switch on the machine tool operator panel is ON, blocks with a slash code (/) at the beginning of the block are ignored.	III 2.8
(28)	Manual absolute ON/OFF	The tool travel distance during manual operation can be added to the absolute coordinate values by adjusting the manual absolute switch on the machine tool as follows. Manual absolute switch ON: added OFF: not added	IV 4.4.4
(29)	Auxiliary function lock	This function stops transmission of BCD and strobe signals of the M, S and T functions to the machine tool.	
(30)	Machine lock	When in the machine lock status, the machine does not move but the position display is updated as if the machine were moving. The machine lock status can be entered even in the middle of a block.	IV 4.6.10
(31)	Feed hold	The feeds for all axes can temporarily be stopped. Push the cycle start button to resume automatic feeding. Manual operation is allowed in the manual mode before resuming the automatic feed.	IV 4.6.2
(32)	Override cancel	A signal from the machine tool can be used to hold the feedrate override at 100%.	
(33)	Emergency stop	When the emergency stop button is pushed, all commands are suspended and the machine stops in a moment.	IV 4.2
(34)	Remote reset	The NC can be reset from outside. All commands are suspended and the machine tool decelerates to a stop.	
(35)	Remote power supply ON/OFF	It is possible to turn the power supply on and off with a contact signal from the machine tool (such as from the operator panel) as well as by using the power supply ON/OFF buttons on the front panel of the NC.	
(36)	Overtravel	When a signal indicating that one of the machine tool's movable parts has reached the stroke end is received, the machine decelerates to a stop and the overtravel indication is displayed.	

No.	Item	Specification	Reference Item
(37)	Manual continuous feed	(1) Jog feed The jog feed rate can be changed in 24 steps with a rotary switch. The ratio of the 24 steps is a geometric series. (2) Manual rapid traverse Rapid traverse can also be effected by manual operation. An override can be applied to the rapid traverse rate set by parameters. However, this rapid traverse override is optional.	IV 4.4.1
(38)	Incremental feed	Positioning in incremental amounts can be performed as follows, permitting efficient manual positioning. Incremental amounts: 0.001, 0.01, 0.1, 1, 10, 100 mm (Metric input) 0.0001, 0.001, 0.01, 0.1, 1, 10 inches (Inch input)	IV 4.4.3
(39)	Origin button	This creates a work coordinate system so that the position of the tool on the axis selected from the MDI keyboard becomes the zero point of that axis.	
(40)	Sequence number search	A sequence number in a program can be searched for by using the MDI & CRT unit.	IV 5.24
(41)	Program number search	A 4-digit program number following address O can be searched for by using the MDI & CRT unit. In ISO code, a colon (:) can be used instead of O.	IV 5.14
(42)	Backlash compensation	This function compensates for lost motion inherent in a machine system. The compensation amount can be set by parameters in least command increment in the range of from 0 to 255 pulses.	
(43)	NC ready signal	This signal is sent to the machine tool when the NC power supply is turned on and control is ready. The signal stops when the power is turned off.	
(44)	Servo ready signal	This signal is sent to the machine tool when the servo system is ready. Apply a brake to an axis when this signal has not been issued.	
(45)	Rewind signal	The NC outputs this signal while the tape reader is rewinding.	
(46)	NC alarm signal	The NC outputs this signal when the alarm status is entered.	
(47)	Distribution end signal	The NC outputs this signal when the execution of a move command has ended. If an M, S or T function and a move command are specified in the same block, this signal can be used to execute the M, S or T function after the move command has been completed.	
(48)	Cycle operation signal	The NC outputs this signal during cycle operation.	
(49)	Cycle start lamp signal	The NC outputs this signal when a cycle starts.	
(50)	Feed hold lamp signal	The NC outputs this signal while the feed hold status is effective.	

No.	Item	Specification	Reference Item
(51)	Key lock of programs	This function can inhibit display, setting or edition of programs No. 9000 to 9899 by key locking.	Appendix 11
(52)	Environmental conditions	(1) Ambient temperature During operation: 32 ~ 113°F (0 ~ 45°C) During housing or transportation -4 ~ 140°F (-20 ~ 60°C) (2) Temperature fluctuations Max. 2.0°F/min. (Max. 1.1°C/min.) (3) Humidity Usually: 75% or less (Relative humidity) Short period of time: Max. 95% (4) Vibration During operation: 0.5G or less (5) Atmosphere When NC equipment is used in an atmosphere with a relatively high density of dust, cutting oil, water, or organic solvents, please contact the NC manufacturer.	
(53)	Weight	Free standing type: Max. 260 kg (with one servo transformer) Max. 300 kg (with two servo transformers) Built-in type 1: Max. 120 kg Built-in type 2: Max. 210 kg Unbundled type: 55 kg The above values are subject to change due to the use of various options and motors.	
(54)	Self-diagnostic function	Following checkings are done. (a) Abnormality from feedback. (b) Abnormality of position controller. (c) Abnormality of servo system. (d) Overheat (e) Abnormality of CPU. (f) Abnormality of ROM. (g) Abnormality of RAM. (h) Data transmission with MDI & DPL. (i) Abnormality of part program memory. (j) Abnormality of tape reader reading. (k) Abnormality of data transmission with PC.	

2. BASIC OPTIONS

No.	Item	Specification	Reference Item														
(1)	Manual data input & CRT character display	There are two types of MDI & CRT units. (1) Built-in NC (2) Built-in Machine	IV 5														
(2)	Connection unit	The connection unit is used to mount an I/O interface circuit for the machine tool. This unit can be mounted either inside the NC or on the machine tool.															
(3)	Part program storage & editing	NC tape contents may be stored and edited using this function. The following six kinds of part program storage & editing are provided depending on memory capacity. <table><tr><th>Name</th><th>Memory capacity by tape length</th></tr><tr><td>Part program storage and editing A</td><td>20 m * **</td></tr><tr><td>Part program storage and editing B</td><td>40 m</td></tr><tr><td>Part program storage and editing C</td><td>80 m</td></tr><tr><td>Part program storage and editing D</td><td>320 m</td></tr><tr><td>Part program storage and editing E</td><td>640 m</td></tr><tr><td>Part program storage and editing F</td><td>1280 m</td></tr></table> * The capacity is reduced by 1 m when user macro option is equipped. ** The capacity becomes 15 m when stored pitch error compensation function is equipped.	Name	Memory capacity by tape length	Part program storage and editing A	20 m * **	Part program storage and editing B	40 m	Part program storage and editing C	80 m	Part program storage and editing D	320 m	Part program storage and editing E	640 m	Part program storage and editing F	1280 m	IV 5.30
Name	Memory capacity by tape length																
Part program storage and editing A	20 m * **																
Part program storage and editing B	40 m																
Part program storage and editing C	80 m																
Part program storage and editing D	320 m																
Part program storage and editing E	640 m																
Part program storage and editing F	1280 m																
(4)	Tape reader	A tape reader is used to input the NC tape data to NC equipment. (1) Tape reader without reels (a) Reading speed: 300 ch/sec (60 Hz) 250 ch/sec (50 Hz) (b) Reading method: Photoelectric (Light emitting diode) (c) Tape box capacity: 30 m (Free standing type) 10 m (Built-in type 1 and 2) (2) Tape reader with reels (For free standing cabinet only) (a) Reading speed: 300 ch/sec (60 Hz) 250 ch/sec (50 Hz)	IV 3														

No.	Item	Specification	Reference Item
(4)	Tape reader (continued)	(b) Rewind speed: 600 ch/sec (60 Hz) 500 ch/sec (50 Hz) (c) Reading method: Photoelectric (Light emitting diode) (d) Reel capacity: Reel diameter 187 mm (dia.) Capacity of a tape 150 m long (0.108 mm thick) (e) Tape box capacity: 30 m (f) Tape rewind The tape is automatically rewound (upon receiving a signal from the machine tool) until % (ISO code) or ER (EIA code) is detected.	IV 3
(5)	Input unit	Four kinds of input units are available depending on the type (use in Japan or other countries), input circuit used for the servo and the capacity of the input fuse for the servo.	
(6)	Position coder	A position coder is connected to the spindle and generates square wave voltage signals with a frequency proportional to the spindle speed to synchronize the feed with the spindle revolution. (1) Position coder A Max. revolution per min: 4000 rpm (2) Position coder B Max. revolution per min: 6000 rpm	

3. ADDITIONAL OPTION

No.	Item	Specification	Reference Item
(1)	Canned cycle (G90, G92, G94)	The following three kinds of canned cycles are provided. (1) Cutting cycle A (G90): For outer cutting (2) Thread cutting cycle (G92) (3) Cutting cycle B (G94): For facing	III 7.1
(2)	Multiple repetitive cycle A (G71, G72, G70, G76)	Multiple repetitive cycle A is composed of four canned cycles. (1) Stock removal in turning (G71) (2) Stock removal in facing (G72) (3) Finishing cycle (G70) (4) Thread cutting cycle (G76)	III 7.2
(3)	Multiple repetitive cycle B (G70 ~ G76)	In addition to the four cycles in multiple repetitive cycle A, multiple repetitive cycle B includes the following cycles. (1) Pattern repeating (G73) (2) Peck drilling on Z axis (G74) (3) Grooving on X axis (G75)	III 7.2
(4)	Multiple repetitive cycle C	Multiple repetitive cycle C includes stock removal in turning and in facing with several pockets (Type II) in addition to the cycles of multiple repetitive cycle B.	III 7.2
(5)	Reference point return A	Reference point return A includes the following functions. (1) Manual reference point return (2) Reference point return check (G27) (3) Automatic reference point return (G28) (4) Return from reference point (G29)	III 5.8
(6)	Reference point return B	In addition to the functions of reference point return A, this function will move the machine to the 2nd reference point (G30).	III 5.8
(7)	Stored stroke limit 1	Stored stroke limit 1 stores in parameters the maximum/minimum travel along any axis beyond which the machine is forbidden to move.	III 5.14
(8)	Stored stroke limit 2, 3 (G22, G23)	(1) Stored stroke limit 2 The forbidden area is the region inside the area specified by parameters. (2) Stored stroke limit 3 The forbidden area is the region inside or outside the area specified by parameters or a program. Stored stroke limit 2 and 3 can be made effective or ineffective with G codes. G22: effective G23: ineffective	III 5.14
(9)	Stored pitch error compensation	This is the function to compensate for the pitch error of the feed screw.	Appendix. 9

No.	Item	Specification	Reference Item
(9)	Stored pitch error compensation (continued)	The function can improve accuracy of machining and increase mechanical life. The pitch error is compensated for by storing within the NC unit the position where the compensation is made and the compensation value at the position without placing a dog on the machine.	Appendix. 9
(10)	Automatic coordinate system setting	When reference point return is performed manually, a work coordinate system is established.	III 5.10.3
(11)	Work coordinate system shift	When the coordinate system used for programming is not the actual coordinate system, the coordinate system can be shifted.	III 5.10.4
(12)	Constant surface speed control A, B (G96, G97)	This function changes the rotational speed of the spindle motor so that the surface speed remains constant at any tool position. The constant surface speed is directly specified in a 4-digit S command. Whether or not the constant surface speed control is effective is specified with the following G codes. G96: effective G97: ineffective The calculated spindle speed is sent to the machine tool in a 12 bit binary signal (in constant surface speed control A) or an analog signal proportional to the spindle speed (in constant surface speed control B).	III 8.2
(13)	Thread cutting by 6-digit E code	Instead of an F code, an E code can be used to specify the precision thread lead. The following thread lead can be programmed: Metric input: 0.0001 ~ 500.0000 mm Inch input: 0.000001 ~ 9.99999 inch	III 5.7
(14)	Variable lead thread cutting	Thread cutting of variable lead is performed by specifying the increased value or decreased value of the lead per one turn of a screw.	III 5.7.2
(15)	Temporary stop of thread cutting	When the feed hold status is entered during a thread cutting cycle of G92 or G76, the tool immediately draws back and returns to the start point of the cycle.	
(16)	Continuous threading	It is possible to designate the threading blocks (G32, G34) continuously. The dead time in the pulse distribution between the threading blocks is eliminated. Therefore, threading can be conducted correctly without damaging the threads even when the same point is threaded repeatedly by changing the depth of cut.	
(17)	Rapid traverse override	An override is applied to the rapid traverse rate. (Fo, 25, 50 and 100%) "Fo" is set by parameters in the following range.	IV 4.6.13

No.	Item	Specification	Reference Item
(17)	Rapid traverse override	Metric output: 6 ~ 15,000 mm/min. Inch output: 0.6 ~ 600.0 inch/min.	IV 4.6.13
(18)	Manual Handle Interruption function	When the tool is to be slightly fed in or relieved during automatic operation, this can be done without interrupting the machining by turning the manual pulse generator. The tool will shift by the distance corresponding to the number of pulses generated by the pulse generator, which are superposed on the command pulses for automatic operation. Thereafter, the work coordinate system is shifted by the distance corresponding to the pulses.	IV 4.7
(19)	Run hour display	Data can be displayed every 2 seconds, min., hr or 0.1 hr and it can be preset.	IV 5.11
(20)	Automatic acceleration/deceleration for feed	By setting a parameter, it is possible to apply acceleration/deceleration of the exponential function type with a time constant from 8 msec. up to 4000 msec. to feeding or jogging.	
(21)	Special G code C	Refer to the G code table in appendix 2.	Appendix. 2
(22)	Incremental offset input	A compensation value to be added to the previously set tool offset value can be input. This option makes the correction of offset values easy.	IV 5.12.2
(23)	Tape input of offset data	A tape punched with the following command can be input for every tool No. from a tape reader: G10P__X__Z__R__;	IV 5.27
(24)	Automatic tool compensation	By moving the tool to the measured point, NC corrects the currently set compensation value of the tool.	III 6.7
(25)	Direct input of offset value measured	The tool offset value is set by measuring the workpiece diameter and length after cutting it manually and inputting the value via MDI.	III 6.6
(26)	Offset value counter input	After letting a standard tool run against the machine fixed point by manual operation, let the tool to be measured run against the same machine fixed point. When U INPUT and W INPUT are pushed, X and Z-axis offset values of the tool are set, respectively.	IV 5.12.7
(27)	Tool life management	The tools which can be set on the tool turret are divided into several groups, the tool life (operation time or frequency) is set for each group. A tool is selected by designating the group No., and the operation time or frequency is counted each time the tool is used. When the set life is reached, the next tool of the same group is automatically selected. When life of all tools of a group runs out, a signal is output to inform that the operator should change the tools.	III 11

No.	Item	Specification	Reference Item
(28)	Tool nose radius compensation (G40 ~ G42)	Tool nose radius compensation can be made by specifying a G code (G41 or G42). The tool nose radius of each tool is stored in memory as is the tool offset value. G40: Compensation cancellation G41: Left compensation in the direction of movement G42: Right compensation in the direction of movement The max. compensation value is ± 999.999 mm or ± 99.9999 inches.	III 6.3
(29)	Tool geometry offset and tool wear offset	The tool offset is divided to the offset for compensating the tool shape or the tool mounting position (tool geometry offset), and the offset for compensating the tool nose wear (tool wear offset). These can be set individually. This feature greatly facilitates the machining. Namely, the operator may only set and change the wear amount (small value) insofar as well controlled tools are used.	III 6.2
(30)	Additional offset pairs A (16 pairs)	In a 4-digit T code command, 16 offset pairs for tool offset and tool nose radius compensation can be added to the standard 16 pairs, resulting in 32 pairs.	
(31)	Additional offset pairs B	The tool offset and the tool nose radius compensation pairs are increased from max. 32 pairs to max. 64 pairs.	
(32)	External tool compensation A	This changes the offset value externally.	III 6.8
(33)	External work No. search A	A work No. is selected using a switch on the machine tool, and the corresponding program is found and executed.	IV 4.6.14
(34)	External data input	This sends data to the NC from an external unit such as the machine tool to perform the specified operation. Depending on the type of external data, it includes following. (1) External work No. search C (2) External tool compensation C (3) External alarm message (4) External operator message	232
(35)	Radius programming on X axis	A move along the X axis can be input as a radius value. When this option is used, a move along the X axis cannot be input as a diameter value.	III 3.8
(36)	X axis mirror image	The sign in tape or MDI commands for the X axis can be reversed. This can be set with a switch on the machine tool control panel.	IV 4.6.12
(37)	Mirror image for double turrets	It is possible to reserve the sign of X axis values in the program by changing the G code to perform symmetrical cutting (mirror image).	III 5.10.6

No.	Item	Specification	Reference Item
(38)	Inch/metric switching (G20, G21)	G codes can be used to select the inch or metric system for input. G20: Input in inch G21: Input in metric	III 5.11
(39)	Chamfering, corner R	In straight cutting, perpendicular/parallel to the axis, chamfering and corner arcs can be inserted at the corners. Addresses I, K and R specify data for the chamfering and corner arc.	III 5.3
(40)	Part program collation	This collates memory and tape contents, and gives an alarm if a discrepancy is detected.	IV 5.18
(41)	Circular interpolation by radius designation	In circular interpolation, the radius of an arc can be specified directly, instead of using addresses I and K. An arc specified with radius designation must be 180° or less.	III 5.5
(42)	Program restart	Specify the sequence No. and restart the program from there.	IV 5.25
(43)	Addition of registerable programs	96 programs are added to the standard programs for a total of 191 programs. (Any of part program storage & editing C to F is required.)	
(44)	Feedrate sine curve control	In circular arc cutting command, it is possible to change the feedrate of an axis in the form of a sine curve by conducting interpolation for the other axis in the arc plane (setting it as the imaginary axis) without moving it.	III 5.6
(45)	Skip function	When a signal is issued during operation of linear interpolation from an external source, it is possible to cancel the remaining move and proceed to the next block.	IV 5.15
(46)	Additional option block skip	Specify a numerical value from 1 to 9 following a / at the start of a block. Also, provide 9 optional block skip switches on the machine tool. When optional block switch n is ON, the block with /n is not executed but skipped.	III 2.8
(47)	Custom macro A, B	A specific function may be created by the machine tool builder or the user. There are custom macro A and B depending on the used functions.	III 10
(48)	Menu switch	Functions of the switches existing on the machine operator's panel can be turned on and off from the MDI & CRT. As a result, it is possible to reduce the switches on the machine operator's panel.	IV 5.32
(49)	Direct drawing dimension programming	The angle of straight line, chamfering value, corner radius value, etc. indicated on the machining drawing can be used directly in programming. Further, it is possible to insert chamfering and corner rounding between straight lines at an arbitrary angle.	IV 5.4

No.	Item	Specification	Reference Item
(50)	Sequence No. comparison stop	When a block of the sequence No. set in advance appears during the execution of a program, the machine stops after executing that block. The sequence No. can be set from the MDI & CRT by the operator. With this function, it is possible to stop the operation at an arbitrary point without changing the program.	IV 5.26
(51)	Graphic display	The programmed locus of the tool being used can be drawn on the 9-inch CRT screen. Advance in machining can be checked by watching the locus displayed on the CRT screen.	Mentioned in the next edition
(52)	Input power transformer	Any of the following input power supplies can be used. AC 200/220/230/240/380/415/440/460/480/550 V (The transformer for the servo is not included.)	
(53)	Input/output interface	Programs, tool offset amount, parameters, etc., stored in memory can be punched out. Selection of EIA or ISO code is made from the MDI panel. The punched paper tape isTV checked.	
(54)	FANUC CASSETTE B1/B2 (Bubble cassette)	It is possible to input/output data from a bubble cassette with built-in bubble memory. (1) External dimensions: 120 x 64 x 22 (mm) (2) Weight: 235 g (3) Memory capacity: Converted to tape length 80 m/160 m	IV 7
(55)	FANUC CASSETTE P1 (LSI cassette)	It is possible to input or output data from a LSI cassette built-in LSI memory. (1) External dimensions 120 x 62 x 24 (mm) (2) Weight 150 g (3) Memory capacity Converted to tape length 20 m	IV 7
(56)	Portable tape reader	A tape reader without reels with RS 232C interface is available. It is portable type and can be used for plural NCs by connecting to each NC.	
(57)	ISO code input	Both ISO (ISO 840) and EIA (EIA RS-244-A) codes can be used for the tape code. Distinction between ISO and EIA codes can be made automatically.	
(58)	Door interlock	This cuts off the power supply when the door of the NC is opened.	
(59)	Remote type position display	A position display provided independently for each axis can be mounted on the machine tool.	IV 6

No.	Item	Specification	Reference Item																						
(60)	Manual pulse generator	A manual pulse generator can be used for fine-adjustment feeding for a machine tool. The manual pulse generator generates 100 pulses per rotation. The distance moved per pulse can be switched to the three levels shown below. 0.001, 0.01, 0.1 mm: For metric input 0.0001, 0.001, 0.01 inch: For inch input	IV 4.4.2																						
(61)	PC	It is possible to build a programmable controller in the NC. FANUC PC-MODEL A and B are provided for this. <table border="1"> <tr> <th colspan="2"></th><th>FANUC PC MODEL A</th><th>FANUC PC MODEL B</th></tr> <tr> <td rowspan="2">Number of input points</td><td>Contact (Max.)</td><td>176 points</td><td>176 points</td></tr> <tr> <td>Voltage (Max.)</td><td>16</td><td>16</td></tr> <tr> <td rowspan="2">Number of output points</td><td>Contact (Max.)</td><td>104</td><td>104</td></tr> <tr> <td>Noncontact (Max.)</td><td>24</td><td>24</td></tr> <tr> <td colspan="2">Number of program steps (Max.)</td><td>2000 steps</td><td>5000 steps</td></tr> </table>			FANUC PC MODEL A	FANUC PC MODEL B	Number of input points	Contact (Max.)	176 points	176 points	Voltage (Max.)	16	16	Number of output points	Contact (Max.)	104	104	Noncontact (Max.)	24	24	Number of program steps (Max.)		2000 steps	5000 steps	
		FANUC PC MODEL A	FANUC PC MODEL B																						
Number of input points	Contact (Max.)	176 points	176 points																						
	Voltage (Max.)	16	16																						
Number of output points	Contact (Max.)	104	104																						
	Noncontact (Max.)	24	24																						
Number of program steps (Max.)		2000 steps	5000 steps																						

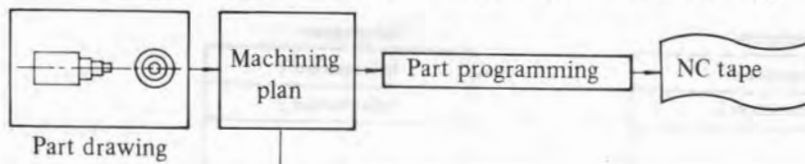
III. PROGRAMMING

III. PROGRAMMING

1. WHAT IS PROGRAMMING

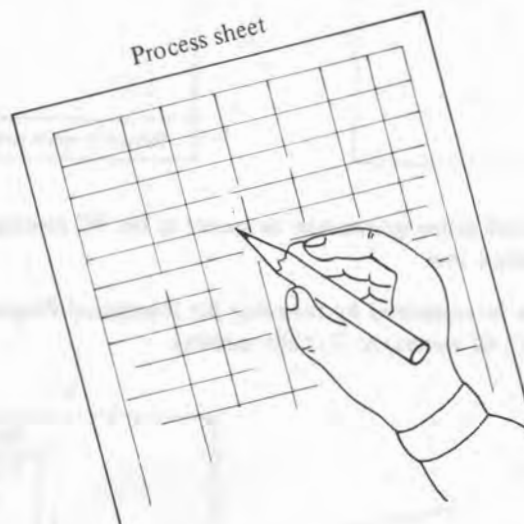
The NC machine tool moves as directed by a program punched on an NC tape. When machining a part using an NC machine tool, the tool path and other machining conditions are programmed. This program is called a part program.

The figure below shows NC tape creation beginning from a part drawing.



- 1) Determination of machining range by NC and selection of NC machine tools used.
- 2) Method of loading workpieces on the machine tool and selection of required jigs and tools.
- 3) Machining sequence (classification of process, tool start point, cutting depth, tool path in rough machining and finishing).
- 4) Selection of cutting tools and tool holders and determination of mounting location on the machine for cutting tools and tool holders.
- 5) Cutting conditions (spindle speed, feedrate, coolant, etc.)

The part program tells the NC the tool path and auxiliary motion of the machine tool. This is written on the process sheet.

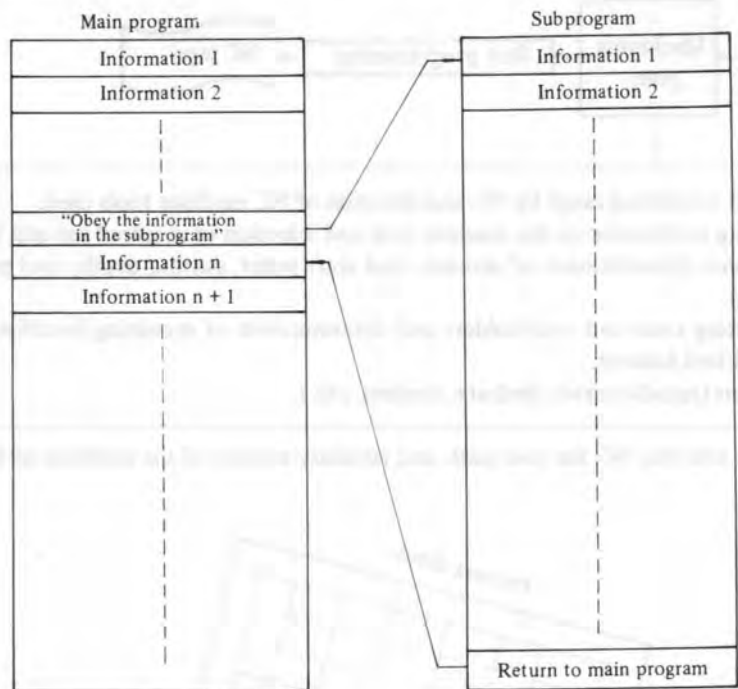


The program on this process sheet is punched on paper tape using a paper tape punch. This paper tape is called an NC tape and is input to the NC.

This chapter explains how to write a part programs.

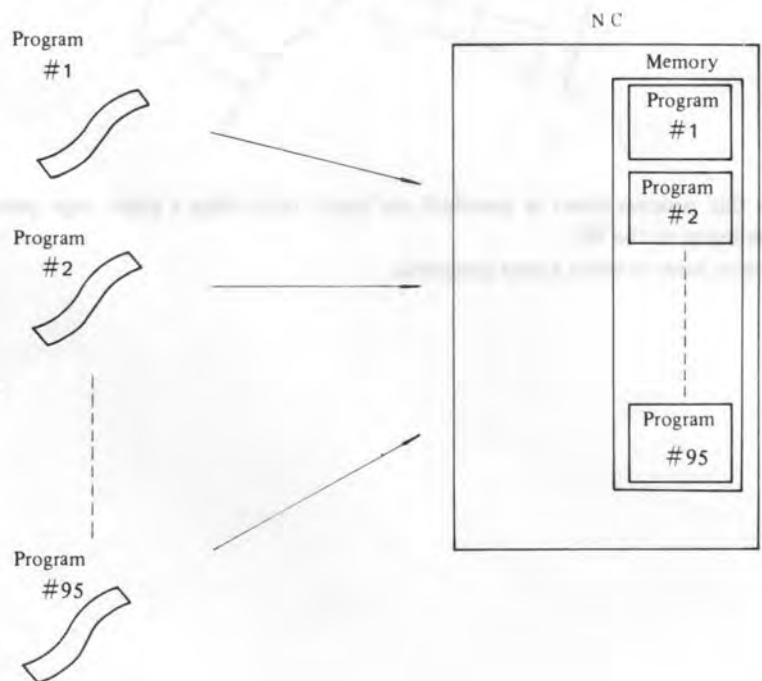
2. PROGRAM COMPOSITION

A program is divided into a main program and subprogram: Normally, the NC operates according to the main program, but when a command calling a subprogram is encountered in the main program, control is passed to the subprogram. When a command indicating to return to the main program is encountered in the subprogram, control is returned to the main program.



A total of 95 main programs and subprograms may be stored in the NC memory, and the NC uses one of these main programs to move the NC machine tool.

(Note 1) Up to 191 programs can be registered by selecting the Registered Programs Addition (option) and Part Program Storage & Edit C (80 meters) to F (1280 meters).



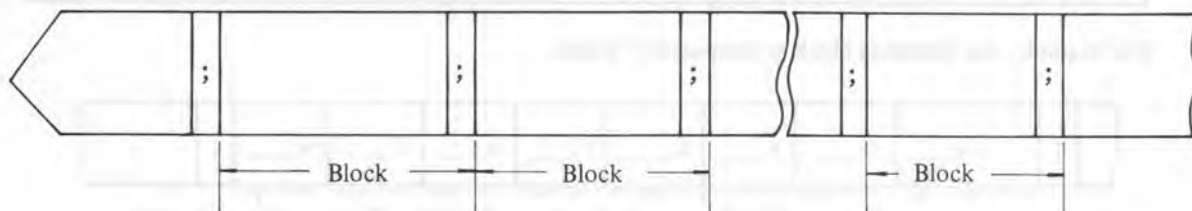
Refer to the section on OPERATION for how to store and select one program.

2.1 Block

A program is composed of several commands.

One command is called a block.

One block is separated from another block with an end of block code. This manual uses ; as the end of block code.

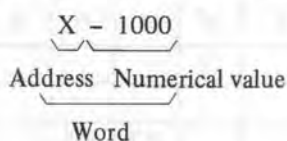


(Note 1) There is no limit to the number of characters in a block.

(Note 2) The end of block code is CR in EIA code and LF in ISO code.

2.2 Word

A block is composed of one or more words. A word is composed of an address followed by numbers as is shown below. (An algebraic sign (+ or -) may be added before the numerical value.)



The address is a letter which indicates the meaning of the numerical value following the address.

Addresses and their meanings in SYSTEM 6T are shown below.

Some addresses may vary in meaning depending on preparatory functions specified in the program.

Function	Address	Meaning
Program number	: (ISO)/O (EIA)	Program number
Sequence number	N	Sequence number
Preparatory function	G	Motion mode (Linear, arc, etc.)
Dimension word	X, Z, U, W	Coordinate axis motion command
	R	Arc radius, corner R
	I, K	Coordinate values of arc center, chamfering
Feed function	F, E	Feedrate, thread lead
Spindle speed function	S	Spindle speed
Tool function	T	Tool number, tool offset number
Miscellaneous function	M	ON/OFF control on the machine tool
Dwell	P, U, X	Dwell time
Program number designation	P	Designation of the subprogram number

2.3 Input Format

Each word in a block must be specified in a format as explained below. The input format used by the SYSTEM 6T is called a variable block format since the number of words in a block and the number of characters in a word can be changed. This format is very convenient for programming.

(1) Metric input

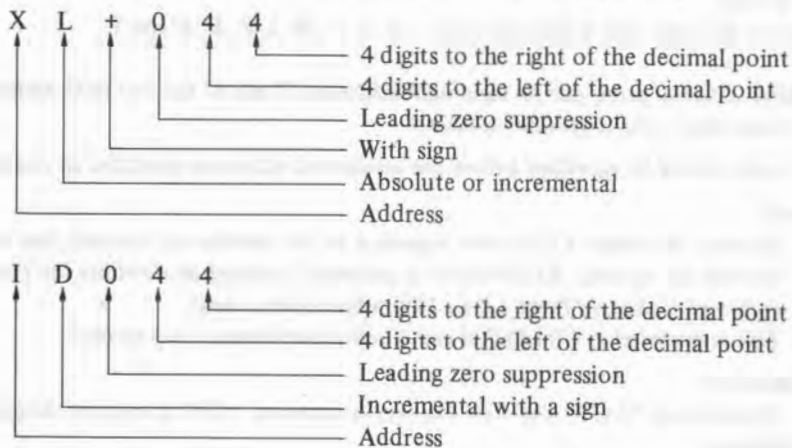
$$N04 \cdot G02 \cdot \alpha L + 053 \cdot \beta L + 053 \cdot \left\{ \begin{matrix} RD053 \\ ID053 \text{ KD} + 053 \end{matrix} \right\} \cdot \left\{ \begin{matrix} F032 \\ F050 \end{matrix} \right\} \cdot E034 \cdot \left\{ \begin{matrix} S02 \\ S04 \end{matrix} \right\} \cdot \left\{ \begin{matrix} T03 \\ T04 \end{matrix} \right\} \cdot M02;$$

(2) Inch input

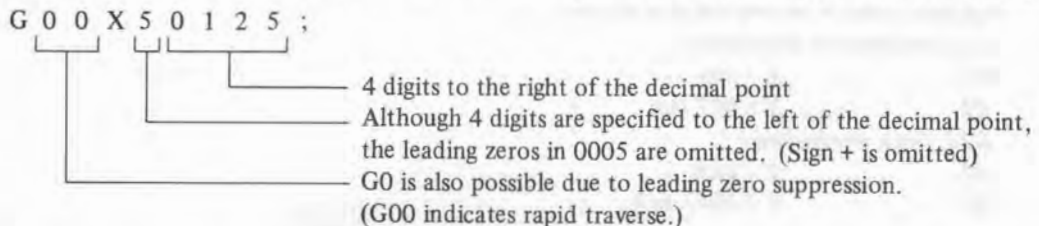
$$N04 \cdot G02 \cdot \alpha L + 044 \cdot \beta L + 044 \cdot \left\{ \begin{matrix} RD044 \\ ID044 \cdot KD044 \end{matrix} \right\} \cdot \left\{ \begin{matrix} F024 \\ F032 \end{matrix} \right\} \cdot E016 \cdot \left\{ \begin{matrix} S02 \\ S04 \end{matrix} \right\} \cdot \left\{ \begin{matrix} T03 \\ T04 \end{matrix} \right\} \cdot M02;$$

(Note 1) α : X or U, β : Z or W

(Note 2) The above addresses and numerical values indicate the followings:



For example, a command to move a tool 5.0125 inches along the X axis at the rapid traverse rate is as follows:



(Note 3) When 2 or more words of the same address are specified in a block, the word specified last is effective as a rule.

No alarm is generated.

(Example)

G01M03S200M08; M08 is effective, while M03 is ignored.

- On G codes, any G codes specified last from each group is effective.
- Either X or U and/or either Z or W which is specified last is effective.
- If both R and I, K are specified for a circular interpolation command, always address R is effective, irrespective of specified sequence.
- If both addresses E and F are specified for thread cutting in the same block, either address specified later is effective.

(Note 4) F032 and F050 in millimeter input can be changed to F033 and F051 respectively by switching parameters.

See Section 4.4.

F032 (millimeter input) and F024 (inch input) are input formats for feed per revolution.

F050 (millimeter input) and F032 (inch input) are input formats for feed per minute.

(Note 5) The above format omits addresses A, P, Q, L and D because they have various meanings.

(Note 6) See Section 2.4 for decimal point programming.

(Note 7) The millimeter input values for the X, Z, U, W, I, K, R and D addresses may be multiplied by ten by setting a parameter.

$$\alpha L + 052 \cdot \beta L + 052 \cdot \begin{cases} RD052 \\ ID052 \cdot KD052 \end{cases} \{ \text{Input in millimeters} \}$$

See Section 3.2.3 (Multiplication of input values by ten)

(Note 8) S04 is optional.

2.4 Decimal Point Programming

SYSTEM 6T can input numerical values with a decimal point. However some addresses cannot use a decimal point. A decimal point may be used with mm, inches or second values. The location of decimal point is mm, inch or second.

Z15.0 Z15 millimeters or Z15 inches.

F10.0 10 mm/rev, 10 mm/min., 10 inch/rev or 10 inch/min.

G04X1. Dwell for one second

The following addresses can be used with a decimal point: X, Z, U, W, I, K, R, E and F

(Note 1) In the dwell command, decimal point can be used with addresses X and U but not with address P.
(This is because P is also used for a sequence number.)

(Note 2) The appropriate G code should be specified before the numerical values are specified in one block.

(1) G20; (Inch dimension)

X1.0G04; Because the value X1.0 is not regarded as the number of seconds, but the distance of motion (in inches), X1000G04 is assumed resulting in dwelling for 10 seconds. Indication also changes from 1.0 to 10.0 when G04 is input.

G04X1.0 This is regarded as G04X1000 and dwell is performed for a second.

(2) G98; (mm/min. designation)

F1.G99; Regarded as F1 G99 and 0.01 mm/rev is assumed. (G99 is mm/rev designation)

G98; (mm/min. designation)

G99F1.; Regarded as G99F100 and 1 mm/rev is assumed. (G99 is mm/rev designation)

(Note 3) There is a great difference in values with and without the decimal point. Programming differs from methods used in electric calculators, etc.

G21; (millimeter dimensions)

X1. X 1 mm

X1 X 0.001 mm

G20; (inch dimensions)

X1. X 1 inch

X1 X 0.0001 inch

(Note 4) Values with and without a decimal point can be specified together.

X1000Z23.7;

X10.Z22359;

(Note 5) Values less than the least input increment are deleted.

When X1.23456 is specified, X1.234 is assumed in millimeter input and X1.2345 is assumed in inch input. In incremental dimensions, errors accumulate.

In absolute dimensions, errors are not accumulated but an error deleted exists in absolute programming. Also, the number of digits must not exceed the maximum number of digits allowed.

X1.23456789 . . This is an error because it exceeds 8 digits.

X1.2345678 . . . This is not an error because it is within 8 digits.

(Note 6) When a number with a decimal point has been input, the number is converted into an integer of the least input increment.

Example

X12.34 → X123400 (input in inches)

This converted integer is checked for its number of digits.

Example

X1234567.8 → X12345678000 (input in inch)

An alarm occurs because the converted integer exceeds 8 digits.

2.5 Maximum Programmable Dimensions

The maximum programmable dimensions of each address are listed in Table 2.5. Note that these figures give the maximum numerical limit, not the mechanical limit of the NC machine tool. Under NC control, the tool may traverse up to 100 m (in millimeter input) along the X axis, but actually travelling distance may be limited to 2 m for the specific machine tool. Similarly, NC control may permit a maximum cutting feed of 15 m/min., but the NC machine tool may be limited to 6 m/min. The manual issued by the machine tool manufacturer should be closely consulted, in addition to this manual when programming to ensure that the actual limitations of the specific machine tool are not exceeded.

**Table 2.5 Basic Addresses and Command Range
(Includes additional options)**

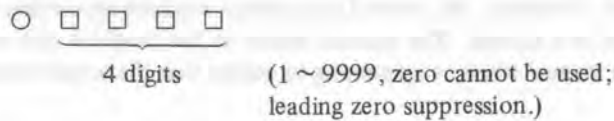
Function	Address	Input in mm Output in mm	Input in inches Output in mm	Input in mm Output in inches	Input in inches Output in inches
Program number	: (ISO) O (EIA)	1 ~ 9999	Same as left	Same as left	Same as left
Sequence number	N	1 ~ 9999	Same as left	Same as left	Same as left
Preparatory function	G	0 ~ 99	Same as left	Same as left	Same as left
Dimension word	X,Z,U, W,R, I,K	±99999.999 mm	±3937.0078 inches	±99999.999 mm	±9999.9999 inches
Feed per revolution. Thread lead	F	0.01 ~ 500.00 mm/rev	0.0001 ~ 50.0000 inches/rev	0.01 ~ 500.00 mm/rev	0.0001 ~ 50.0000 inches/rev
Feed per revolution. Thread lead (Feedrate 1/10) (Parameter setting)	F	0.001 ~ 500.000 mm/rev	Same as above	0.001 ~ 500.000 mm/rev	Same as above
Feed per minute	F	1 ~ 15000 mm/min.	0.01 ~ 600.00 inches/min.	1 ~ 15000 mm/min.	0.01 ~ 600.00 inches/mm
Feed per minute (Feedrate 1/10) (Parameter setting)	F	0.1 ~ 15000.0 mm/min.	Same as above	0.1 ~ 15000.0 mm/min.	Same as above
Thread lead	E	0.0001 ~ 500.0000 mm	0.000001 ~ 9.999999 inches	0.0001 ~ 500.0000 mm	0.000001 ~ 9.999999 inches
Thread lead (Feedrate 1/10) (Parameter setting)	E	0.00001 ~ 99.99999 mm	Same as above	0.00001 ~ 99.99999 mm	Same as above
Spindle speed function	S	0 ~ 9999	Same as left	Same as left	Same as left
Tool function	T	0 ~ 9964	Same as left	Same as left	Same as left
Miscellaneous function	M	0 ~ 99	Same as left	Same as left	Same as left
Dwell	X,U,P	0 ~ 99999.999 second	Same as left	Same as left	Same as left
Designation of sequence number	P,Q	1 ~ 9999	Same as left	Same as left	Same as left
Repetition count	L	1 ~ 9999	Same as left	Same as left	Same as left
Parameter	Angle	A	Discrete value	Same as left	Same as left
	Cutting depth	D,I,K	Same as dimension word	Same as left	Same as left
	Count	D	1 ~ 9999	Same as left	Same as left

(Note 1) Feed per revolution and thread lead are actually determined by the speed converted into feed per minute with relation to the spindle speed. Commands can be input with value in the table above.

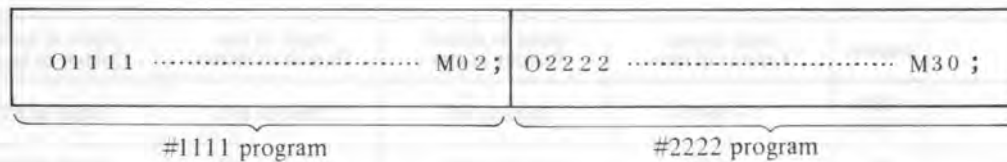
2.6 Program Number

The controller can store programs in its memory. The program number is used to differentiate one program from another.

The program number is identified as follows:

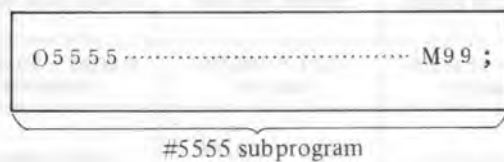


A program begins with a program number and ends with M02;, M30; or M99;.



M02; or M30; means the end of a main program.

M99; means the end of a subprogram.



(Note 1) In ISO code, colon (:) can be used instead of O.

(Note 2) A block with an optional block skip code such as /M02; /M30; or /M99; is not regarded as the end of a program.

(Note 3) When the program number is not specified at the start of the program, the first sequence number (N...) in that program is regarded as the program number. However, N0 is not used as the program number.

(Note 4) When neither program number nor sequence number are specified, a program number must be specified from the MDI keyboard when the program is registered.
(See Section 5.15 in Chapter IV.)

(Note 5) When a tape contains more than one program, an EOB code is unnecessary before the second and subsequent programs. However, when a preceding program ends with ER (EIA code) or % (ISO code), an EOB code is necessary at the head of a program. This is because of the label skip feature.

(Note 6) A programmed tape operation can be performed even when using a programmed tape without a program number. Program numbers are necessary for subprograms.

(Note 7) In some cases, program numbers 9000-9899 are reserved and cannot be used by users.

(Note 8) When the robot option is used, program numbers 9900-9999 are used for robot data.

(Note 9) There is a parameter NEOP to specify that a program end is not assumed by M02, M30 or M99 but ER (EIA), % (ISO) or the next program number (O).

2.7 Sequence Number

A sequence number can be specified (1 - 9999) following address N at the head of a block. The order of sequence numbers is arbitrary and need not be consecutive. Sequence numbers can be specified in all blocks or only in blocks requiring them.

It is recommended that sequence numbers be specified sequentially at important points such as when a tool is changed and a new tool is used.

(Note 1) Sequence number N0 should not be used for compatibility with other NC equipment.

(Note 2) Because program number 0 is invalid, a sequence number regarded as a program number must not be 0.

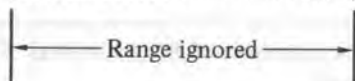
2.8 Optional Block Skip

When a slash followed by a number /n ($n = 1 - 9$) is specified at the beginning of a block and the optional block skip switch n on the machine operator panel is set ON, the block with /n corresponding to switch number n is ignored in tape or memory operation.

When the optional block skip switch n is set OFF, the block with /n is valid. That is, the operator can skip a block with /n at his discretion. 1 in /1 may be omitted.

The following range is ignored when the optional block skip switch is ON.

; /2N123G01X4 ; N7856



Example:

N100X100 ;
N101/2Z100 ;
N102/2/3X200 ;
N103/3Z200 ;

In the above example, the blocks of

N101 and N102 when switch No. 2 is ON

N102 and N103 when switch No. 3 is ON

are skipped.

(Note 1) A slash (/) must be specified at the start of a block. If it is placed elsewhere in the block, the information from the slash (/) to the EOB code is ignored, while information before the slash (/) is effective.

(Note 2) When the optional block skip switch is ON, TH and TV checks are performed for skipped portions, as with the switch is OFF.

(Note 3) The optional block skip is identified when the information is read into the buffer storage from tape or memory. When a block preceded by slash has been read into the buffer, it is not ignored even if the OPTIONAL BLOCK SKIP switch is turned on.

(Note 4) This function is also effective during a sequence number search.

(Note 5) When storing the program in memory, this function is ineffective. A block with a slash (/) is also stored in the memory regardless of the position of the OPTIONAL BLOCK SKIP switch.

(Note 6) When punching out the program from memory, the program is punched out regardless of the position of the OPTIONAL BLOCK SKIP switch.

(Note 7) Some of the optional block skip switches (1 to 9) may not be used on some machine tools. Ask the machine tool manufacturer how many switches that can be used.

(Note 8) If more than one optional block skip code are specified in a block for the system with additional optional block skip function, 1 in /1 cannot be omitted.

Please specify /1 under the above condition.

(Example)

Error

//3 G00X10.0;

Correct

/1/3 G00X10.0;

3. DIMENSION WORD

A dimension word specifies a tool movement and is composed of the address of the axis along which to be moved and the value indicating the move direction and amount. The value varies with absolute or incremental programming. (See Section 3.7.)

Address of dimension word		Meaning
Basic axes	X Z	Specifies a certain target position in a coordinate system (Absolute programming).
	U W	Specifies a move distance (Incremental programming). U is for the X axis and W is for the Z axis.
Parameters in circular interpolation	R	Specifies the radius of an arc.
	I K	Specifies a distance from the start point to the center of an arc along the X axis (I) or along the Z axis (K).

(Example) U-200.00 The tool moves along the X axis in the negative direction for a distance of 200.00.

3.1 Controlled Axis

Axes along which the machine tool moves which can be controlled by the NC are called controlled axes. Each controlled axis is specified by the address of the dimension word used by the NC. There are two controllable axes, X and Z.

The number of controlled axes that can be specified in a block is also two.

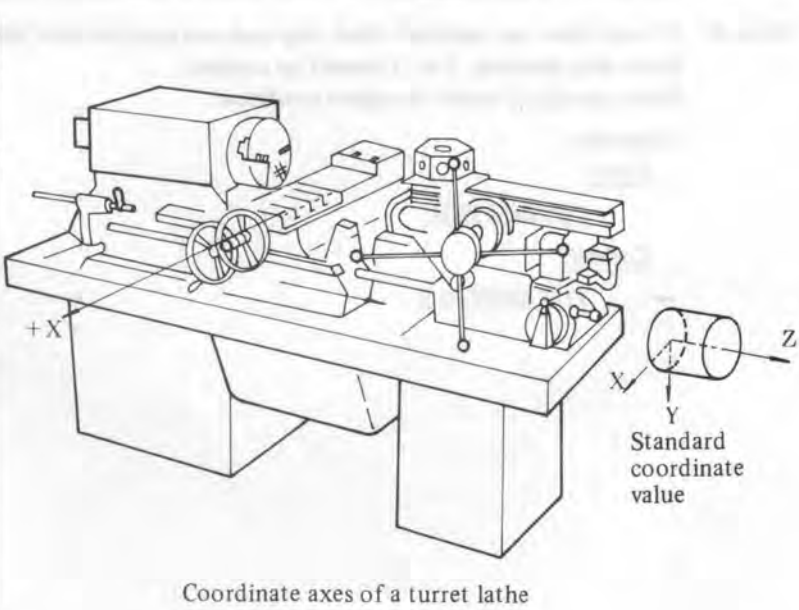
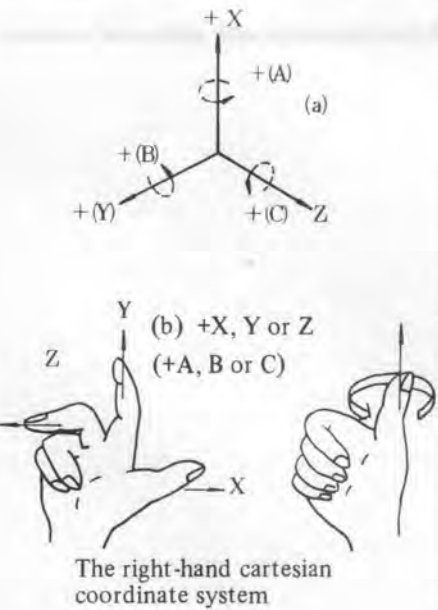
Controlled axes	Simultaneously controllable axes
X, Z	Two

3.1.1 Coordinate axes and symbol of motion

If machine tools have different correlations between the coordinate axes of the machine tool and the symbols of tool motion, severe confusion will result when programming. Basic concepts on this matter are established in EIA RS-267-A and ISO 841.

However, note the following carefully when programming:

- (a) Programs must refer to the standard coordinate system (the right-hand cartesian coordinate system).
- (b) When programming, assume that the workpiece stands still and the tool is moved around it.



3.2 Increment System

The increment system is determined by the following two factors:

3.2.1 Least input increment (Input unit)

The minimum unit of tool travel specified in a command tape. This value is in mm or inches.

3.2.2 Least command increment (Output unit)

The minimum unit of command that the NC can give to the machine tool, in mm or inches. One of the following combinations is used:

	Least input increment	Least command increment
mm input, mm output	0.001 mm	0.001 mm
inch input, mm output	0.0001 inch	0.001 mm
mm input, inch output	0.001 mm	0.0001 inch
inch input, inch output	0.001 inch	0.0001 inch

In diameter programming, the least command increment for the X axis is also the diameter value.

Whether the least command increment is 0.001 mm or 0.0001 inches is determined by the individual machine tool and must be selected by setting a parameter (SCW) in advance.

Selection of a least input increment between 0.001 mm and 0.0001 inches can be made with G codes or set from the MDI.

G20 Least input increment 0.0001 inch

G21 Least input increment 0.001 mm

Either G20 or G21 which has been selected before the power is turned on is effected at the power on.

3.2.3 Multiplication of input unit by ten

The least increment for the metric input can be changed to 0.01 mm by parameter (MIC) setting. The unit of input in inches cannot be changed.

	Address	Input unit	
		mm input	inch input
Dimension word	X, Z, U, W, R, I, K	0.01 mm	0.0001 inch
Dwell	X, P, U	0.01 second	0.001 second
Parameter	D	0.01 mm	0.0001 second

The following do not change either.

- (1) Input other than the above
- (2) Unit of display
- (3) Maximum command range
- (4) Unit of step feed and handle feed
- (5) Input of offset value
- (6) Others

(Note 1) From now on in this manual, the input unit is either 0.0001 inch or 0.001 mm.

(Note 2) The unit of display can be made 0.01 mm by parameter setting (MDI).

3.2.4 Input unit 1/10

The least input increment and the feedrate unit are reduced to one-tenth.

Basic Addresses and Range of Command Value (Includes Additional Option)

Function	Address	mm input mm output	inch input mm output	mm input inch output	inch input inch output
Dimension word	X, Z, I, K, Q, R	±9999.9999 mm	±393.70078 inches	±9999.9999 mm	±999.99999 inches
Feed per minute	F	0.1 ~ 12000.0 mm/min	0.001 ~ 470.000 inches/min	0.1 ~ 12000.0 mm/min	0.001 ~ 470.000 inches/min
Feed per minute (Feedrate 1/10) (Parameter setting)	F	0.01 ~ 12000.00 mm/min	Same as above.	0.01 ~ 12000.00 mm/min	Same as above.
Syn-chronous feed	F	0.001 ~ 99.999 mm/rev	0.00001 ~ 9.99999 inches/rev	0.001 ~ 99.999 mm/rev	0.00001 ~ 9.99999 inches/rev
Syn-chronous feed (Feedrate 1/10) (Parameter setting)	F	0.0001 ~ 99.9999 mm/rev	Same as above.	0.0001 ~ 99.9999 mm/rev	Same as above.
Dwell	X, P	0 ~ 99999.999 sec.	Same as left.	Same as left.	Same as left.

(Note) Max. feedrate may differ from the one shown in this table depending on output unit.

Dimensions of other addresses than the above addresses do not change. The following are also reduced to one-tenth.

- (1) Maximum programmable dimensions.
- (2) Step feed and handle feed units.
- (3) Input offset values.

3.3 Maximum Stroke

The maximum stroke that can be specified is listed in the table below:

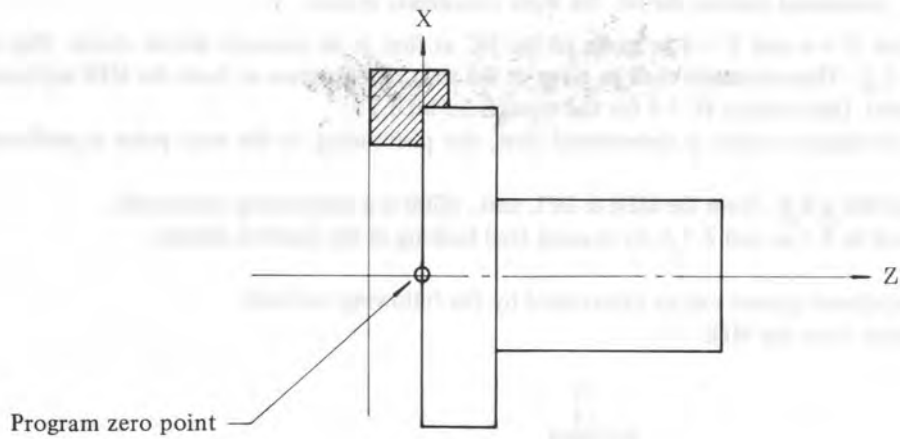
mm input mm output	inch input mm output	mm input inch output	inch input inch output
±99999.999 mm	±3937.0078 inches	±99999.999 mm	±9999.9999 inches

(Note) The above stroke, of course, varies depending on the machine tool.

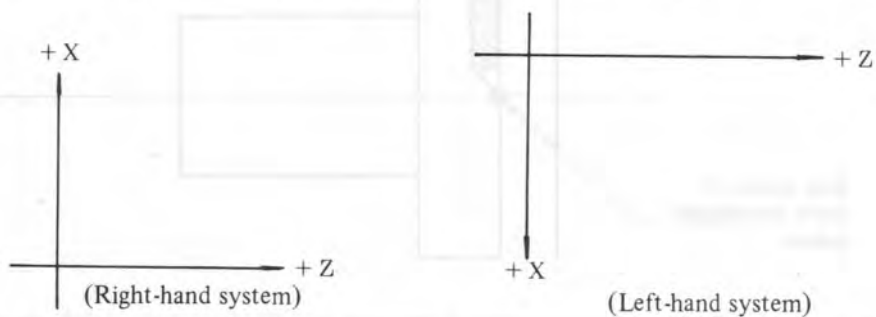
3.4 Program Zero Point and Coordinate System

A program zero point and a coordinate system must be determined when programming. The program zero point is usually placed at an arbitrary position on a workpiece.

For example, the X axis zero point is specified at the workpiece center and the Z axis zero point is specified at the left side surface of a workpiece.

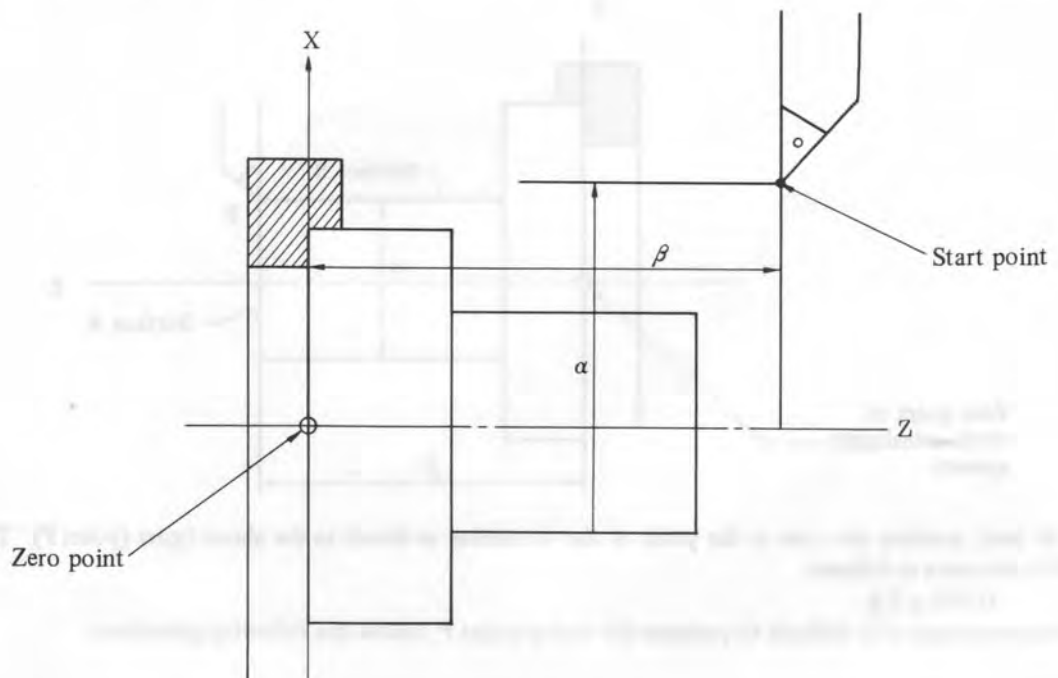


This coordinate system is called the work coordinate system. There are two coordinate systems, right- and left-hand systems. This manual uses the right-hand system.



3.5 Coordinate System and Start Point

The work coordinate system used in programming must be stored in the NC. A tool moves from the start point and the program also starts from the start point. But the coordinate value of a tool at the start point must be given to the NC by using a G50 command.



A $G50X_{\alpha}Z_{\beta}$; command teaches the NC the work coordinate system.

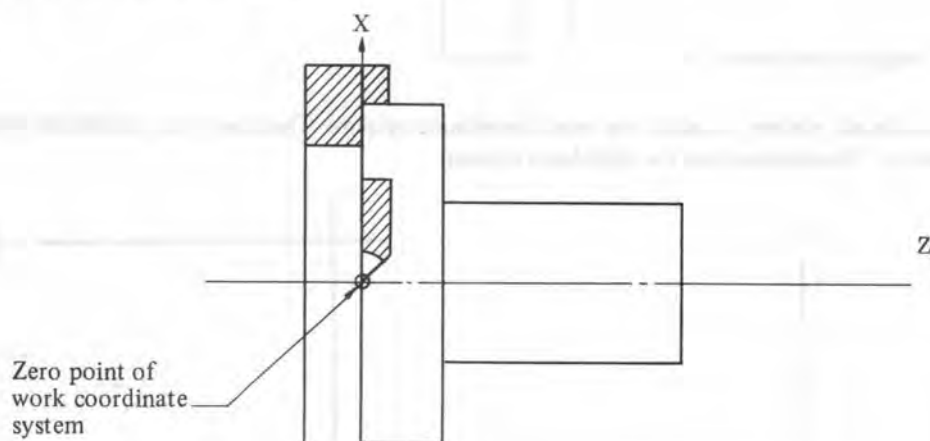
It is enough that $X = \alpha$ and $Z = \beta$ be given to the NC at first in an example shown above. This command is specified as $G50X_{\alpha}Z_{\beta}$; This command shall be given at the start of a program or from the MDI keyboard when the tool is at the start point. (See section IV 5.9 for the input from MDI.)

If the work coordinate system is determined first, the positioning to the start point is performed easily as follows.

- (1) Command $G00X_{\alpha}Z_{\beta}$; from the MDI & DPL unit. ($G00$ is a positioning command).
- (2) Move the tool to $X = \alpha$ and $Z = \beta$ by manual feed looking at the position display.

The work coordinate system can be determined by the following methods.

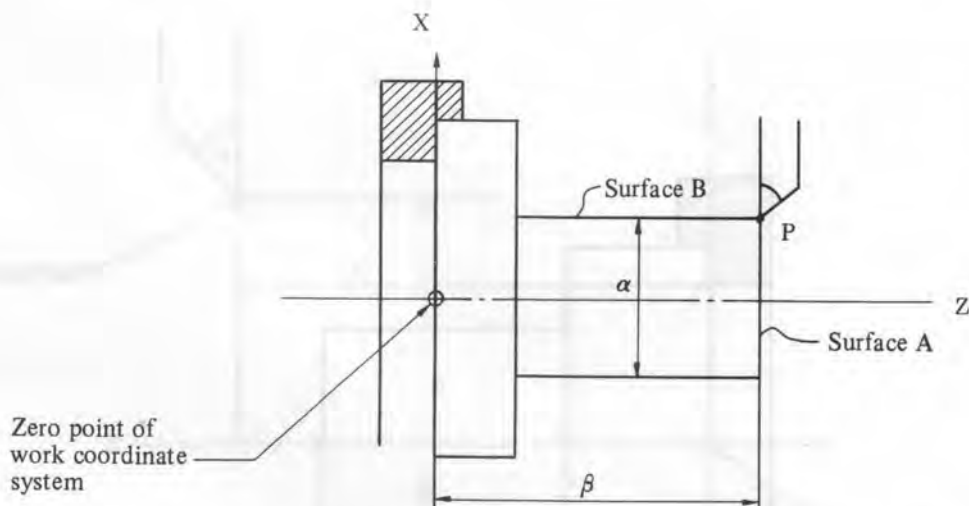
(Method 1) $G50$ input from the MDI



If the tool can be positioned at the zero point of the work coordinate system, specify the following command from the MDI after positioning.

$G50X0Z0$;

However, positioning at the zero point is usually impossible.



At first, position the tool at the point of the workpiece as shown in the above figure (point P). Then specify the $G50$ command as follows:

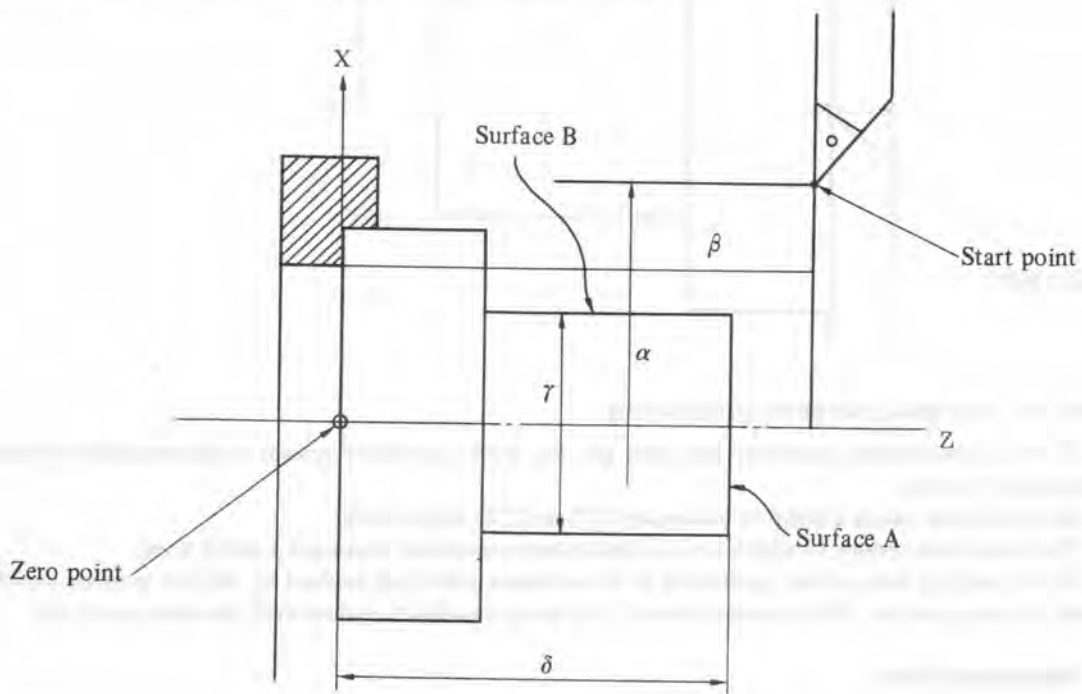
$G50X_{\alpha}Z_{\beta}$;

However since it is difficult to position the tool at point P, follow the following procedure.

- (1) Cut the workpiece along surface A manually.
- (2) Release the tool along the X axis without moving along the Z axis and stop spindle rotation.
- (3) Measure distance " β " and specify G50Z β and push and buttons.
- (4) Cut the workpiece along surface B manually.
- (5) Release the tool along the Z axis without moving along the X axis and stop spindle rotation.
- (6) Measure the distance " α " and specify G50X α and push and buttons in the MDI mode and the work coordinate system is set in the NC.

(When using the same tool as used to cut the A and B surfaces, the offset amount is zero. When a different tool is used, the difference between the tools should be stored as the offset value. See Section 6.5 in Chapter III for how to measure the offset value.)

(Method 2) Coordinate system setting after positioning to the start point.



To position the tool at the start point, follow the following procedure.

- (1) Cut the workpiece along surface A manually.
- (2) Release the tool along the X axis without moving along the Z axis and stop spindle rotation.
- (3) Reset the relative coordinate value of the Z axis (address W). (See Section 5.6 in Chapter IV.)
- (4) Measure distance " δ " in the above figure.
- (5) Cut the workpiece along surface B manually.
- (6) Release the tool along the Z axis without moving along the X axis and stop spindle rotation.
- (7) Reset the relative coordinate value of the X axis (address U). (See Section 5.6 in Chapter IV.)
- (8) Measure distance " γ ".
- (9) Position the tool at the point with the following coordinate values looking at the relative coordinate display.

$$x = \alpha - \gamma$$

$$z = \beta - \delta$$

x, z: relative coordinates

Here, α and β are the coordinate values of the start point.

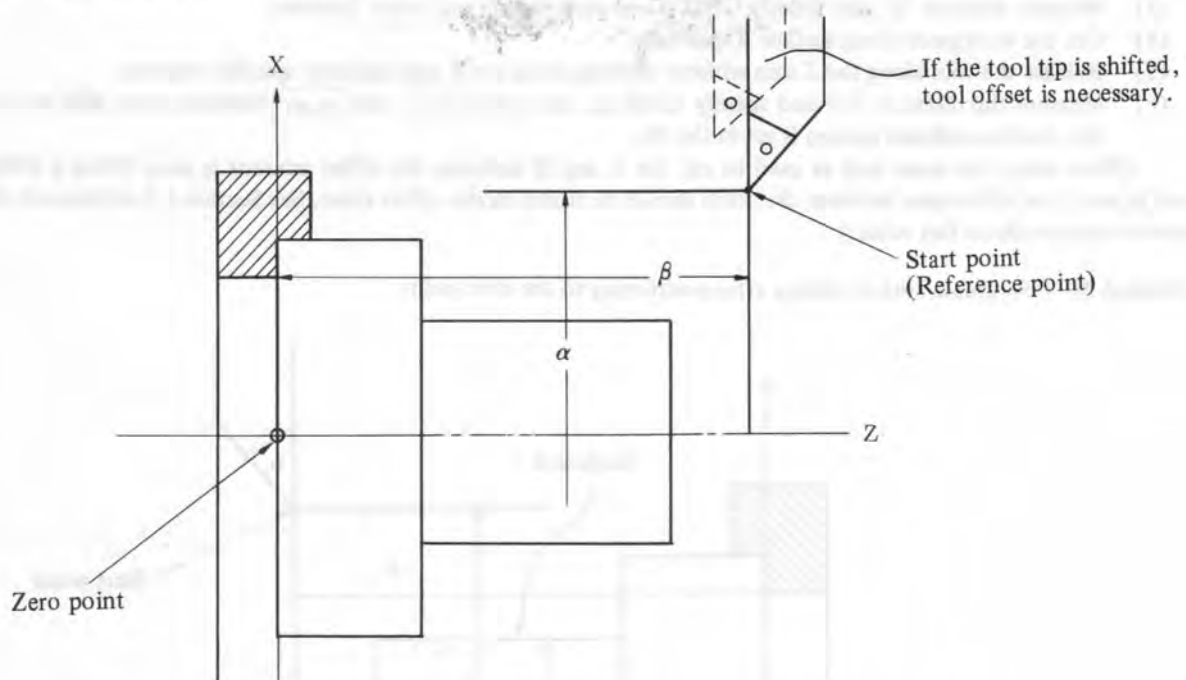
(If the tool is exchanged during actual machining, set the offset amount as in method 1.) If the G50 command is specified at the start of a program with the following coordinate dimensions after positioning at the start point, the desired work coordinate system is set at the start of the program execution.

G50(G92)X α Z β ;

(Method 3) Use the reference point as a start point

Positioning at the start point is performed by reference point return. Then follow method 2. The tool tip may not be positioned at the reference point but the tool standard point is.

The difference between these should be compensated by tool offset (see Section 6.1 in Chapter III) or by shifting the work coordinate system (see Chapter III, Section 5.8.).



(Method 4) Automatic coordinate system setting

If the corresponding parameter has been set, the work coordinate system is automatically set after manual reference point return.

Set coordinate values α and β to parameters 375 and 376 respectively.

The coordinate system, in which the reference point coordinate values are α and β , is set.

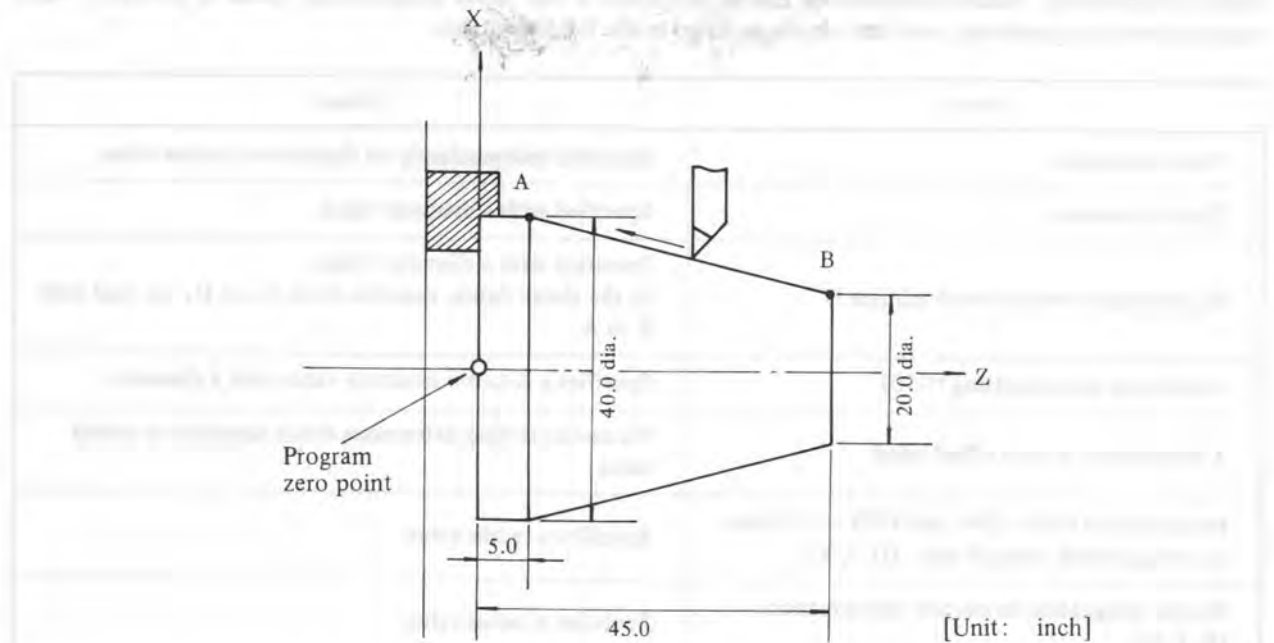
As the tool tip may not be positioned at the reference point (see method 3), the tool position should be compensated by using the tool offset function (see 6.1) or work coordinate system shift function (see 5.10).

3.6 Reference Point

The reference point is a fixed position on a machine tool. The reference point return function returns the tool to the reference point.

Accordingly, a program may not start from a certain position in the work coordinate system but may start from a reference point. In this case, because the reference point is a certain point on the machine tool and the program uses the work coordinate system with a zero point on the workpiece, a G50 command must be specified when the tool is returned to the reference point position to establish a work coordinate system.

3.7 Absolute and Incremental Programming



Command method		Address	Command specifying movement from B to A above
Absolute programming	Specifies an end point in the work coordinate system	X (Coordinate value on the X axis) Z (Coordinate value on the Z axis)	X400.0Z50.0 ;
Incremental programming	Specifies a distance from start point to end point	U (Distance along the X axis) W (Distance along the Z axis)	U200.0W-400.0 ;

(Note 1) Absolute and incremental commands can be used together in a block. In the above example, the following command can be specified:

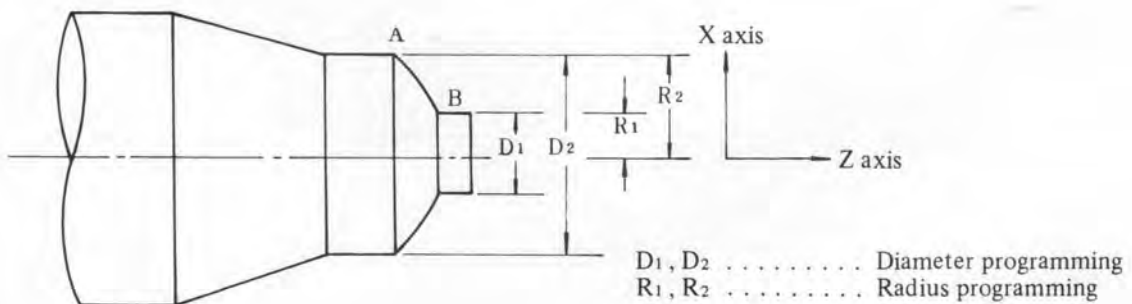
X400.0W-400.0 ;

(Note 2) When both X and U or W and Z are used together in a block, the one specified later is effective.

3.8 Diameter and Radius Programming

Since the work cross section is usually circular in NC lathe control programming, its dimensions can be specified in two ways:

Diameter and Radius



When the diameter is specified, it is called diameter programming and when the radius is specified, it is called radius programming. Radius programming can be performed if the radius programming option is provided. When using diameter programming, note the conditions listed in the following table.

Item	Notes
Z axis command	Specified independently of diameter or radius value.
X axis command	Specified with a diameter value.
Incremental command with address U	Specified with a diameter value. In the above figure, specifies from D ₂ to D ₁ for tool path B to A.
Coordinate system setting (G50)	Specifies a X axis coordinate value with a diameter.
X component of tool offset value	Parameter setting determines either diameter or radius value.
Parameters in G90 – G94, and G70 – G76 such as cutting depth along X axis. (D, I, K)	Specifies a radius value
Radius designation in circular interpolation (R, I, K)	Specifies a radius value
Feedrate along X axis	Change of radius/rev Change of radius/min.
Display of X-axis position	Displayed as diameter value.

(Note 1) In the following explanations, although the type of programming (diameter or radius) is not specified, a X axis graduation indicates a diameter value in diameter programming and a radius value in radius programming.

(Note 2) When a diameter value is specified for the tool offset value it indicates that when the outer diameter is cut with a new tool offset value, the outer diameter is changed by the offset value.
For example, when the offset value changes by 10 mm while the tool remains unchanged, the outer diameter changes by a diameter value of 10 mm.

(Note 3) When using a radius value for the tool offset value, the tool length itself can be set.

4. FEED FUNCTION

4.1 Rapid Traverse Rate

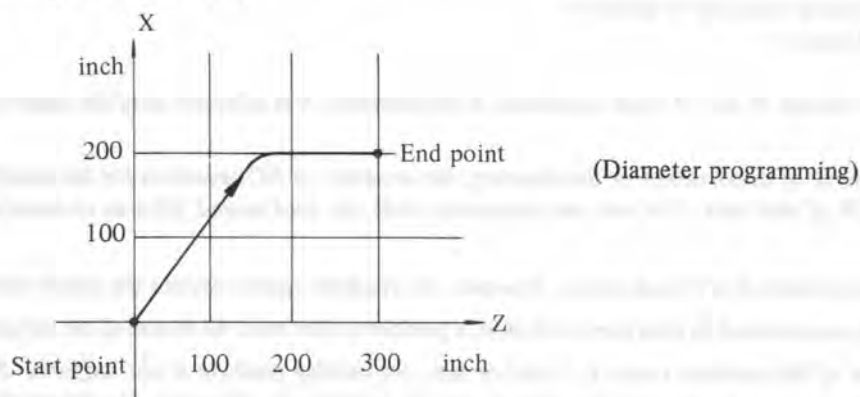
During rapid traverse, the machine moves at the specified rapid traverse rate for each axis.

This rapid traverse rate is usually determined and set (by parameter RPDFX, RPDFZ) by the machine tool builder before shipping.

Since the machine moves along each axis independently, the time it takes to move from the start to the end point on each axis are not equal.

For example, when the X and Z axis rapid traverse rates are 500 inches/min. and 800 inches/min. respectively and G00U200.0W300.0 is specified, it takes 12 sec. to move from the start to end point along the X axis and 22.5 sec. along the Z axis. If radius programming is specified for the X axis, the time is 24 sec.

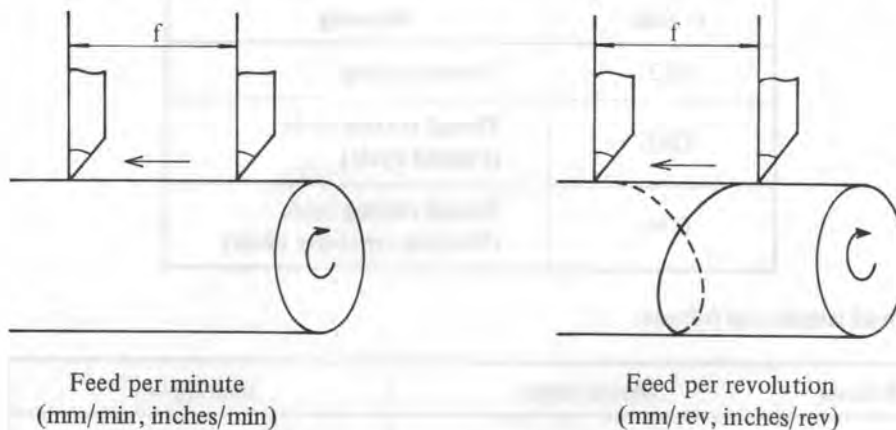
(Tool path in the above example)



An override can be applied to a rapid traverse rate by using the switch on the machine operator panel. (Fo, 25%, 50%, 100%) Fo is specified by parameter (RPDFL) and is not a percentage (%), but mm/min. or inches/min.

4.2 Feedrate

The feedrate (per revolution or per minute) is specified by the number following address F.



		Feed per minute	Feed per revolution
Meaning		Tool feed distance per minute	Tool feed distance per revolution
Programming address		F	F
Setting G code		G98	G99
Range	Metric input	1 ~ 15,000 mm/min. (F1 ~ F15000)	0.01 ~ 500.00 mm/rev (F1 ~ F50000)
	Inch input	0.01 ~ 600.00 inches/min. (F1 ~ F60000)	0.0001 ~ 50.0000 inches/rev (F1 ~ F500000)

	Feed per minute	Feed per revolution
Clamp value	The cutting feedrate is clamped at a constant rate. This value (clamp value) is specified by the machine tool builder. (An overridden feedrate is also clamped.)	
Override	Override from 0 to 200% can be applied in 10% increments.	

The clamp value is specified in mm/min. or inches/min. In feed per revolution, cutting feedrate should be converted to mm/min. or inches/min. by the following expression.

$$f_m = f_r \times R$$

f_m: Feed per minute (mm/min. or inch/min.)

f_r: Feed per revolution (mm/rev or inch/rev)

R: Spindle speed (rpm)

(Note 1) G98 and G99 are modal. If one of these commands is programmed, it is effective until the other command is specified.

(Note 2) Except in the course of acceleration or deceleration, the accuracy of NC operation for the specified feedrate is within $\pm 2\%$ of that rate. The rate was measured while the tool moved 500 mm or more under the normal status.

(Note 3) Up to 7 digits are permitted in F code input. However; the feedrate cannot exceed the clamp value.

(Note 4) If the feedrate is programmed in feed per revolution, a position coder must be mounted on the spindle.

(Note 5) If the revolution of the position coder is 1 rpm or less, the cutting feedrate is not uniform. If nonuniformity of revolution can be ignored, machining may be performed. However, the slower the spindle speed, the more nonuniform the cutting feedrate.

4.3 Lead of Thread

In thread cutting, the lead is specified by the number following address F or E. Thread cutting is specified by G32, G76 or G92.

G code	Meaning
G32	Thread cutting
G92	Thread cutting cycle (Canned cycle)
G76	Thread cutting cycle (Multiple repetitive cycle)

The range of lead length is as follows:

Address	Metric input	Inch input
F	0.01 ~ 500.00 mm	0.0001 ~ 50.0000 inch
E	0.0001 ~ 500.0000 mm	0.000001 ~ 9.999999 inch

The spindle speed is limited as follows:

$$R \leq \frac{\text{Max. feedrate}}{\text{Lead length of thread}}$$

(Provided that $R \geq 1$)

R: Spindle speed (rpm)

Lead: mm or inch

Max. feedrate: mm/min. or inches/min.

Max. feedrate is the smaller of the following rates;

Max. commanded feedrate in feed per minute and the max. feedrate limited by motor or machine capacity.

(Note 1) The spindle speed is read, (by a position coder mounted on the spindle), and converted into feed per minute and the tool is moved at the converted feedrate.

(Note 2) The override is not applied to the converted feedrate.

(Note 3) The converted feedrate is clamped.

Address F is used for thread lead specification as well as specification of feed per minute or feed per revolution. Address E is used only for thread lead length.

The E or F code specified last is effective for thread lead. For feed per minute and feed per revolution, the existing F code is effective regardless of E code. Therefore it is a good idea to use the E code for thread lead specification and the F code for feed per minute and feed per revolution specifications because it is unnecessary to respecify the feed rate after thread cutting.

Example (Programmed in the following sequence)	Effective feedrate/thread lead
G01Z F100;	F100
G32Z ;	F100
G32Z E200;	E200
G01Z ;	F100
Z F300;	F300
G32Z ;	F300
G32Z F100 E200;	E200
G32Z E200 F100;	F100

4.4 Feedrate Reduction 1/10

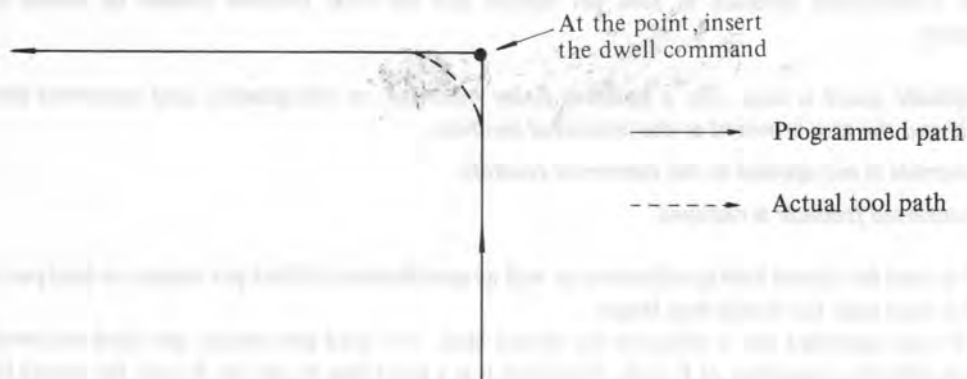
The metric feedrate input can be changed to one tenth of the usual increment by setting a parameter (FMIC).

Item	Least input increment	Range
Feed per minute	0.1 mm/min.	F1 ~ F150,000 (0.1 ~ 15000.0 mm/min.)
Feed per revolution	0.001 mm/rev	F1 ~ F500,000 (0.001 ~ 500.000 mm/rev)
Thread lead F command	0.001 mm	F1 ~ F500,000 (0.001 ~ 500.000 mm)
Thread lead E command	0.00001 mm	E1 ~ E9999999 (0.00001 ~ 99.99999 mm)

4.5 Automatic Acceleration and Deceleration

In feed start or feed stop, the acceleration or deceleration is applied automatically with a time constant so that the machine tool system is not jarred. Therefore, this need not be considered when programming.

Because of automatic acceleration and deceleration, corners are not cut sharply. The dwell command (G04) must be inserted between the two blocks to cut a sharp corner.



If the dwell command is inserted, the actual tool path matches the programmed path. The faster the feedrate and the larger the acceleration/deceleration time constant, the larger the error at the corner.

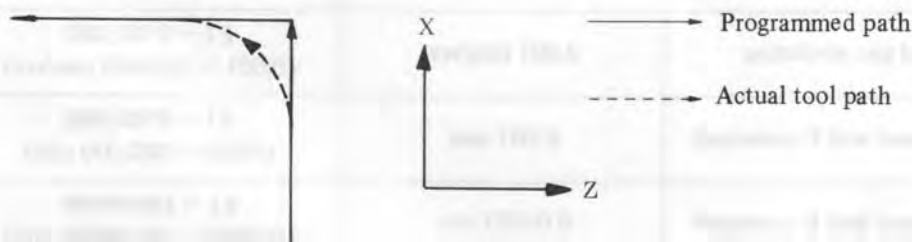
(Note 1) The following chart shows feedrate changes between blocks of information specifying different types of movement.

New block \ Previous block	Positioning	Feed	Not moving
	Positioning	Feed	Not moving
Positioning	X	X	X
Feed	X	O	X
Not moving	X	X	X

X: The next block is executed after commanded rate has decelerated to zero.

O: The next block is executed sequentially so that the feedrate is not changed by very much.

(Note 2) The actual tool path does not match the programmed path because acceleration and deceleration are applied along each axis (X and Z) independently and the feedrate for each axis is changed between blocks. For example, if the tool moves along the X axis only in one block and along the Z axis in the next block, the feedrate for the X axis decelerates while motion along the Z axis accelerates and the actual tool path is as follows.



In circular interpolation, the actual arc radius is smaller than that of the programmed arc.

(See the appendix.) This error can be minimized by making the acceleration/deceleration time constant of feedrate small.

5. PREPARATORY FUNCTION (G FUNCTION)

A 2-digit number following address G determines the meaning of the command for the concerned block. G codes are divided into the following two types.

Type	Meaning
One-shot G code	The G code is effective only in the block in which it is specified.
Modal G code	The G code is effective until another G code of the same group is specified.

(Example)

G01 and G00 are modal G codes in group 01.

G01X _____ ;
 Z _____ ;
 X _____ ;
 G00Z _____ ;

} G01 is effective in this range.

There are three series of G codes. One is the standard G code, one is the special G code B and the other is the special G code C. A parameter (GSP) is set to select either code. In this manual, the standard G code is written. The special G codes and the corresponding standard G codes shown in the following table have the same functions.

Only G90 and G91 are different. In special G codes, G90 indicates absolute commands (X and Z used in G90 mode are the same as X and Z in standard mode). G91 indicates incremental commands (X and Z used in G91 mode are the same as U and W in standard mode). However, when commands are input from the MDI, addresses X and Z indicate absolute commands and U and W indicate incremental commands regardless of the G90/G91 mode.

In special G code mode, addresses U and W specify an incremental move distance even in G90 mode.

Standard G code	Special G code B	Special G code C	Group	Function	Basic/Option
G00	G00	G00	01	Positioning (Rapid traverse)	B
G01	G01	G01		Linear interpolation (Feed)	B
G02	G02	G02		Circular interpolation CW	B
G03	G03	G03		Circular interpolation CCW	B
G04	G04	G04	00	Dwell	B
G07	G07	G07		Feedrate sine curve control	O
G10	G10	G10		Offset value setting	O
G20	G20	G70	06	Inch data input	O
G21	G21	G71		Metric data input	O
G22	G22	G22	04	Stored stroke limit ON	O
G23	G23	G23		Stored stroke limit OFF	O
G27	G27	G27	00	Reference point return check	O
G28	G28	G28		Return to reference point	O
G29	G29	G29		Return from reference point	O
G30	G30	G30		Return to 2nd reference point	O
G31	G31	G31		Skip cutting	O
G32	G33	G33	01	Straight and taper thread cutting	B
G34	G34	G34		Variable lead thread cutting	O
G36	G36	G36	00	Automatic tool compensation X	O
G37	G37	G37		Automatic tool compensation Z	O
G40	G40	G40	07	Tool nose radius compensation cancel	O
G41	G41	G41		Tool nose radius compensation left	O
G42	G42	G42		Tool nose radius compensation right	O

Standard G code	Special G code B	Special G code C	Group	Function	Basic/Option
G50	G92	G92	00	Programming of absolute zero point Maximum spindle speed setting	B O
G65	G65	G65		User macro simple calling	O
G66	G66	G66	12	User macro modal calling	O
G67	G67	G67		User macro modal call cancellation	O
G68	G68	G68	13	Mirror image for double turrets ON	O
G69	G69	G69		Mirror image for double turrets OFF	O
G70	G70	G72	00	Finishing cycle	O
G71	G71	G73		Stock removal in turning	O
G72	G72	G74		Stock removal in facing	O
G73	G73	G75		Pattern repeating	O
G74	G74	G76		Peck drilling on Z axis	O
G75	G75	G77		Grooving on X axis	O
G76	G76	G78		Thread cutting cycle	O
G90	G77	G20	01	Cutting cycle A	O
G92	G78	G21		Thread cutting cycle	O
G94	G79	G24		Cutting cycle B	O
G96	G96	G96	02	Constant surface speed control	O
G97	G97	G97		Constant surface speed control cancel	O
G98	G94	G94	05	Per minute feed	B
G99	G95	G95		Per revolution feed	B
—	G90	G90	03	Absolute programming	B
—	G91	G91		Incremental programming	B

(Note 1) The G codes marked with  are initial G codes in each group. That is, these G codes are set when the power is turned on or when the reset button is pressed while the system parameter that specifies to initialize G codes is effective. For G22 and G23, G22 is selected when the power is turned on. After reset, G22 or G23 (the one effective before resetting) is set.

For G00 and G01, G98 and G99, or G90 and G91, the value selected for the initial G codes is specified by parameter (G00, G98, G90).

For G20 and G21, the one in effect before the power was cut or the reset button was pressed is selected.

(Note 2) The G codes in the group 00 are not modal. They are effective only in the block in which they are specified.

(Note 3) An alarm occurs when a G code not listed in the above table is specified or when an optional G code not defined in the controller is specified (No. 010). However, G60 and G61 are ignored.

(Note 4) A number of G codes can be specified in a block even if they do not belong to the same group. When a number of G codes of the same group are specified, the G code specified last is effective.

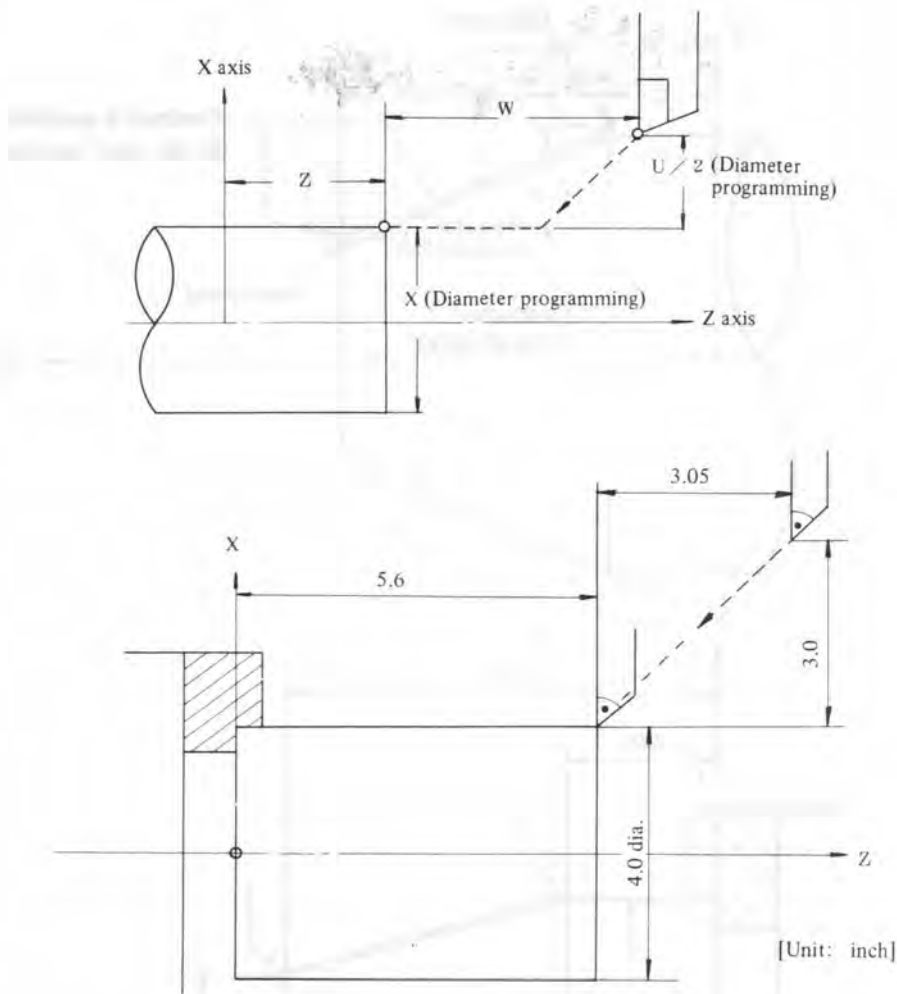
(Note 5) A G code from each group is displayed.

5.1 Positioning (G00)

G00 specifies positioning.

A tool moves to the (X, Z) position in the work coordinate system or from its current position to the position specified as the (U, W) distance at the rapid traverse rate along each axis independently.

G00X(U)—Z(W)—;



(Radius programming) G00X2.0Z5.6; or G00U-3.0W-3.05;

(Note 1) The rapid traverse rate in the G00 command is set for each axis independently by the machine tool builder. Accordingly, the rapid traverse rate cannot be specified in the address F.

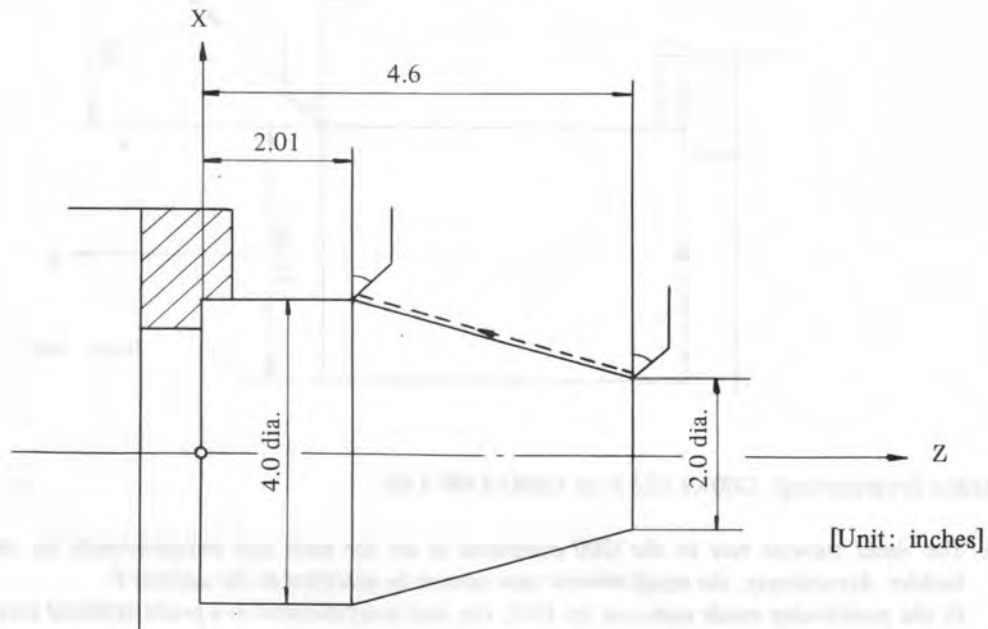
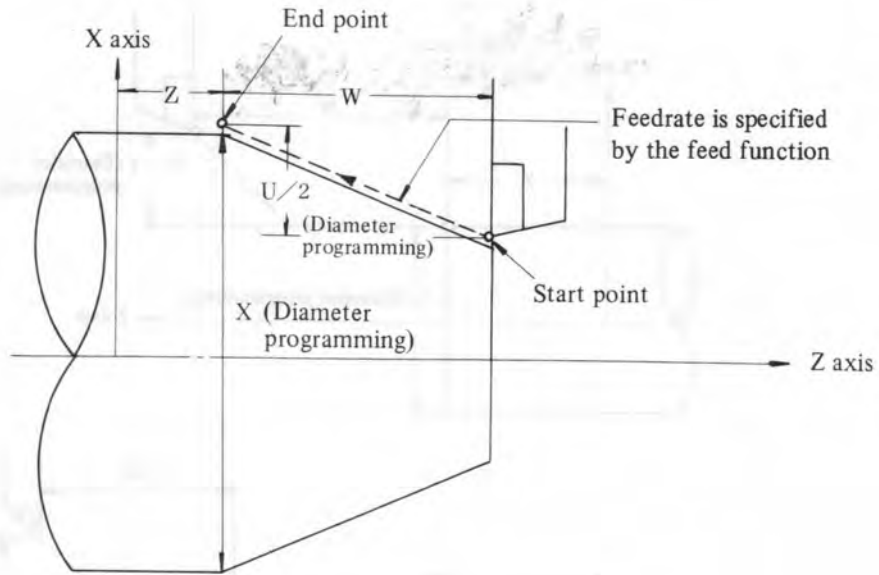
In the positioning mode actuated by G00, the tool is accelerated to a predetermined speed at the start of the block and is decelerated at the end of the block. Then if the parameter (CINP) specifying in-position checking has been set, execution proceeds to the next block after confirming the in-position. (See Note 2).

(Note 2) "In-position" means that the feed motor is within the specified range. (This range is determined by the machine tool builder).

5.2 Linear Interpolation (G01)

Linear interpolation moves a tool linearly to the (X, Z) position in the work coordinate system or from its current position to the position specified by the (U, W) values at the feedrate specified in the address F. Linear interpolation is specified by G01.

G01X(U) — Z(W) — F —;



(Diameter programming) G01X4.0Z2.01F2.0; or G01U2.0W-2.59F2.0;

The feedrate specified by the address F is the speed that the tool moves along a straight line.

If the feedrate has not yet been specified it (feedrate) is zero.

(Note 1) The feedrate along each axis is as follows:

$$\text{Feedrate along X axis: } F_x = \frac{L_x}{L} F$$

$$\text{Feedrate along Z axis: } F_z = \frac{L_z}{L} F$$

when F: Specified feedrate

L: Move distance

L_x: Move distance along X axis

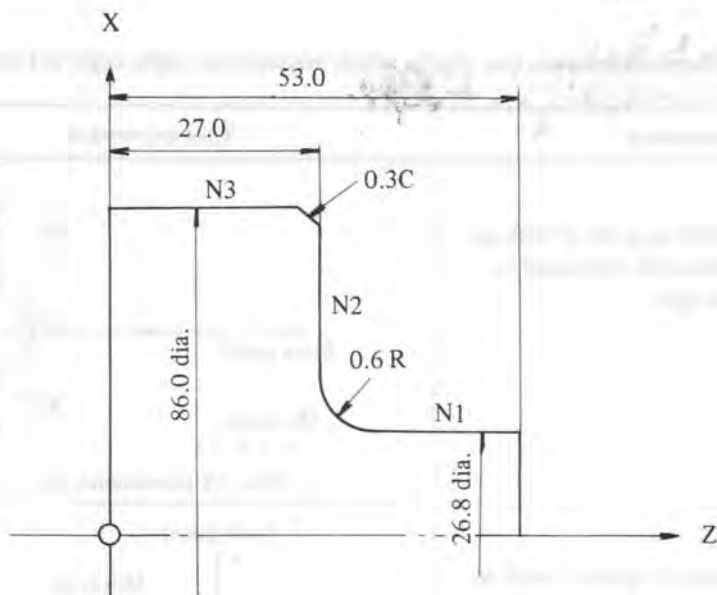
L_z: Move distance along Z axis

5.3 Chamfering and Corner R

A chamfer or corner arc can be inserted between two blocks which intersect at a right angle as follows:

Item	Command	Tool movement
Chamfering $Z \rightarrow X$	$G01Z(W)b \text{ I } \pm i ;$ ↑ Specifies movement to point b with an absolute or incremental command in the figure on the right.	Moves as $a \rightarrow d \rightarrow c$ (For $-X$ movement, $-i$)
Chamfering $X \rightarrow Z$	$G01X(U)b \text{ K } \pm k ;$ ↑ Specifies movement to point b with an absolute or incremental command in the figure on the right.	Moves as $a \rightarrow d \rightarrow c$ (For $-Z$ movement, $-k$)
Corner R $Z \rightarrow X$	$G01Z(W)b \text{ R } \pm r ;$ ↑ Specifies movement to point b with an absolute or incremental command in the figure on the right.	Moves as $a \rightarrow d \rightarrow c$ (For $-X$ movement, $-r$)
Corner R $X \rightarrow Z$	$G01X(U)b \text{ R } \pm r ;$ ↑ Specifies movement to point b with an absolute or incremental command in the figure on the right.	Moves as $a \rightarrow d \rightarrow c$ (For $-Z$ movement, $-r$)

I and K always specify a radius value.



(Diameter programming)

```
N1 Z27.0 R0.6 ;
N2 X86.0 K-0.3 ;
N3 Z0 ;
```

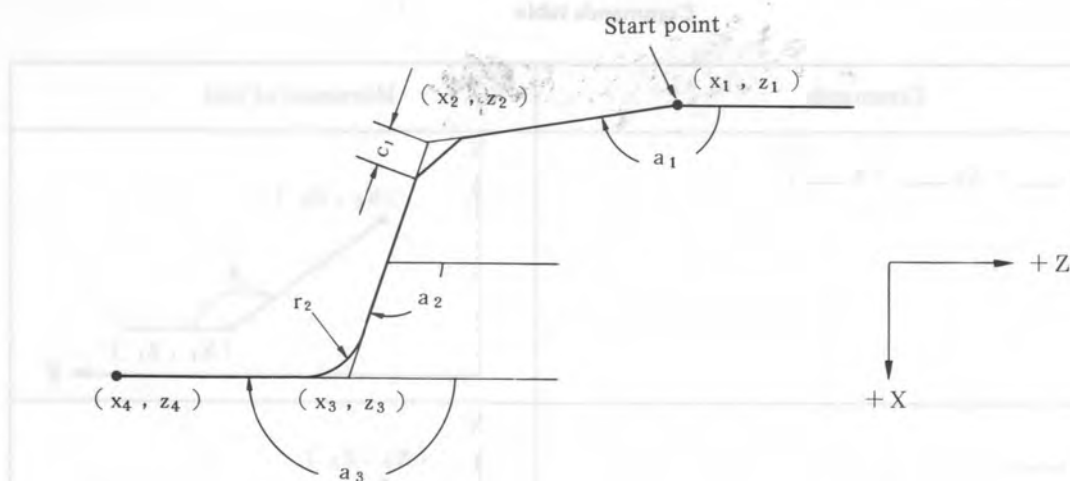
[Unit: inch]

- (Note 1) The movement for chamfering or corner R must be a single movement along the X or Z axis in G01 mode. The next block must be a single movement along the X or Z axis perpendicular to the former block.
- (Note 2) The movement specified in the next block must start from point b in the figure. Note that this is not point c. In incremental programming, the distance from point b must be specified.
- (Note 3) The following conditions generate alarms.
- 1) Both the X and Z axes were specified and either I, K or R was specified in G01 mode. (Alarm No. 054)
 - 2) Two of I, K and R were specified in the same block in G01 mode. (Alarm No. 053)
 - 3) Both X and I or Z and K were specified in G01 mode. (Alarm No. 056)
 - 4) The move distance along the X or Z axis is smaller than the chamfering or corner R value in the block in which the chamfering or corner R was specified. (Alarm No. 055)
 - 5) The motion specified in block after the block specifying chamfering or corner R is not perpendicular to motion specified in the former block in G01 mode. (Alarm No. 051, 052)
- (Note 4) In single block operation, the tool stops at point c rather than at point d.
- (Note 5) Chamfering and corner R cannot be used in a thread cutting block.

5.4 Direct Input of Drawing Dimensions

Angles of straight lines, chamfering values, corner rounding values, and other dimensional values on machining drawings can be programmed by directly inputting these values. In addition, the chamfering and corner rounding can be inserted between straight lines having an optional angle.

(Program example)



$X(x_2) Z(z_2), C(c_1);$

$X(x_3) Z(z_3), R(r_2);$

$X(x_4) Z(z_4);$

or

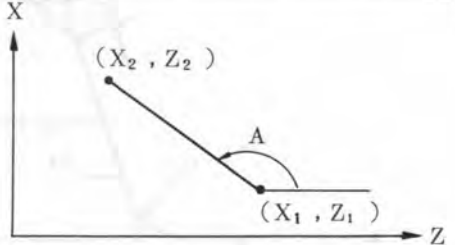
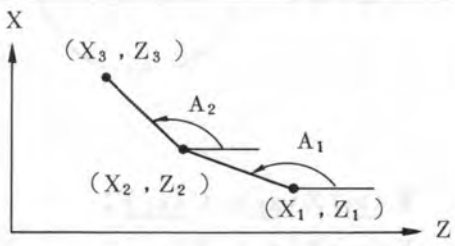
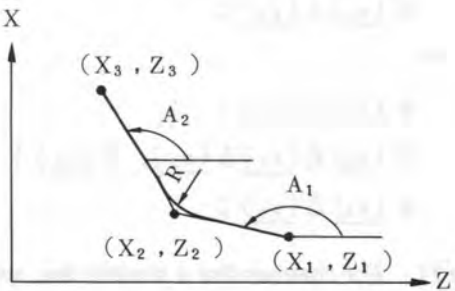
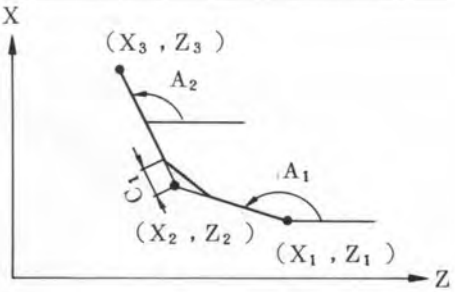
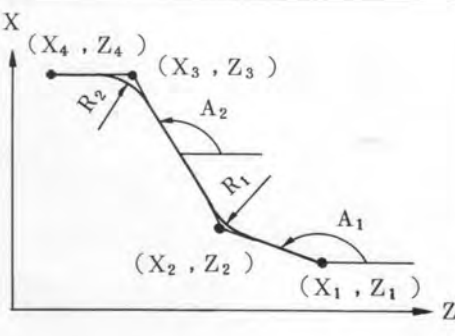
$A(a_1), C(c_1);$

$X(x_3) Z(z_3) A(a_2), R(r_2);$

$X(x_4) Z(z_4);$

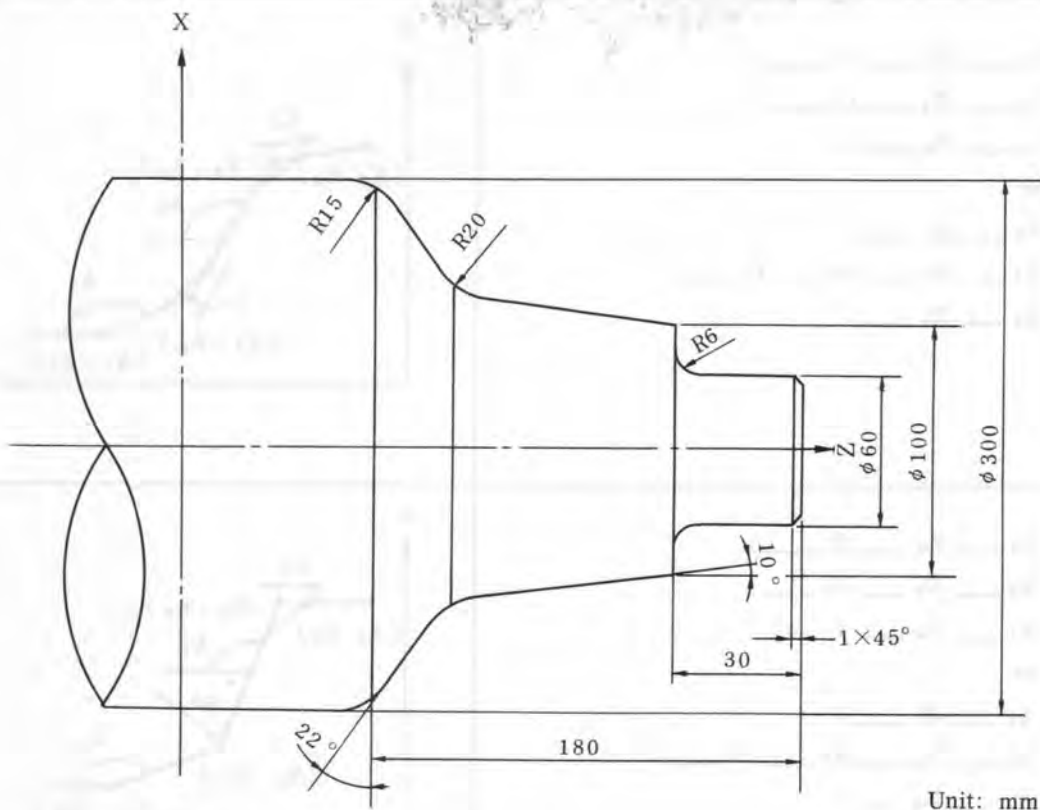
(Note 1) For commanding a straight line, specify one or two out of $X(U)$, $Z(W)$, and A . If only one is specified, the straight line must be primarily defined by a command in the next block. For designating the chamfering and corner rounding, specify them as $C-$ and $R-$ with a comma.

Commands table

	Commands	Movement of tool
1	$X_2 \text{ --- } (Z_2 \text{ ---}) A \text{ ---};$	
2	$A_1 \text{ ---};$ $X_3 \text{ --- } Z_3 \text{ --- } A_2 \text{ ---};$	
3	$X_2 \text{ --- } Z_2 \text{ ---}, R_1 \text{ ---};$ $X_3 \text{ --- } Z_3 \text{ ---};$ or $A_1 \text{ ---}, R_1 \text{ ---};$ $X_3 \text{ --- } Z_3 \text{ --- } A_2 \text{ ---};$	
4	$X_2 \text{ --- } Z_2 \text{ ---}, C_1 \text{ ---};$ $X_3 \text{ --- } Z_3 \text{ ---};$ or $A_1 \text{ ---}, C_1 \text{ ---};$ $X_3 \text{ --- } Z_3 \text{ --- } A_2 \text{ ---};$	
5	$X_2 \text{ --- } Z_2 \text{ ---}, R_1 \text{ ---};$ $X_3 \text{ --- } Z_3 \text{ ---}, R_2 \text{ ---};$ $X_4 \text{ --- } Z_4 \text{ ---};$ or $A_1 \text{ ---}, R_1 \text{ ---};$ $X_3 \text{ --- } Z_3 \text{ --- } A \text{ ---}, R_2 \text{ ---};$ $X_4 \text{ --- } Z_4 \text{ ---};$	

	Commands	Movement of tool
6	$X_2 \text{ --- } Z_2 \text{ ---}, C_1 \text{ ---};$ $X_3 \text{ --- } Z_3 \text{ ---}, C_2 \text{ ---};$ $X_4 \text{ --- } Z_4 \text{ ---};$ or $A_1 \text{ ---}, C_1 \text{ ---};$ $X_3 \text{ --- } Z_3 \text{ --- } A_2 \text{ ---}, C_2 \text{ ---};$ $X_4 \text{ --- } Z_4 \text{ ---};$	The diagram shows a coordinate system with X (vertical) and Z (horizontal) axes. A tool path is defined by four points: (X_1, Z_1), (X_2, Z_2), (X_3, Z_3), and (X_4, Z_4). The path consists of three segments: a horizontal segment from (X_1, Z_1) to (X_2, Z_2) with angle A_1, a curved segment from (X_2, Z_2) to (X_3, Z_3) with center C_1 and angle A_2, and a horizontal segment from (X_3, Z_3) to (X_4, Z_4) with center C_2.
7	$X_2 \text{ --- } Z_2 \text{ ---}, R_1 \text{ ---};$ $X_3 \text{ --- } Z_3 \text{ ---}, C_2 \text{ ---};$ $X_4 \text{ --- } Z_4 \text{ ---};$ or $A_1 \text{ ---}, R_1 \text{ ---};$ $X_3 \text{ --- } Z_3 \text{ --- } A_2 \text{ ---}, C_2 \text{ ---};$ $X_4 \text{ --- } Z_4 \text{ ---};$	The diagram shows a coordinate system with X (vertical) and Z (horizontal) axes. A tool path is defined by four points: (X_1, Z_1), (X_2, Z_2), (X_3, Z_3), and (X_4, Z_4). The path consists of three segments: a horizontal segment from (X_1, Z_1) to (X_2, Z_2) with angle A_1, a curved segment from (X_2, Z_2) to (X_3, Z_3) with center R_1 and angle A_2, and a horizontal segment from (X_3, Z_3) to (X_4, Z_4) with center C_2.
8	$X_2 \text{ --- } Z_2 \text{ ---}, C_1 \text{ ---};$ $X_3 \text{ --- } Z_3 \text{ ---}, R_2 \text{ ---};$ $X_4 \text{ --- } Z_4 \text{ ---};$ or $A_1 \text{ ---}, C_1 \text{ ---};$ $X_3 \text{ --- } Z_3 \text{ --- } A_2 \text{ ---}, R_2 \text{ ---};$ $X_4 \text{ --- } Z_4 \text{ ---};$	The diagram shows a coordinate system with X (vertical) and Z (horizontal) axes. A tool path is defined by four points: (X_1, Z_1), (X_2, Z_2), (X_3, Z_3), and (X_4, Z_4). The path consists of three segments: a horizontal segment from (X_1, Z_1) to (X_2, Z_2) with angle A_1, a curved segment from (X_2, Z_2) to (X_3, Z_3) with center C_1 and angle A_2, and a horizontal segment from (X_3, Z_3) to (X_4, Z_4) with center R_2.

(Program example) Diameter specification, metric input



```

N001 G50 X0.0 Z0.0 ;
N002 G01 X60.0 A90.0, C1.0 F80 ;
N003 Z-30.0 A180.0, R6.0 ;
N004 X100.0 A90.0 ;
N005 A170.0 R20.0 ;
N006 X300.0 Z-180.0 A112.0 R15.0 ;
N007 Z-230.0 A180.0 ;

```

(Note 2) Direct input commands of drawing dimensions are applicable to the memory operation only.

(Note 3) The following G codes are not applicable to the same block as commanded by direct input of drawing dimensions or between blocks of direct input of drawing dimensions which define sequential figures.

- (1) G codes (other than G04) in group 00.
- (2) G02, G03, G90, G92, and G94 in group 01.
- (3) G22 and G23 in group 04.

(Note 4) Corner rounding cannot be inserted into a threading block.

(Note 5) Neither chamfering and corner rounding commands specified in 5.3 nor those of direct input of drawing dimensions can be used concurrently.

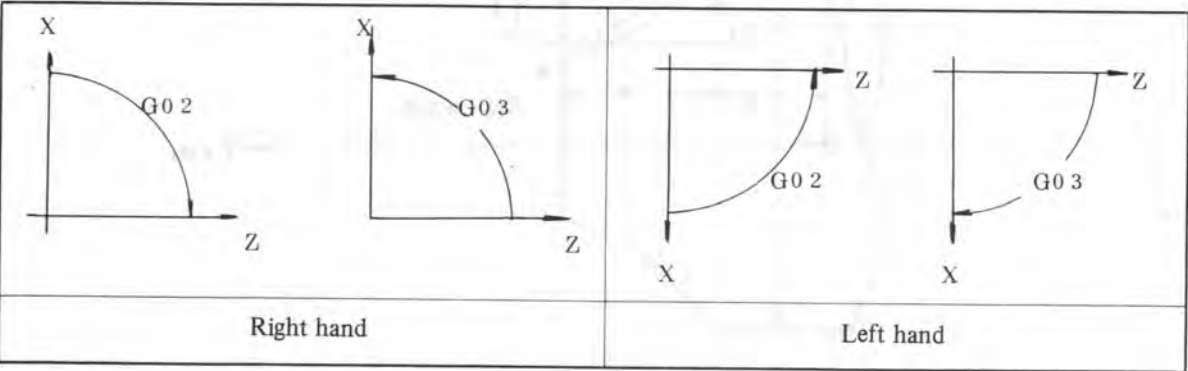
(Note 6) When the end point of the previous block is determined in the next block according to sequential commands of direct input of drawing dimensions, the single block stop is not done, but the feed hold stop is done at the end point of the previous block.

5.5 Circular Interpolation (G02, G03)

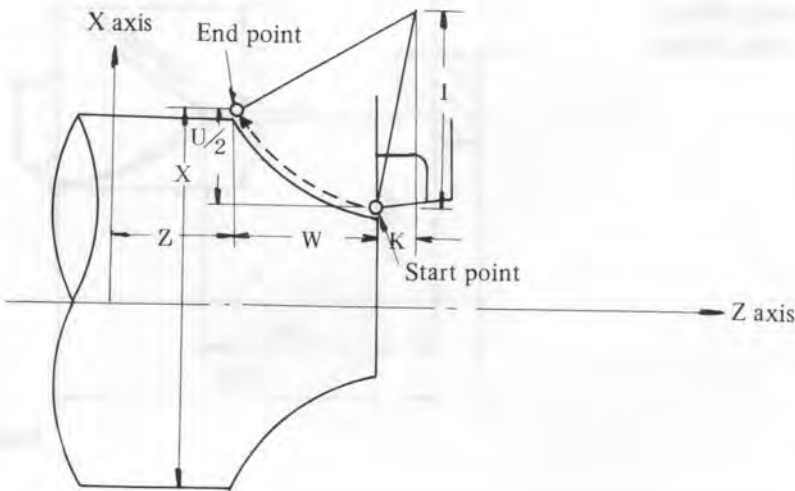
The following commands will move a tool along a circular arc. These commands are specified in the following format:

Data to be given		Command	Meaning
1	Rotation direction	G02	Clockwise direction (CW)
		G03	Counterclockwise direction (CCW)
2	End point position	X, Z	End point position (X, Z) in the work co-ordinate system.
	Distance to the end point	U, W	Distance from start point to end point.
3	Distance from start point to center	I, K	Distance from start point to center. (Always a radius value)
	Radius of arc.	R	Radius of arc. However an arc cannot be more than 180 degrees. (Always a radius value)

The clockwise or counterclockwise direction varies in right or left hand coordinate systems.

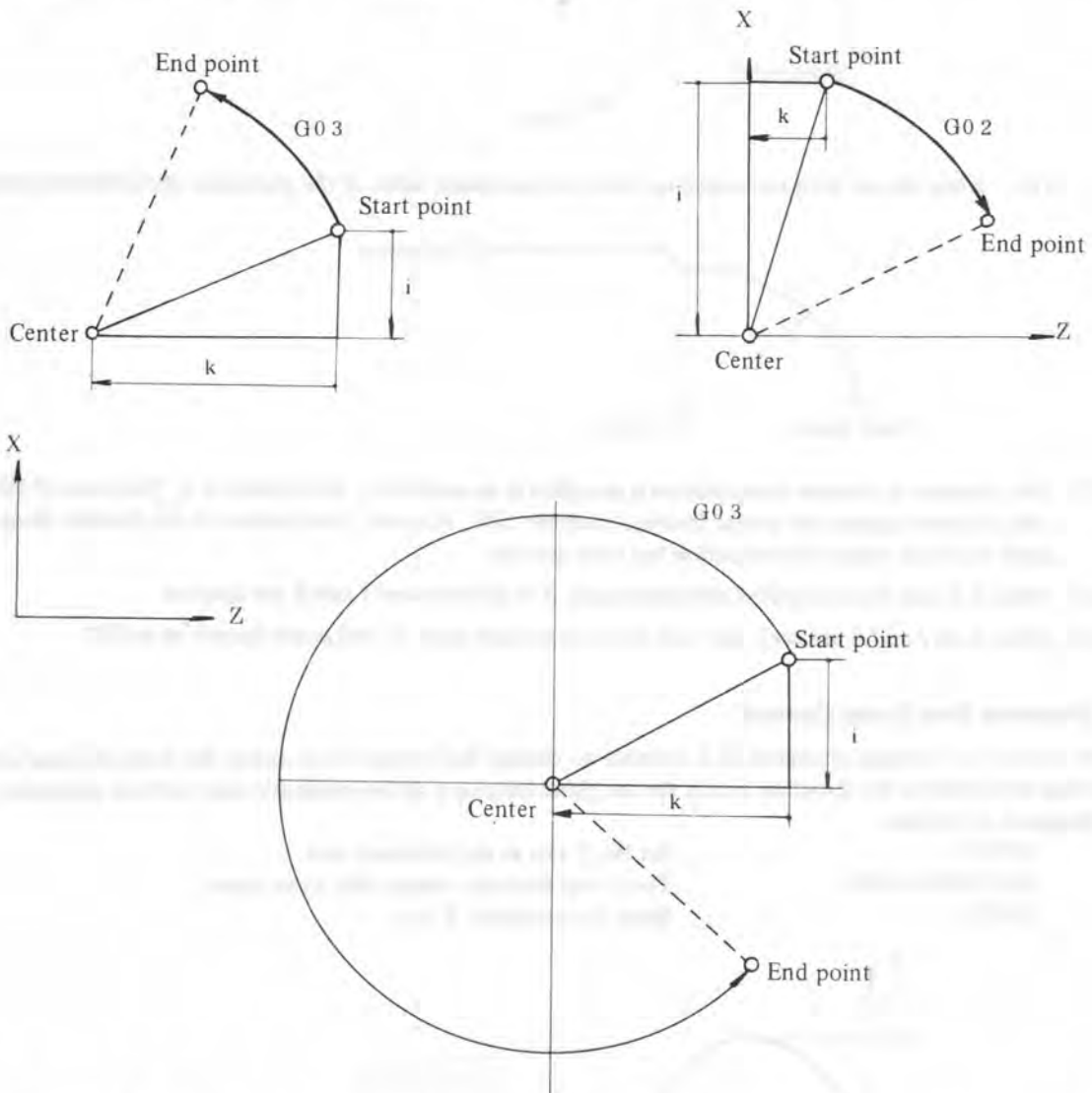


G02X(U)—Z(W)—I—K—F— ; (Diameter programming)



(Note 1) When either I or K is zero, the word can be omitted.

(Note 2) I and K specify the distance from the start point to the arc center along the X and Z axes respectively considering their signs.



(Note 3) When X, Z, U and W are all omitted, or when both U and W are zero or when X and Z are specified as the same position as the start position, and when the arc center is programmed using addresses I and/or K, a 360 degree circle (full circle) is assumed to have been specified.

Example: G02I—;

(Note 4) When address R is used, an arc of less than 180 degrees is assumed, so an arc of more than 180 degrees cannot be specified.

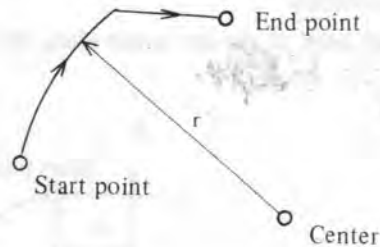
Accordingly: G02R—; and G03R—;

do not specify a full circle but a circle of zero degrees is assumed. Therefore the tool does not move.

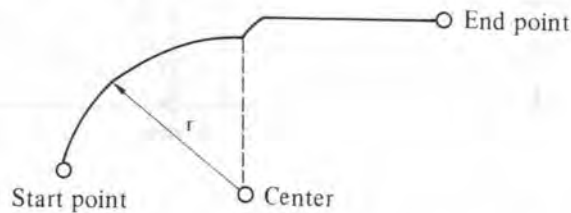
(Note 5) When address R specifies a value less than half of the distance between the start point and the end point, R is ignored and a half circle will be generated.

(Note 6) When the end point is not on the arc, the tool moves as shown in the following figures.

(i) After a tool has reached the coordinate value of the end point on one axis, it moves to the coordinate value of the end point on the other axis.



(ii) When the arc does not match up with the coordinate value of the end point, the following results.



(Note 7) The feedrate in circular interpolation is specified in an address F. See Section 4.2. The error of the specified feedrate against the actual feedrate is within $\pm 2\%$. However, this feedrate is the feedrate along an arc after tool nose radius compensation has been applied.

(Note 8) When I, K and R are specified simultaneously, R is effective and I and K are ignored.

(Note 9) When both I and K are zero, the tool moves from start point to end point linearly as in G01.

5.6 Feedrate Sine Curve Control

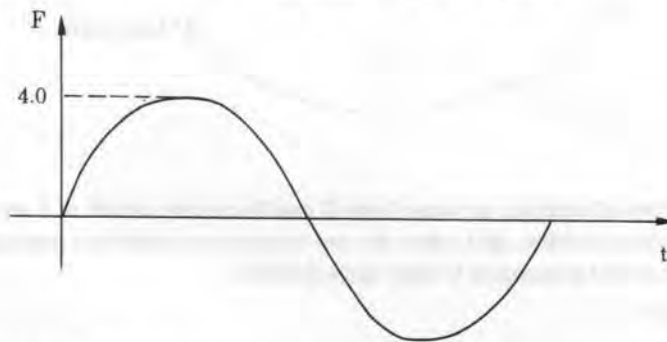
In circular arc cutting command, it is possible to change the feedrate of an axis in the form of a sine curve by conducting interpolation for the other axis in the arc plane (setting it as the imaginary axis) without moving it.

Designate as follows:

G07Z0 ; Set the Z axis as the imaginary axis.

G02X0Z0I10.0F4. ; The X-axis feedrate changes like a sine curve.

G07Z1 ; Reset the imaginary Z axis.



The imaginary axis is specified as follows.

G07 α 0 ; (α is specified as a imaginary axis.)

G07 α 1 ; (α is specified as a real axis)

(Note 1) The imaginary axis is effective for automatic operation only. It is ineffective for manual operation, and it moves.

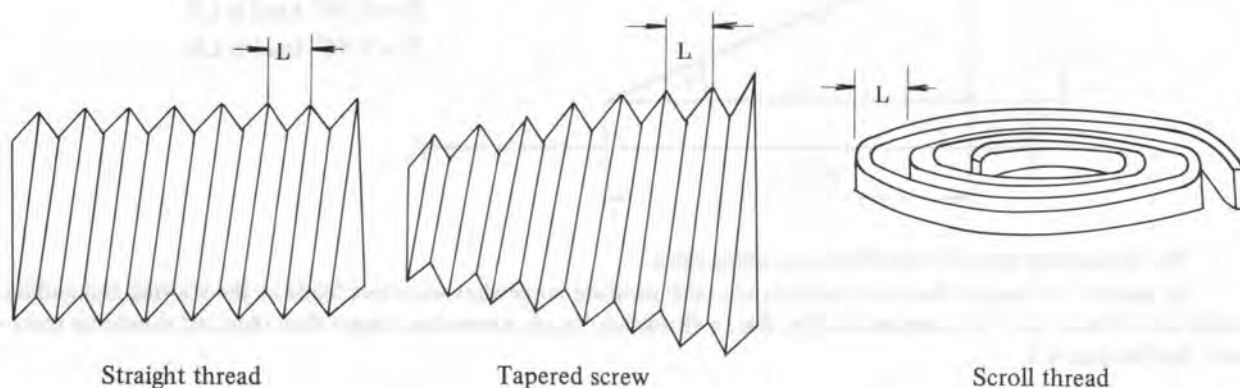
(Note 2) The interlock, stroke limit, and external deceleration are effective for the imaginary axis, too.

(Note 3) The handle interrupt is also effective for the imaginary axis. In other words, the axis moves by the handle interrupt component.

5.7 Thread Cutting (G32, G34)

5.7.1 Straight, taper thread cutting (G32)

Tapered screws and scroll threads in addition to straight threads can be cut by using a G32 command.



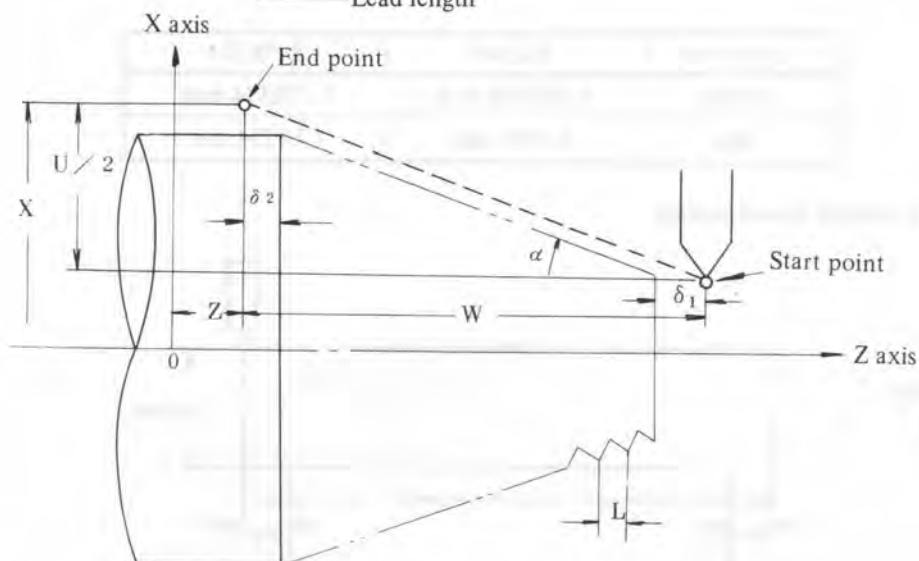
The command shown below is used for thread cutting and the lead is specified numerically following address F or E.

The end point is specified by coordinate position (X, Z) in the work coordinate system or by move distance (U, W) from the start point to end point.

G32X(U)___Z(W)___F___;

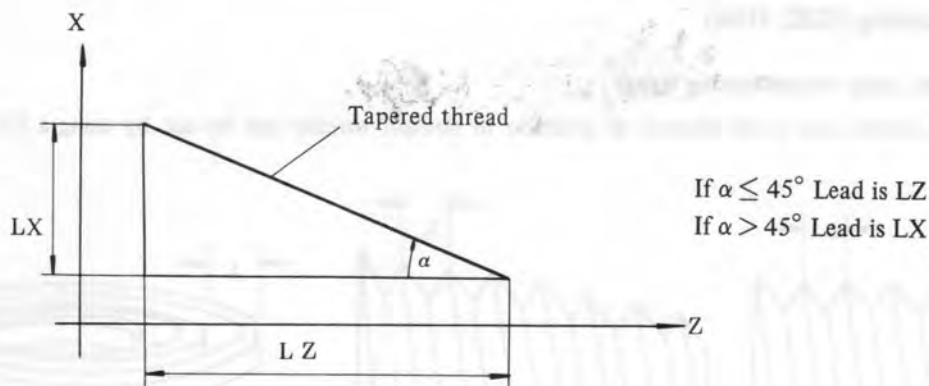
G32X(U)___Z(W)___E___;

Lead length



In general, thread cutting is repeated along the same tool path in rough cutting and finish cutting for a screw. Since thread cutting starts when the position coder mounted on the spindle detects a 1-turn signal, threading is started at a fixed point and the tool path on the workpiece is unchanged for repeated thread cutting. Note that the spindle speed must remain constant from rough cutting through finish cutting. If not, incorrect thread lead will occur.

In cutting tapered screws, the lead with the greater value in the X or Z direction must be specified.



The thread lead must be specified as a radius value.

In general, the lag of the servo system, etc. will produce somewhat incorrect leads at the starting and ending points of a thread cut. To compensate for this, a threading length somewhat longer than required should be specified. See Section 4.3.

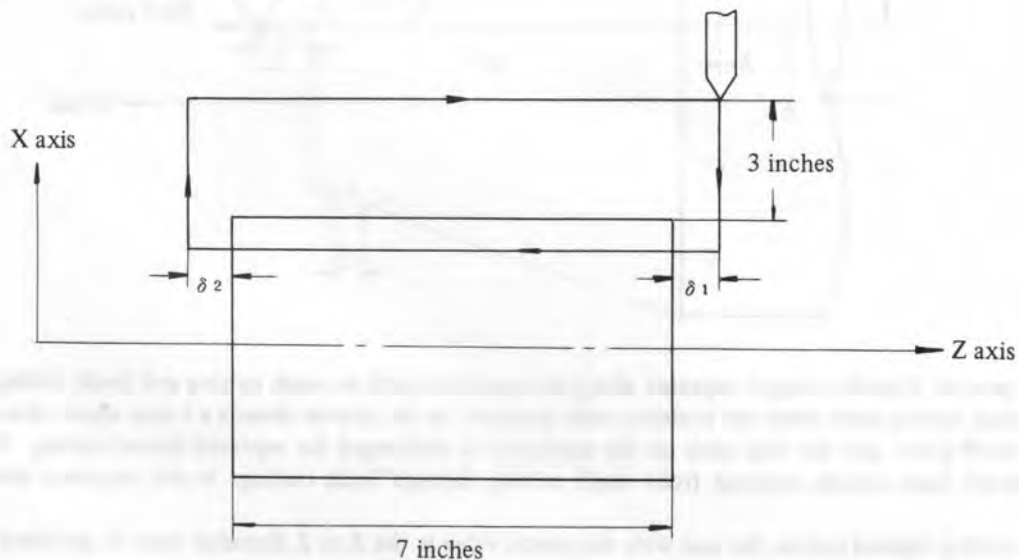
Example: Specified value and thread lead . . . F code designation

Input unit	F14	F5265
Inches	0.0014 inch	0.5265 inch
mm	0.14 mm	52.65 mm

Example: Command value and thread lead . . . E code designation

Input unit	E2346	E176534
Inches	0.002346 inch	0.176534 inch
mm	0.2346 mm	17.6534 mm

Example: 5.5.1.1 Straight thread cutting



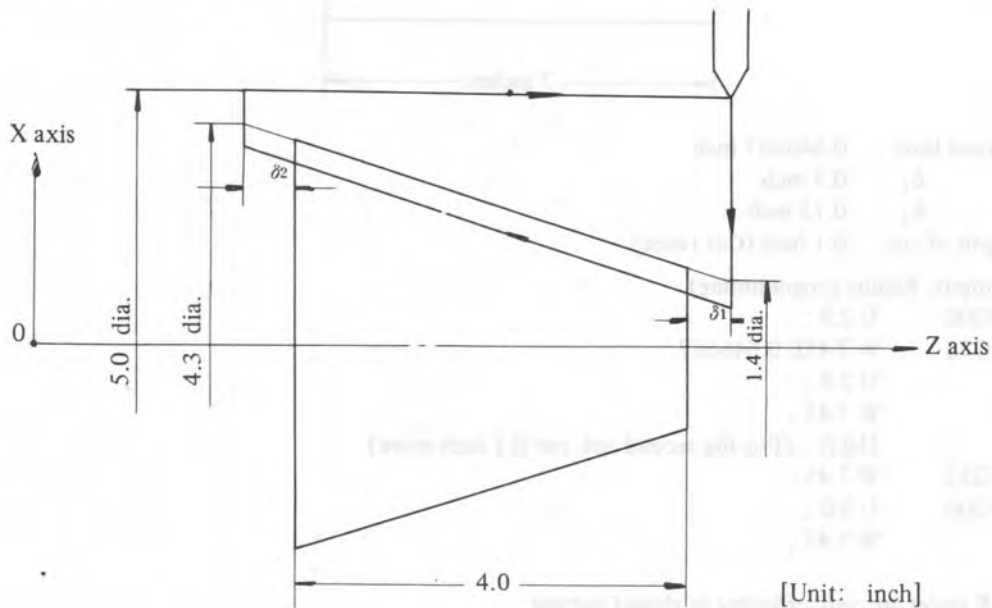
The following values are used in programming:

{ Thread lead : 0.4 inch
 δ_1 : 0.3 inch
 δ_2 : 0.15 inch
 } Depth of cut: 0.1 inch (Cut twice)

(Inch input, radius programming)

```
G00 U-3.1 ;
G32 W-7.45 F0.4 ;
G00 U 3.1 ;
      W 7.45 ;
      U-3.2 ; (For the second cut, cut 0.1 inch more)
G32 W-7.45 ;
G00 U 3.2 ;
      W-7.45 ;
```

Example: 5.5.1.2 Taper thread cutting



The following values are used in programming:

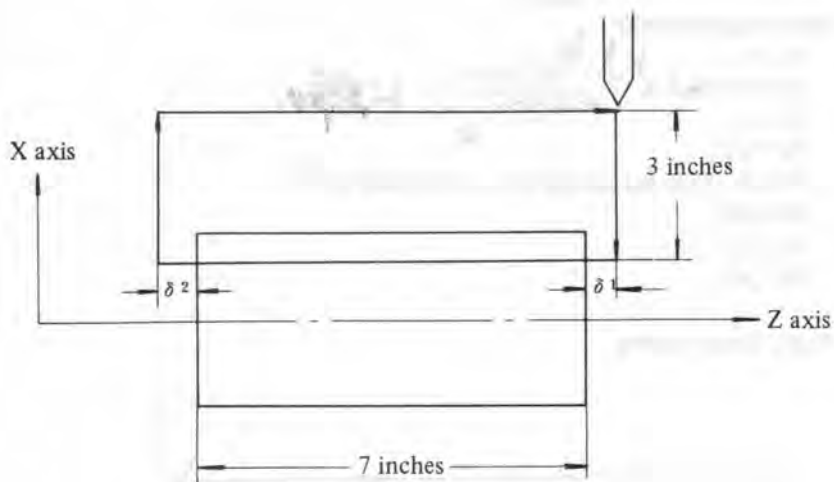
Thread lead : 0.35 inch in the direction of the Z axis
 δ_1 : 0.2 inch
 δ_2 : 0.1 inch
 Cutting depth in the X axis direction is 0.1 inch (Cut twice)

(Inch input, Diameter programming)

```
G00 X 1.2 ;
G32 X 4.1W-4.3 F0.35 ;
G00 X 5.0 ;
      W 4.3 ;
      X 1.0 ;
G32 X 3.9W-4.3 ;
G00 X 5.0 ;
      W 4.3 ;
```

Example: 5.5.1.3 Precision thread cutting

Specifying the thread lead with an E code enables accurate thread lead cutting.



{ Thread lead : 0.846667 inch
 δ_1 : 0.3 inch
 δ_2 : 0.15 inch
 Depth of cut: 0.1 inch (Cut twice)

(Inch input, Radius programming)

```

G00    U-2.9 ;
G32    W-7.45E 0.846667 ;
        U 2.9 ;
        W 7.45 ;
        U-3.0 ; (For the second cut, cut 0.1 inch more)
G32    W-7.45 ;
G00    U 3.0 ;
        W 7.45 ;
  
```

(Note 1) E codes are only effective in thread cutting.

(Note 2) If an F code is specified, the E code value is deleted. See Section 4.2.3.

(Note 3) Feedrate override is ineffective (fixed at 100%) during thread cutting.

(Note 4) A parameter is used to enable or disable a dry run during thread cutting.

(Note 5) It is very dangerous to stop feeding the thread cutter without stopping the spindle. This will suddenly increase the cutting depth. Thus, the feed hold function is ineffective while thread cutting. If the feed hold button is pressed during thread cutting, the tool will stop after a block not specifying thread cutting is executed as if the SINGLE BLOCK button were pushed.

However, the feed hold lamp (SPL lamp) lights when the FEED HOLD button on the machine control panel is pushed. Then, when the tool stops, the lamp is turned off (Single Block stop status).

(Note 6) When the FEED HOLD button is again pushed during the first block not specifying thread cutting or when it has been continuously pushed, the tool stops.

(Note 7) When thread cutting is executed in the single block status, the tool stops after execution of the first block not specifying thread cutting block.

(Note 8) When the mode is changed from automatic operation to manual operation during thread cutting, the tool stops at the first block not specifying thread cutting as when the feed hold button is pushed as mentioned in (Note 6).

However, when the mode was changed from one automatic operation mode to another, the tool stops after execution of the block not specifying thread cutting block as for the single block mode (Note 5).

(Note 9) When the previous block was a thread cutting block, cutting will start immediately without waiting for detection of the 1-turn signal even if the present block is a thread cutting block.

G32Z— F—;

Z—;

(A 1-turn signal is not detected before this block.)

G32;

(Even this block is regarded as a thread cutting block.)

Z— F—;

(Accordingly, a 1-turn signal is also not detected.)

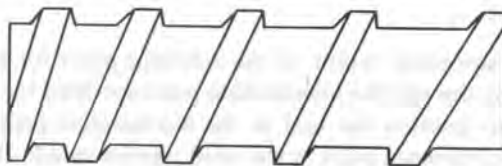
- (Note 10) When thread cutting blocks are consecutive, incorrect thread will occur for 0.01 seconds between the two blocks. That is, the feedrate changes for 0.01 seconds. When cutting consecutive short screws, if buffering is not in time, incorrect thread will occur for a longer period of time.
- (Note 11) Because constant surface speed control is effective during scroll thread or tapered screw cutting and the spindle speed changes, the correct thread lead may not be cut. Accordingly, constant surface speed control should not be used during thread cutting.
- (Note 12) A movement block preceding the thread cutting block must not specify chamfering or corner R.
- (Note 13) A thread cutting block must not specify chamfering or corner R.
- (Note 14) The spindle speed override is effective in thread cutting mode. However, if the override is applied to thread cutting, the accurate thread cannot be cut by the delay of the servo system and so on. On the other hand the spindle speed override can be made ineffective by changing a parameter (SOVC).
- (Note 15) The "Temporary stop during thread cutting" is not effective for G32.

With continuous threading option, it is possible to continue threading blocks without producing incorrect threading portion between blocks.

Do not change the longer axis (Normally Z axis ; X axis for face threading) between blocks.

5.7.2 Variable-lead thread cutting (G34)

Specifying an increment or a decrement value for a lead per screw revolution enables variable-lead thread cutting to be performed.



Variable-lead screw

K specifies an increment or a decrement per screw revolution in the following command.

G34X(U) — Z(W) — $\left\{ \begin{array}{l} F \\ E \end{array} \right\} K$ — ;

Addresses other than K are the same as in straight/taper thread cutting with G32.

K can take values of:

Metric input	$\pm 0.0001 \sim \pm 100.0000$ mm/rev
Inch input	$\pm 0.000001 \sim \pm 1.000000$ inches/rev

E can be used only with E 6-digit option provided.

(Note 1) A lead increased or decreased beyond the command range generates the alarm (014).

(Note 2) "Temporary Stop during Thread Cutting" is not effective for G34.

5.7.3 Continuous threading

It is possible to designate the threading blocks (G32, G34) continuously. The dead time in the pulse distribution between the threading blocks (max. 8 msec.) is eliminated. Therefore, threading can be conducted correctly without damaging the threads even when the same point is threaded repeatedly by changing the depth of cut.



5.8 Automatic Reference Point Return (G27 ~ G30)

5.8.1 G27-Reference point return check

The reference point is a fixed point on the machine and manual reference point return moves a tool to the reference point.

A G27 command is used to confirm whether or not the tool has reached the reference point in a program executing reference point return.

G27X(U) — Z(W) —;

The tool moves to the specified position at the rapid traverse rate when using the above command. If the tool has reached the reference point, the reference point return lamp goes on. If the reference point is reached on only one axis, the reference point return lamp for that axis goes on. If the reference point on the specified axis is not reached an alarm (No. 092) is displayed.

(Note 1) The position specified by the G27 command will be shifted by the offset value if an offset has been specified. Accordingly, the lamp does not go on unless a tool which has been offset reaches the reference point. Normally, offset should be cancelled before G27 is specified.

(Note 2) When making metric input to an inch machine the program may miss the reference point by 0.0001 mm and yet the reference point return lamp will still go on. This is because the least command increment is smaller than the input unit.

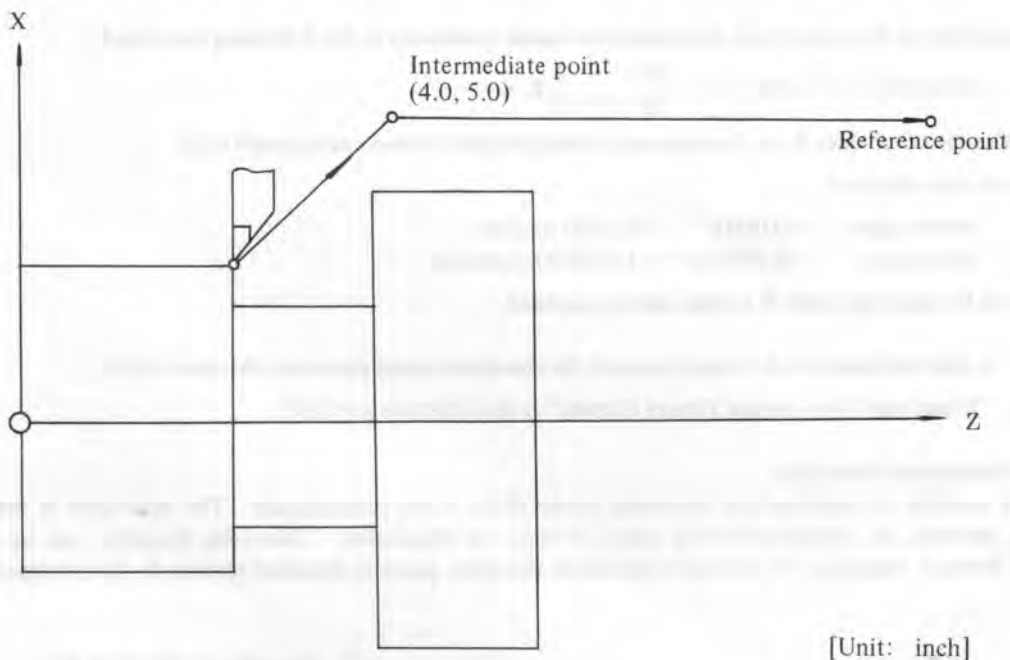
5.8.2 G28 Automatic return to reference point

G28X(U) — Z(W) —;

This command specifies automatic return to the reference point for the specified axes. Reference point positioning is made by first going through the intermediate point specified by X(U) — Z(W) —;

The G28 block is used to position the tool at the intermediate point of all specified axes at the rapid traverse speed, then move to the reference point at the rapid traverse speed. If the machine lock has not been set, the Reference Point Return lamp goes on.

Intermediate point or to reference point positioning is equivalent to the positioning with G00.



G28X4.0Z5.0 ;

In general, this command is used for automatic tool changing (ATC). Therefore, for safety, the tool nose radius compensation and tool offset should be cancelled before executing this command.

(Note 1) When the G28 command is specified when manual return to the reference point has not been performed after the power has been turned on, the movement from the intermediate point is the same as in manual return to the reference point. In this case, the direction from the intermediate point is equal to that for return to reference point selected by parameter setting.

(Note 2) The coordinate value of the end point in G28 command is memorized for all axes as is the coordinate value of the intermediate point. In other words, for axes not specified in the G28 block, the current coordinate value is taken as the coordinate value of the intermediate point for the axis.

Example:

```
N1X1.0Z2.0;
N2G28X4.0;      Intermediate point (4.0, __)
N3G28Z6.0;      Intermediate point (4.0, 6.0)
```

5.8.3 G29 Automatic return from reference point

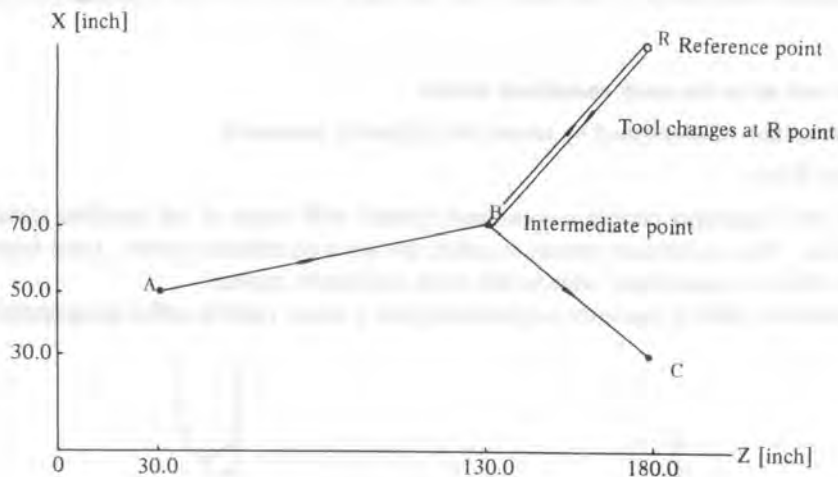
G29X—Z—;

This command is used to move the tool to the specified position through an intermediate point, for the specified axes. In general, this command is used immediately following a G28 or G30 command.

In an incremental command, the incremental distance from the intermediate point must be specified.

In a G29 block, the tool is moved to the intermediate point on all specified axes defined by the earlier G28 or G30 command at the rapid traverse speed, then it is moved to the specified end point at rapid traverse rate.

Example of application of G28 and G29



(Radius programming)

```
G28U20.0W100.0; (Program of A to B)
T0202;
G29U-40.0W50.0; (Program of B to C)
```

As shown in the above example, the programmer is not required to calculate the actual travelling distance from the intermediate point to the reference point.

5.8.4 G30 2nd reference point return

The command below is used to move the tool to the second reference point on the specified axes.

G30X(U)—Z(W)—;

The second reference point is determined by parameter setting as a distance from the reference point with field adjustment.

This function is the same as reference point return executed with G28 except that the tool does not return to the first reference point, but to a second reference point. A G29 command following a G30 command is used to position the tool to the commanded position through the intermediate point which was established by the G30 command.

This motion is the same as that executed by G29 following a G28 command. A G30 command is generally used when the automatic tool change (ATC) position differs from the reference point.

(Note 1) Manual reference point return or automatic reference point return (G28) must be performed before using this G30 command.

5.9 Dwell (G04)

G04X (t) ; or G04U (t) ; or G04P (t) ;

Any one of these command can be used for dwelling. Upon completion of the previous block (t) in sec. elapses before beginning the next block.

The maximum command time is 99999.999 seconds. The error of time t is within 16 ms.

Example: Dwell for 2.5 seconds

G04X2.5; or G04U2.5; or G04P2500;

(Note 1) A decimal point cannot be specified in Address P.

(Note 2) Dwell starts in the following two ways.

Parameter CIMP determines which is effective.

(1) After the speed of a movement specified in the earlier block becomes zero.

(2) After the tool is positioned at the command value. (After in-position is checked.)

5.10 Programming of Absolute Zero Point (G50)

The coordinate system must be set in advance when moving a tool to a certain position with an absolute command.

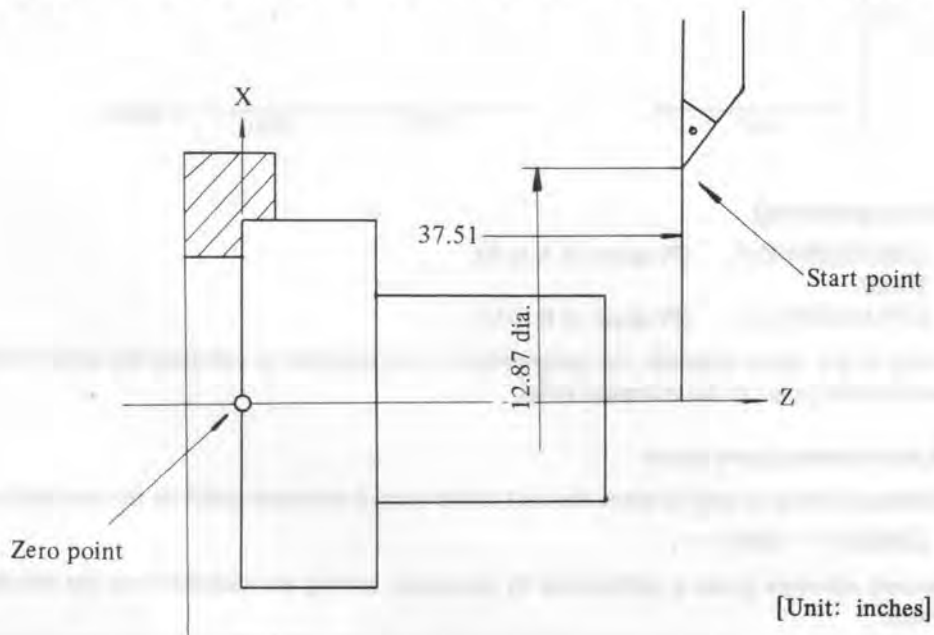
5.10.1 Command used to set the work coordinate system

The coordinate system is established by issuing the following command;

G50X (x) Z (z) ;

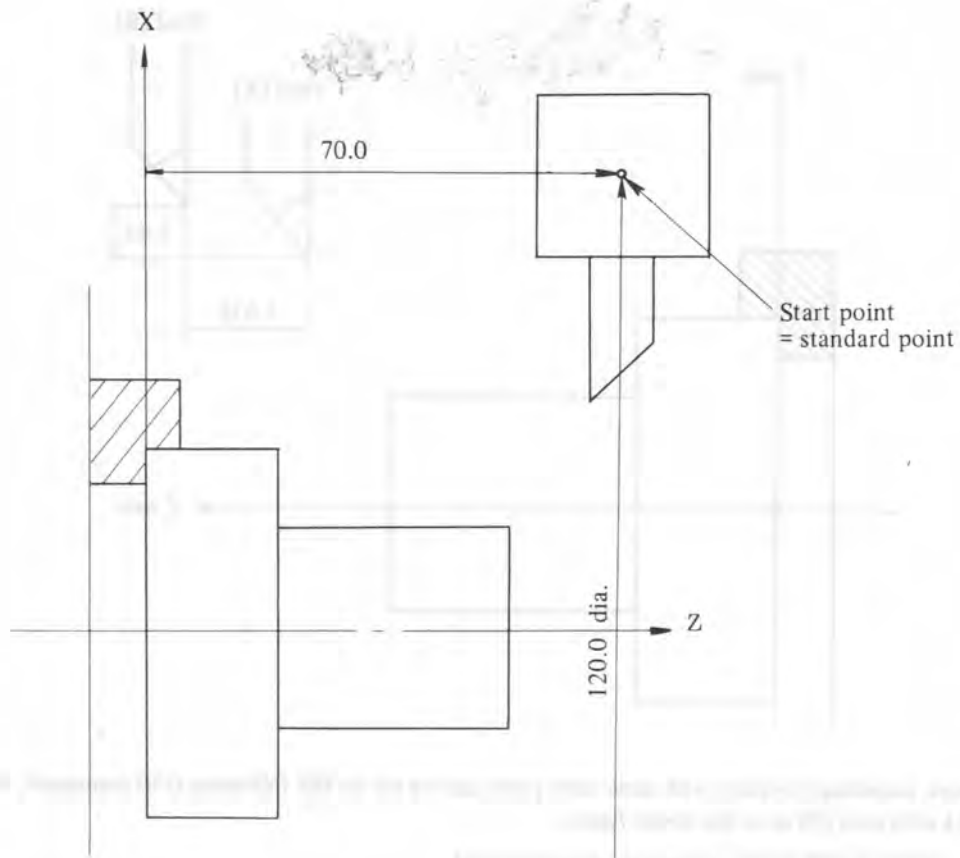
Execution of this command creates a coordinate system with origin at the specified distances (X and Z) from the tool position. This coordinate system is called the work coordinate system. Once created, subsequent absolute commands refer to a coordinate value in this work coordinate system.

Value x is a diameter value in diameter programming and a radius value in radius programming.



G50X12.87Z37.51; (Diameter programming)

As shown in the above program, G50 is used to insure that the tool tip coincides with the program start point and should be used at the start of a program.



G50X60.0Z70.0; (Radius programming)

As shown above, G50 should be at the start of the program to line up the tool point and the program start point.

If an absolute command is performed, the standard point moves along the commanded path. In order to move the tool tip along the programmed path, the distance from tool tip to the standard position must be compensated for by using the tool offset value.

(Note 1) When the coordinate system is set with a G50 command during offset mode, a coordinate system in which coordinate value of a tool excluding the offset value is the specified position, is established.

(Note 2) The tool nose radius compensation is cancelled temporarily with a G50 command.

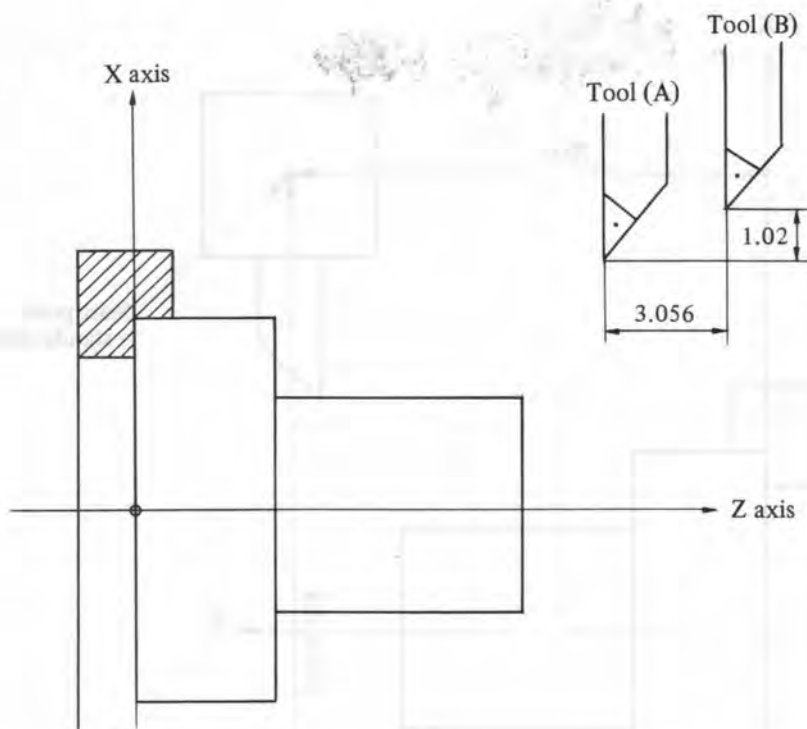
5.10.2 Shifting the work coordinate system

The work coordinate system can be shifted by using the following command.

G50U(u) W(w);

In the new shifted work coordinate system, the coordinate values of the tool tip are $(x + u)$ and $(z + w)$ when the old coordinate values were x and z .

Values x and u should be diameter value in diameter programming, or radius values in radius programming.



A work coordinate system with same zero point can be set by the following G50 command, when exchanging tool (A) with tool (B) as in the above figure.

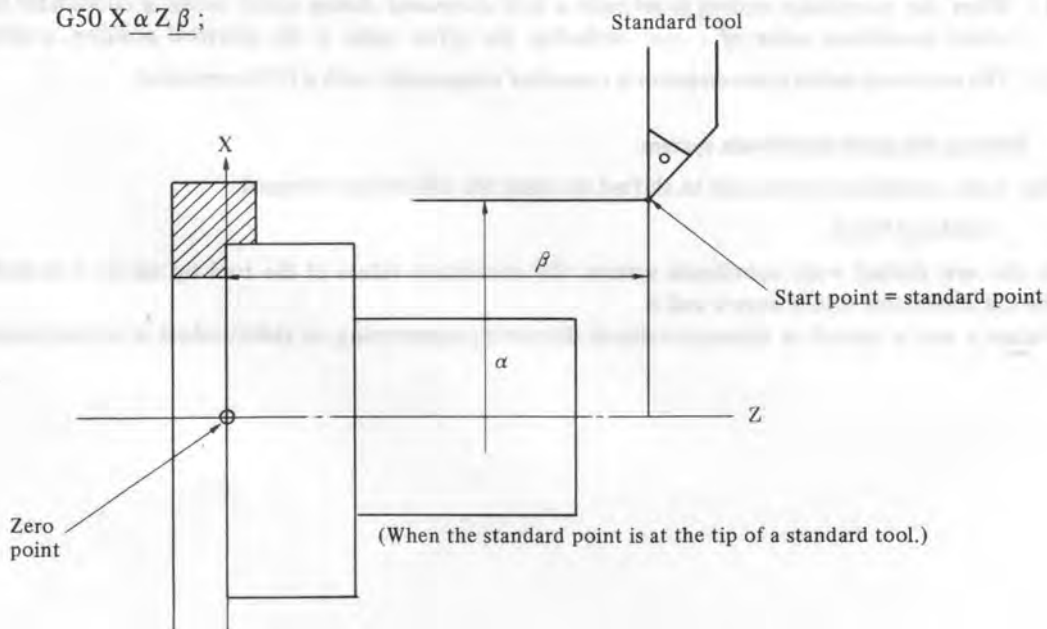
G50U 2.04W 3.056; (diameter programming)

5.10.3 Automatic coordinate system setting

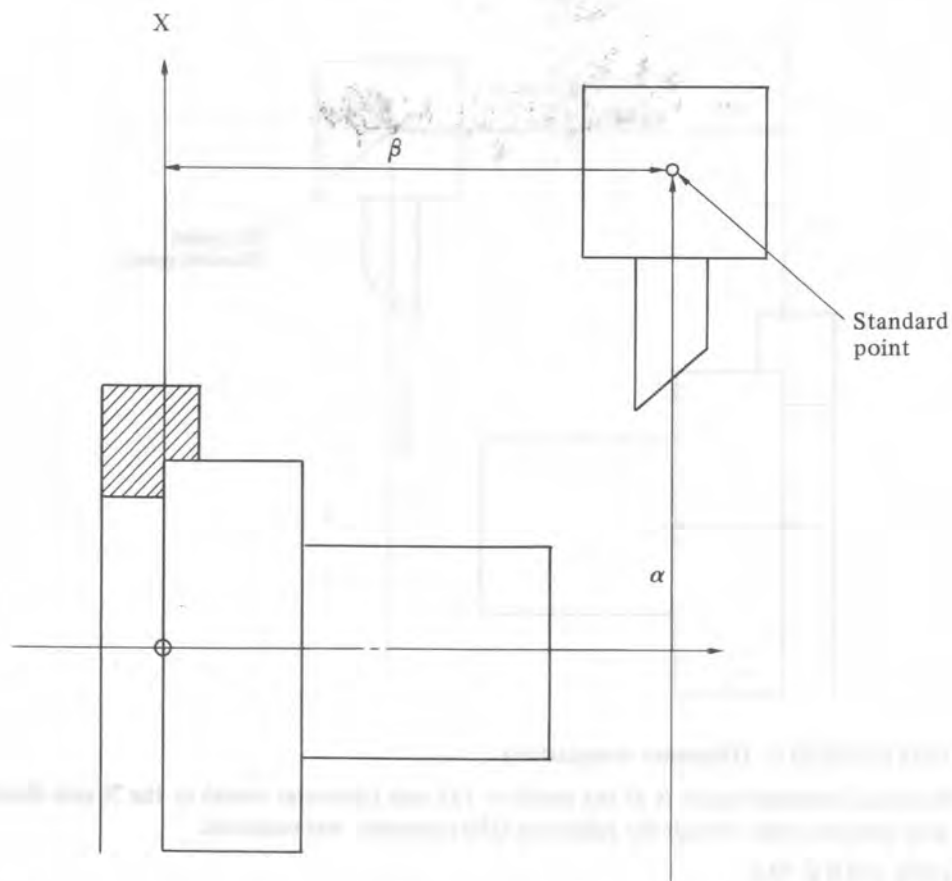
When manual reference point return is performed, the work coordinate system is automatically set according to parameter (parameter APX and APZ (No. 309)). In this case, coordinate values " α " and " β " must be set in parameters 375 and 376 in advance.

If the above parameters are set in advance, the work coordinate system in which the coordinate values of the standard point of the turret or the tool tip is " α " and " β " is automatically set at the completion of manual reference point return. This is equivalent that the following G50 command be specified at the reference point.

G50 X α Z β ;



(When the reference point is the tip of a standard tool.)



(When the standard point is at the center of the turret.)

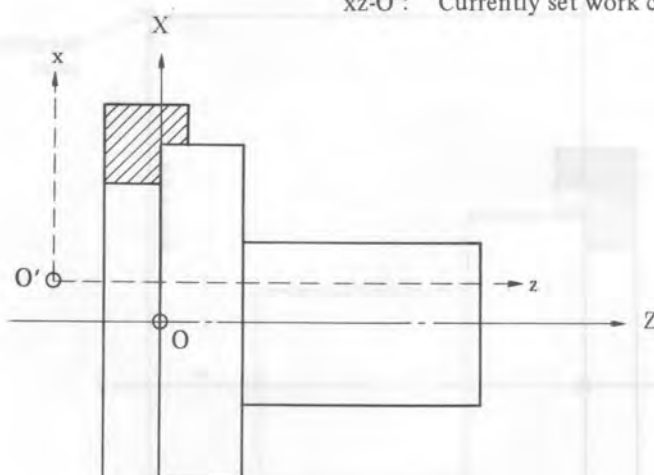
5.10.4 Work coordinate system shifting

If the work coordinate system at with a G50 command or with the automatic coordinate system setting function is different from the coordinate system used in programming, the work coordinate system set in advance can be shifted.

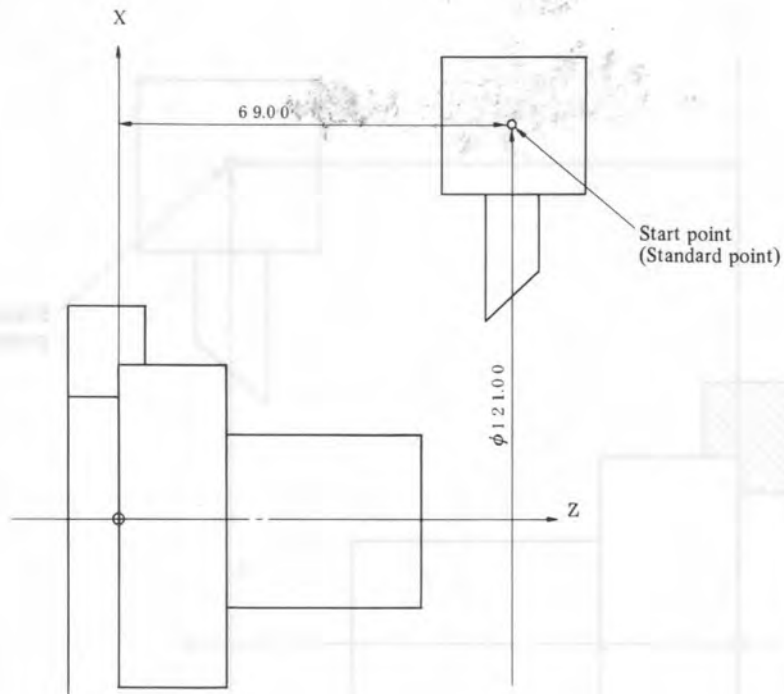
Set the shift amount in offset memory at number 00. This is set in the same way as the tool offset amount.

XZ-O: Coordinate system used in programming

xz-O': Currently set work coordinate system



Set the shift amount, O' to O, in offset memory at number 00.

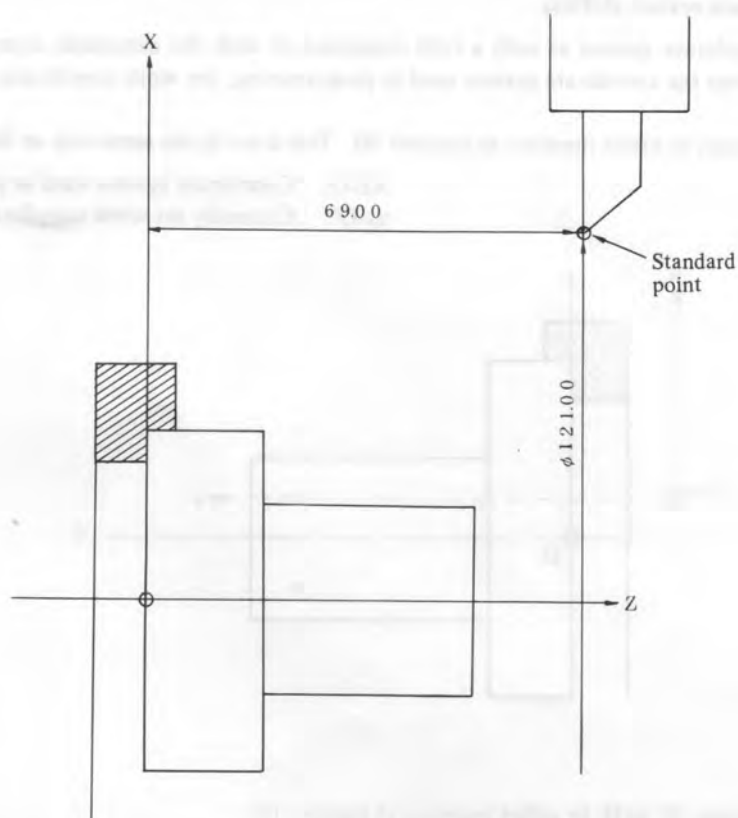


G50X120.0Z70.0; (Diameter designation)

When the actual standard point is at the position 121 mm (diameter value) in the X axis direction and 69 mm in the Z axis direction even though the following G50 command was executed.

G50X 120.0 Z 70.0 ;

the desired work coordinate system can be set by shifting the current coordinate system 1.0 mm in the X axis direction and -1.0 mm in the Z axis direction.



G50 X 120.0 Z 70.0; (Diameter programming)

If the standard point is at the tool tip as shown in the above figure, the coordinate system can be shifted and the desired work coordinate system can be set in the same way. The shift amount is 1.0 mm in the X axis direction and -1.0 mm in the Z axis direction.

When setting the shift amount, the method explained in Section 5.8.5 is simplest and most convenient.

(Note 1) The work coordinate system is shifted immediately after setting the shift amount.

(Note 2) If the G50 command is programmed after the shift amount is set, the shift amount is ignored.

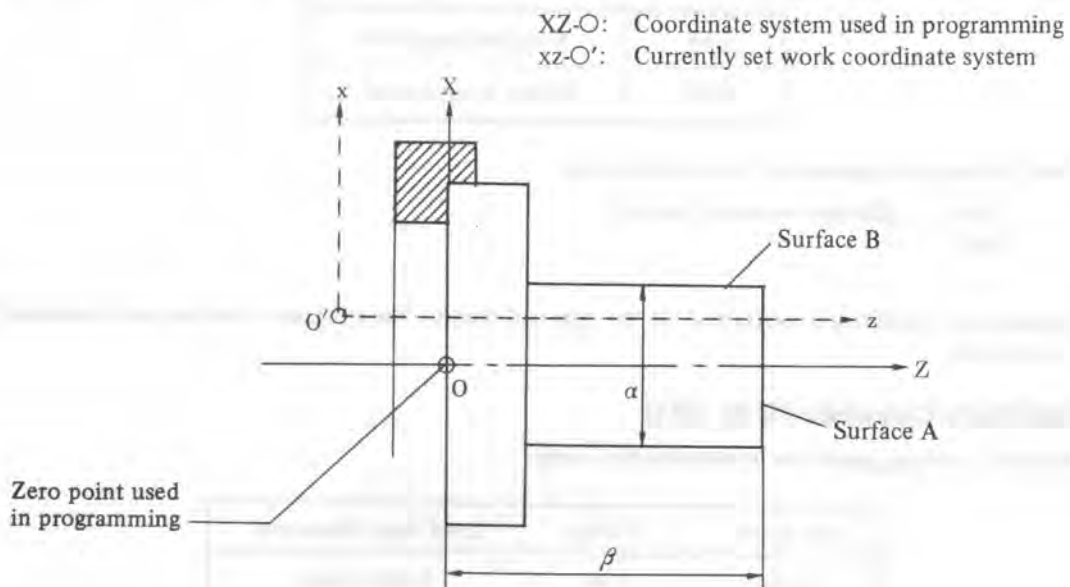
(Example) If the following command is specified, the coordinate values of the current standard point are $X = 100.0$ and $Z = 80.0$ regardless of the shift amount previously set.
G50 X 100.0 Z 80.0 ;

(Note 3) When a shift amount has been set before automatic coordinate system setting is performed, the coordinate system is shifted immediately at the end of manual reference point return.

(Note 4) The shift amount in the X axis direction is specified as a diameter/radius value depending on values used in the program.

5.10.5 Direct measured value input in work coordinate system shifting

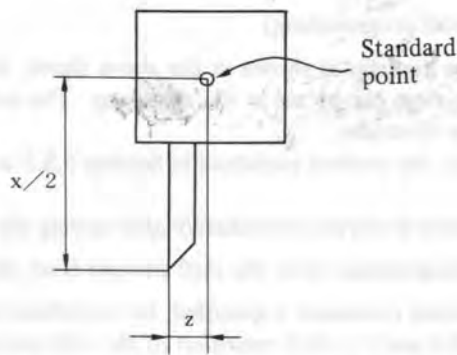
When the work coordinate system set with a G50 command or with automatic coordinate system setting function is different from the coordinate system used in programming, the coordinate system can be shifted by storing the measured distance directly instead of the shift amount as follows:



- (1) Cut the workpiece along surface A manually using the standard tool.
- (2) Release the tool after pressing the tool measuring button on the machine operator's panel and stop spindle rotation.
- (3) Measure distance " β " in the above figure and input the value into the offset memory area with offset number 100 for the Z axis.
- (4) Cut the workpiece along surface B manually.
- (5) Release the tool along the Z axis without moving along the X axis and stop spindle rotation.
- (6) Measure distance " α " and input this value into the offset memory area with offset number 100 for the X axis.

The shift amount, O to O', is automatically set in the offset memory area with offset number 00, and the work coordinate system is shifted immediately.

If the offset amount for the standard tool is zero, the work coordinate system in which coordinate values of the tool tip are $X = 0.0$ and $Z = 0.0$ when the standard tool tip is positioned at the zero point is set.



However, if the tool offset amount is set as shown in the above figure the tool offset function is effective, and the direct measured value input for the work coordinate system shift is used, the work coordinate system in which coordinate values for the standard point are $X = 0.0$ and $Z = 0.0$ when the standard point is positioned at the zero point is set.

(Note) Distance " α " should be a diameter value.

5.10.6 Mirror image for double turrets (G68, G69)

A G code is used to generate mirror image to the X axis.

G code	Meaning
G68	X mirror image ON
G69	Mirror image cancel

The following is programmed for double turrets.

```
G50U___(Distance between turrets);
G68;
}
```

Subsequent machining is performed on the opposed turret. The program is written as if machined by the turret on this side.

5.11 Inch/Metric Conversion (G20, G21)

Either inch or metric input can be selected by G code.

Unit system	G code	Least input increment
Inches	G20	0.0001 inch
mm	G21	0.001 mm

These G codes must be specified in a single block before the work coordinate system setting block at the start of the program.

The following change according to these G codes.

1. Feedrate command F or E code
2. Display of positions
3. Offset value
4. Unit of scale on the manual pulse generator
5. Move distance during incremental feed.

(Note 1) When the power is turned on, the G code in effect before the power was turned off is still effective.

(Note 2) Either inch/or metric input can be selected from the MDI. The selected status is changed by G20 or G21 command.

(Note 3) G20 and G21 must not be changed in a program.

(Note 4) When the unit system of a machine and the one used in programming are different, the maximum switching error is half of the least command increment. This error is not accumulated.

5.12 Feedrate Command Switching (G98, G99)

The following G codes specify either feed per minute or feed per revolution.

G code	Feedrate unit
G98	Feed per minute
G99	Feed per revolution

See Section 4.2 for detail.

5.13 Constant Surface Speed Control (G96, G97)

The following G codes specify whether or not constant surface speed control is in effect.

G code	CSSC	Meaning
G96	ON	Constant surface speed control is performed until G97 is specified.
G97	OFF	Constant surface speed control is not performed until G96 is specified.

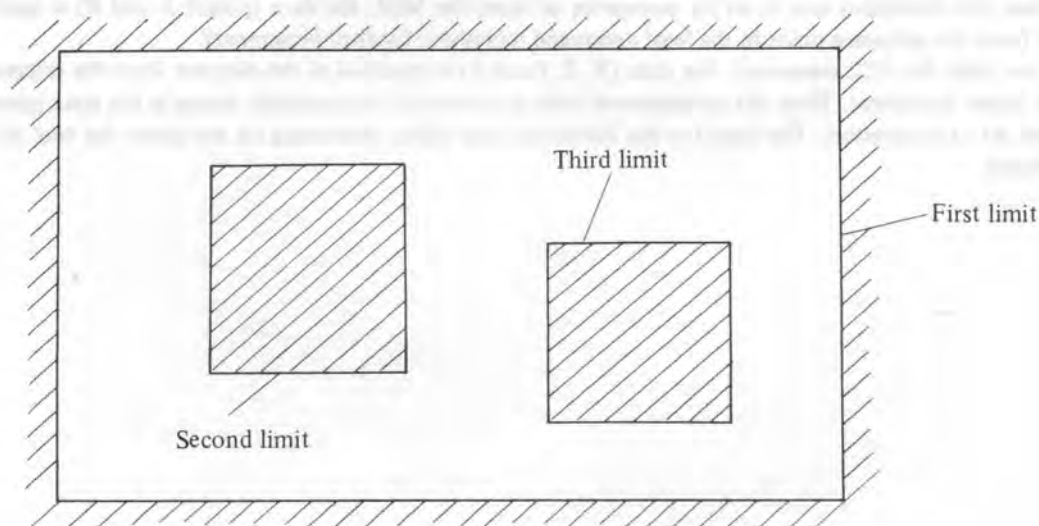
CSSC: Constant Surface Speed Control

See Section 8.2 for detail.

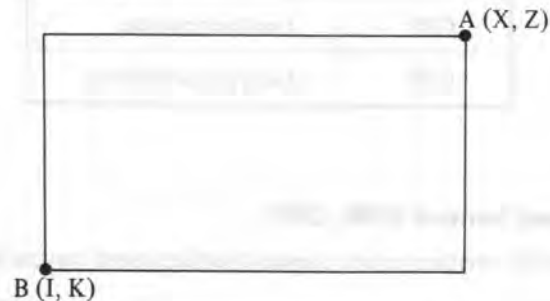
(Note 1) When turning the power on, G97 (Constant surface speed control off) is set.

5.14 Stored Stroke Limit (G22, G23)

The area in which a tool may not travel (forbidden area) can be specified in the following three ways. (Tools cannot enter the shaded areas)

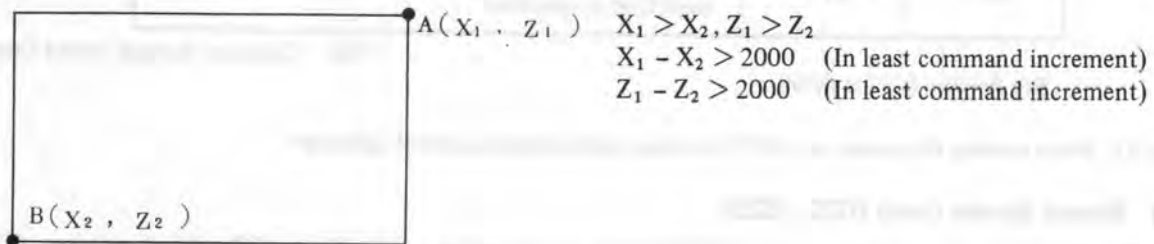


- First limit: Parameter set boundary. The area outside the set area is the forbidden area. This is normally not changed once the machine tool builder sets this area. This area is usually set at the maximum stroke and is called the stored stroke limit. This is equivalent to a conventional software limit.
- Second limit: Boundary is set by a parameter or from MDI. The area inside the set area is the forbidden area. A G22 command is used to inhibit the tool from entering the forbidden area and a G23 command is used to allow the tool to enter the forbidden area as well as a third limit area.
- Third limit: Boundary is set by a parameter, from MDI, or by a program. Inside or outside the set area is the forbidden area. Parameter (RWL) is used to select inside or outside. A G22 command is used to inhibit a tool from entering the forbidden area and a G23 command is used to allow a tool to enter the forbidden area as well as a second limit area.
- The command below defines or changes the third limit.
- G22X—Z—I—K—;



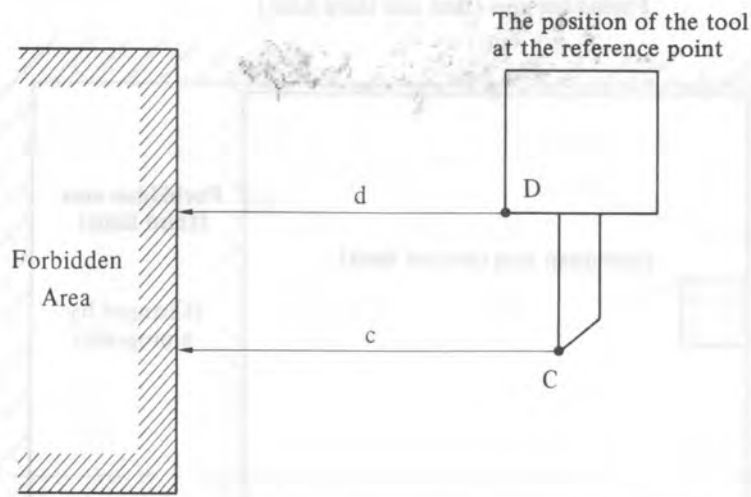
$X > I, Z > K$
 $X - I > 2000$ (In least command increment)
 $Z - K > 2000$ (In least command increment)

When set by parameter or from MDI, points A and B in the figure below must be set.

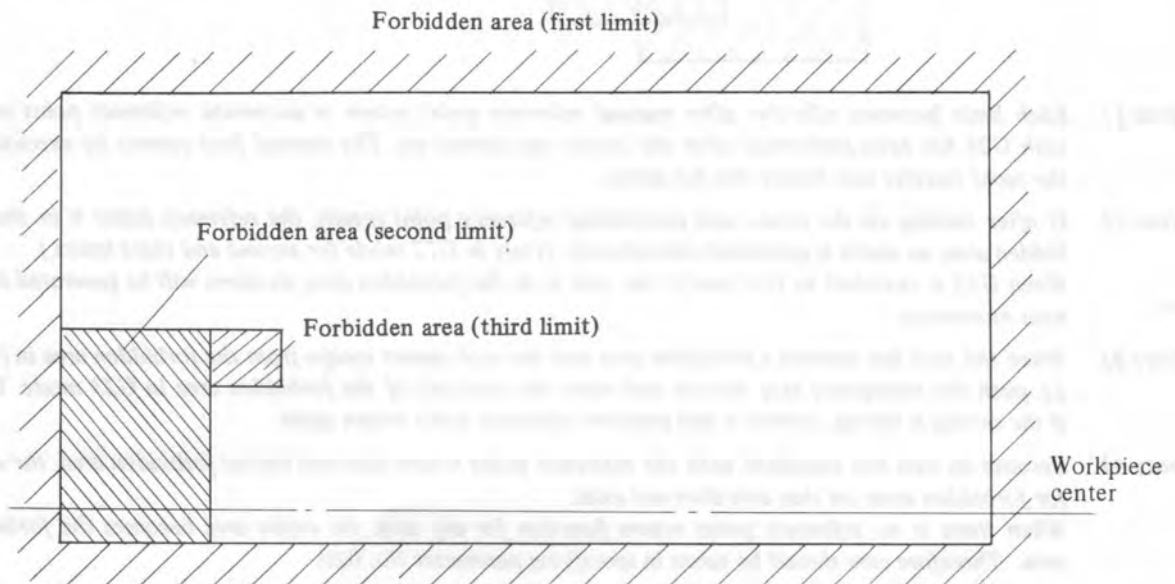


When the forbidden area is set by parameter or from the MDI, the data (points A and B) is specified as the distance from the reference point in the least command increment (output increment).

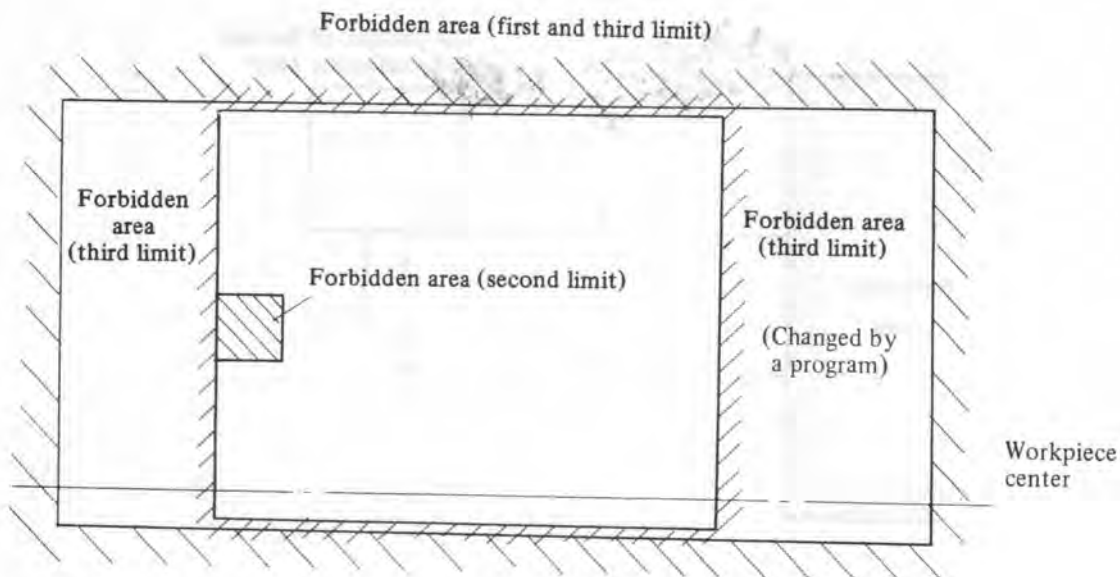
If set with the G22 command, the data (X, Z, I and K) is specified as the distance from the reference point in the least input increment. Then the programmed data is converted to numerical values in the least command increment and set as parameters. The data for the forbidden area varies depending on the place the tool is checked, as shown below.



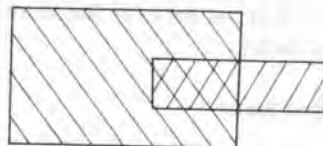
When point C (tool nose) is checked, the distance c is set as data for the stored stroke limit function. However, when point D (tool post) is checked, distance d must be set.



The chuck can be protected by setting as shown above. It is also possible to change the forbidden area according to the movement of the tail stock and protect the chuck as shown below.



Areas can overlap.



- (Note 1) Each limit becomes effective after manual reference point return or automatic reference point return with G28 has been performed after the power was turned on. The manual feed cannot be executed at the rapid traverse rate before this for safety.
- (Note 2) If after turning on the power and performing reference point return, the reference point is in the forbidden area, an alarm is generated immediately. (Only in G22 mode for second and third limits.) When G23 is switched to G22 and if the tool is in the forbidden area, an alarm will be generated in the next movement.
- (Note 3) When the tool has entered a forbidden area and the tool cannot escape from the forbidden area in (Note 2), push the emergency stop button and move the tool out of the forbidden area in G23 mode. Then, if the setting is wrong, correct it and perform reference point return again.
- (Note 4) Because an axis not equipped with the reference point return function has no forbidden area, the alarm for forbidden areas on that axis does not exist.
When there is no reference point return function for any axis, the entire area becomes the forbidden area. Therefore care should be taken in specifying parameter No. 020.
- (Note 5) If when setting a forbidden area, the two points to be set are the same, the following forbidden areas are defined:
First limit: No forbidden area is defined
Second limit: The entire area is the forbidden area
Third limit: The entire area is the forbidden area when "inside" is specified.
No forbidden area is defined when "outside" is set.
- (Note 6) Unnecessary limit should be set beyond the machine stroke.
- (Note 7) When a tool enters a forbidden area and an alarm is generated, the tool can be moved in reverse of the direction that it travelled into the forbidden area.
- (Note 8) When setting the area, even if the size of the coordinate value of the two points is specified incorrectly, a rectangular area with the two points as the two apexes will be set as the area.
- (Note 9) G22 ____; and G23; should be specified in a single block respectively.
- (Note 10) When movement along an axis stops after entering a forbidden area, it will stop after entering with a margin of moving value for 8 m sec. max. from the boundary line.
- (Note 11) The boundaries of the 2nd and 3rd limit can be set by a parameter or from the MDI.

(Note 12) When movement along an axis enters an area in common with forbidden areas set by the 1st, 2nd and 3rd limit, the tool cannot escape from the forbidden area. Execute the operation mentioned in (Note 3).

5.15 Skip Function (G31)

X(U) and Z(W) following G31 specifies linear interpolation as in G01. Input of the skip signal during execution of this command interrupts the rest of the block and executes the next block.

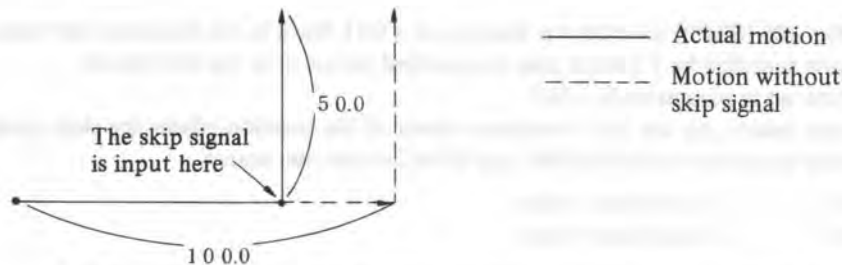
G31 is a one-shot command, effective only in the block concerned.

The motion after input of the skip signal depends on whether the next block contains an incremental or absolute command.

(1) When the next block contains an incremental command

The motion of the next block is incremental from the interrupted position.

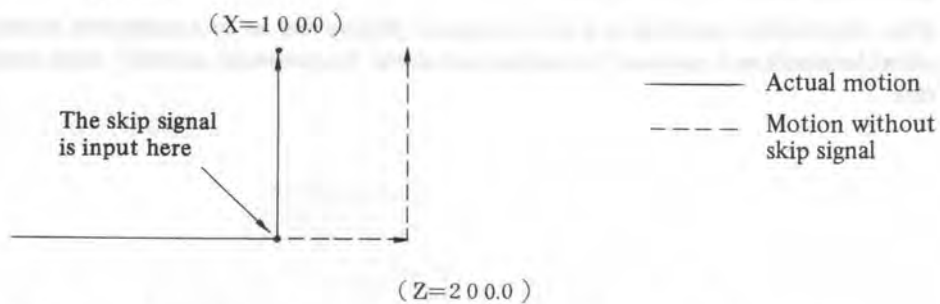
Example: G31W100.0;
U50.0;



(2) When the next block contains an absolute command specifying only one axis.

The tool moves along the specified axis to the specified position; the position on the other axis remains the same as when the skip signal was input.

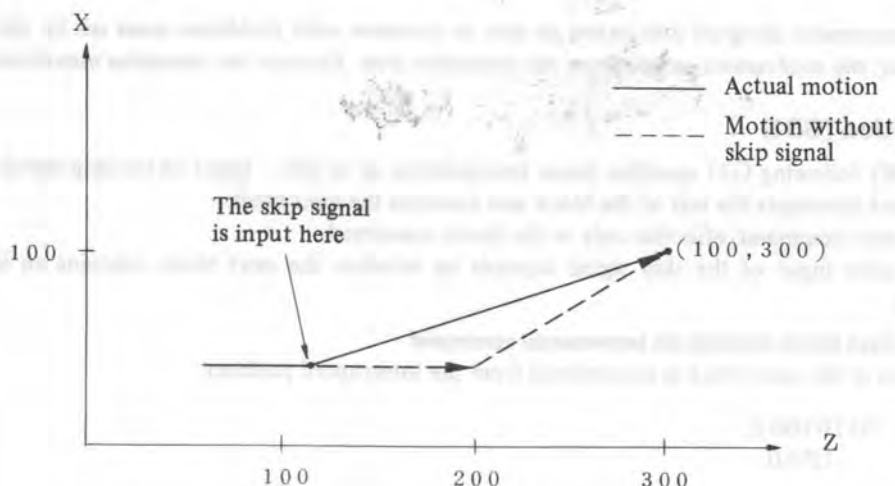
Example: G31Z200.0;
X100.0;



(3) When the next block contains an absolute command specifying two axes

The tool moves to the specified position regardless of input of the skip signal.

Example: G31Z200.0;
X100.0Z300.0;



Parameter No. 306 (SKPF) specifies the feedrate of a G31 block in the following two ways.

- (1) Feedrate specified by F (which may be specified before or in the G31 block)
- (2) Feedrate set in parameter No. 342

The user macro can use the coordinate values of the position where the skip signal was issued, since they are stored in system variables #5061 and #5062 of the user macro.

#5061 X coordinate value

#5062 Z coordinate value

The skip function can be used when the movement amount is not clear, as for:

- (a) Standard-size feed by grinder
- (b) Measurement of a tool by carrying it to a touch sensor

(Note 1) The G31 command generates alarm No. 035 when the tool nose R compensation is used. Before specifying the G31 command, cancel the tool nose R compensation using G40.

(Note 2) When the feedrate in a G31 command follows that set by a parameter, the feedrate during a dry run also follows that set by the parameter.

(Note 3) When the feedrate specified in a G31 command follows that set by a parameter, no automatic acceleration/deceleration is executed to increase automatic measurement accuracy when using the skip function.

6. COMPENSATION FUNCTION

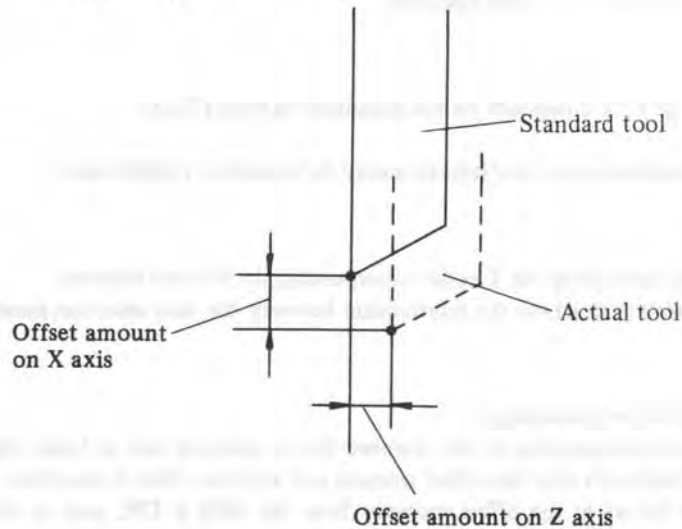
When a tool different from the imagined tool is used in programming, or tool nose radius compensation is performed by the NC, the compensation function is used. This function includes tool offset and tool nose radius compensation.

6.1 Tool Offset

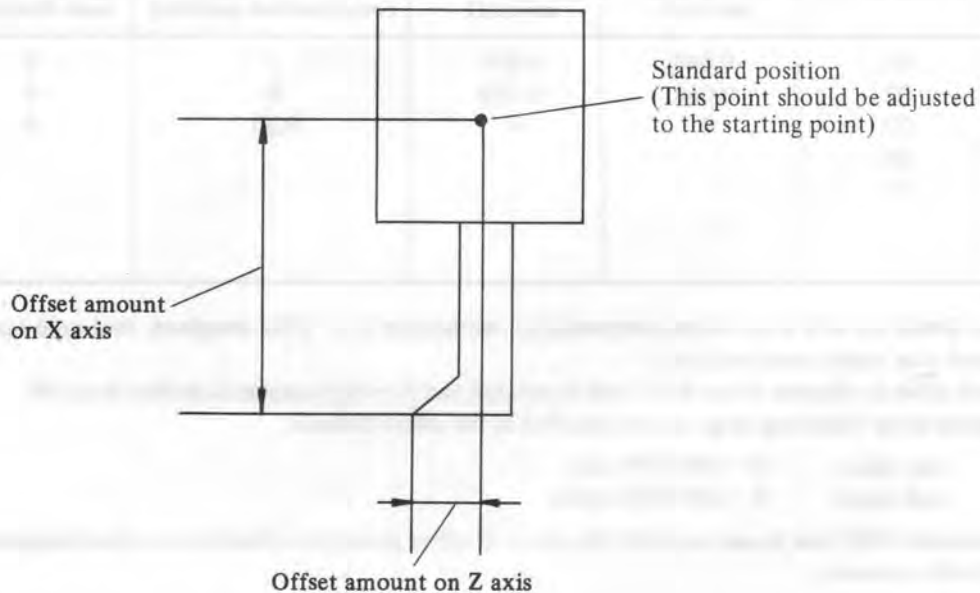
In this unit, there is no G code to specify tool offset. The tool offset is specified by T code.

6.1.1 Basic tool offset

Tool offset is used to compensate for the difference when the tool actually used differs from the imagined tool used in programming (usually, standard tool).



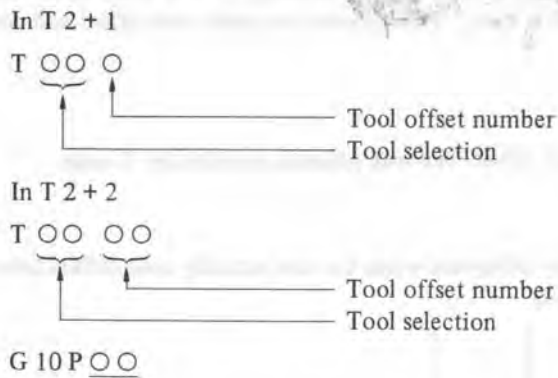
Programming is done so that the tip of the standard tool moves over the desired path, and the workpiece is set to match the standard tool starting point in the program. However, the actual tool used seldom matches the standard tool exactly. Rather than attempting to set the tool tip position, it is more convenient to measure the distance between the standard and actual tool tip positions. This distance is used as the offset amount.



The standard position can be determined in two ways: the point that is the starting position when the reference point return is completed, or a certain point when the reference point return is completed.

6.1.2 T code for tool offset

T codes have the following meaning.



The selection of T2 + 1 or T2 + 2 depends on the parameter setting (T2D).

(Note 1) Depending on the machine tool, tool selection may be set with a 1-digit code.

6.1.3 Tool selection

Tool selection is made by specifying the T code corresponding to the tool number.

Refer to the manufacturer's manual for the relationship between the tool selection number and the tool.

6.1.4 Tool offset number

The tool offset number has two meanings:

It specifies the offset distance corresponding to the number that is selected and to begin the offset function. A tool offset number of 0 or 00 indicates that the offset amount is 0 and the offset is cancelled.

An offset distance must be set in the offset memory from the MDI & DPL unit or the tape reader corresponding to the offset number. (See Section 5.26 in Chapter IV.)

There are three types of compensation corresponding to the specified offset number.

Offset by compensation of X and Z is called Tool Offset, and compensation of R is called Tool Nose Radius Compensation.

Offset number	OFX (X axis offset amount)	OFZ (Z axis offset amount)	OFR (Tool nose radius compensation amount)	OFT (Imaginary tool nose direction)
01	0.040	0.020	0	0
02	0.060	0.030	0	0
03	0	0	0.20	0
04	.	.	.	.
05	.	.	.	.
.	.	.	.	.
.	.	.	.	.

For details on tool nose radius compensation, see Section 6.2. (The imaginary tool nose direction is also used in tool nose radius compensation.)

Tool offset is effective when the T code is selected and its offset number is neither 0 nor 00.

Values in the following range can be specified as the offset distance:

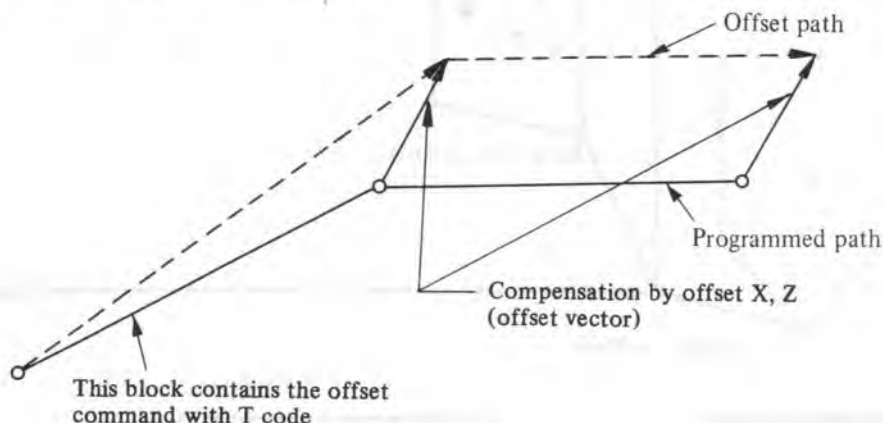
mm input 0 ~ ±999.999 mm
inch input 0 ~ ±99.9999 inches

Parameter ORC can be set to switch the axis tool offset amount to diameter or radius designation for the X axis tool offset amount.

(Note 1) When mm input is 0.01 mm, the input unit of offset amount is 0.001 mm.

6.1.5 Offset

The tool path is offset by the X and Z values for the programmed path. The offset distance corresponding to the number specified by the T code is added to or subtracted from the end position of each programmed block.



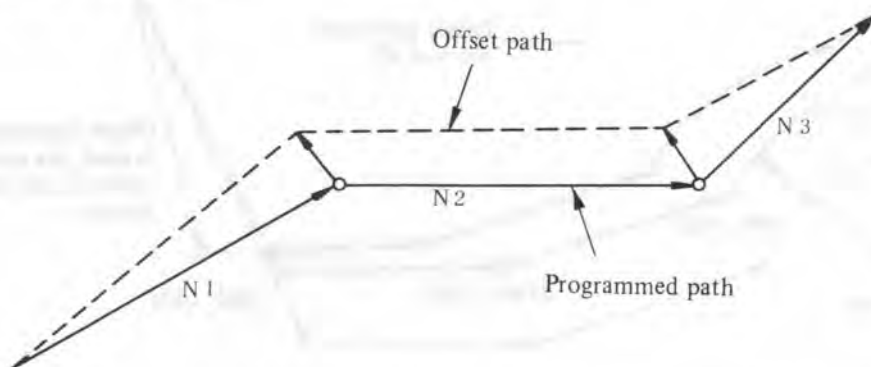
6.1.6 Offset vector

In the diagram above, the vector with offset X, Z is called the offset vector. Compensation is the same as the offset vector.

6.1.7 Offset cancel

Offset is cancelled when T code offset number 0 or 00 is selected. At the end of the cancelled block, the offset vector becomes zero.

```
N1 U 50.0 W 100.0 T0202 ;
N2 W 100.0 ;
N3 U 50.0 W 50.0 T0200 ;
```



(An offset value is assumed to have been entered in the 02 offset memory OFX and OFZ, respectively.)

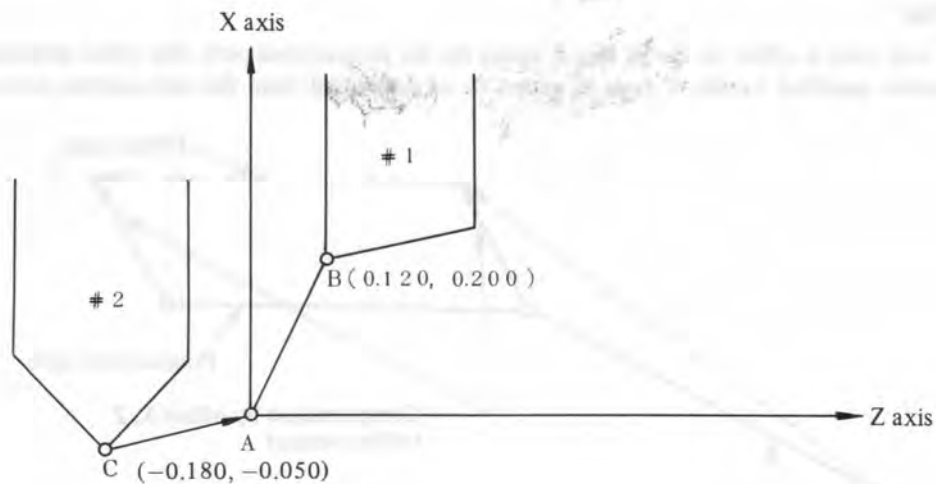
When the power is first turned on, and the reset key on the MDI unit is pushed or the rest signal is input to the NC from the machine tool, the offset is cancelled and the offset number becomes 0 or 00.

Parameter TOC can be set so that the offset will not be cancelled by pressing the reset key or by reset input.

(Note) When reference point return is performed manually or with G28 or G30, the offset vector for the axis reaching the reference point is temporarily cancelled. The offset vector is restored with the next program command. But when parameter TLCC is set, the offset vector isn't restored until the next T code is specified.

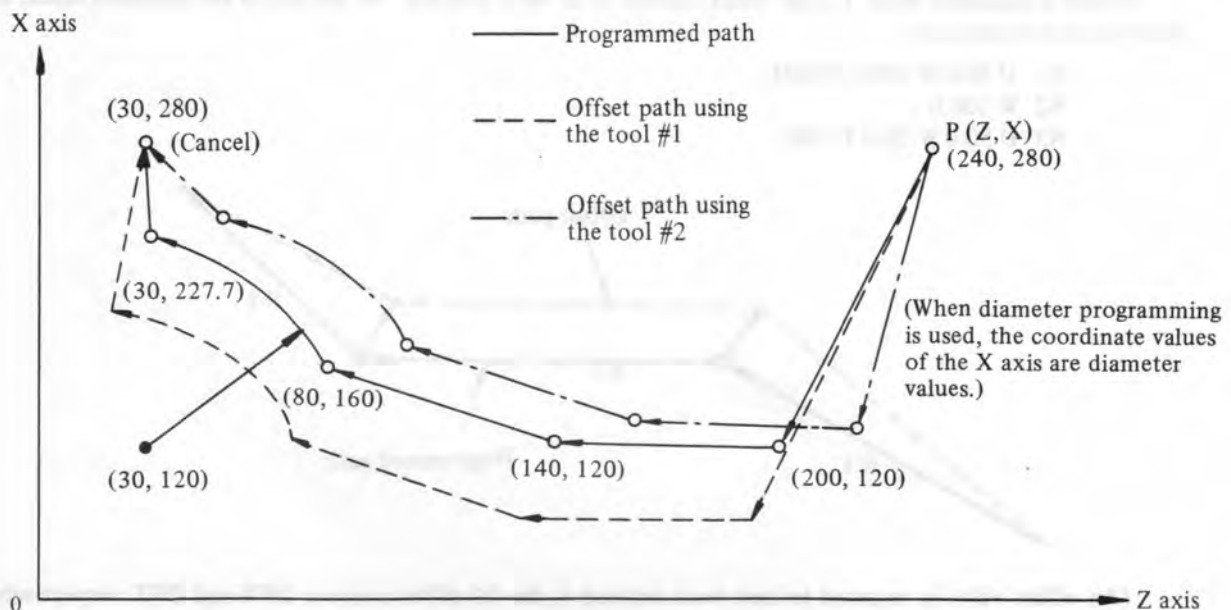
6.1.8 Program example

Tool tip coordinate value (Z, X)	Tool number
Tool #1 B (0.120, 0.200)	01
Tool #2 C (-0.180, -0.050)	02



A: Programmed tool nose
 B: Actual tool nose (#1)
 C: Actual tool nose (#2)

	Tool number	
	01	02
X	-0.200	+0.050
Z	-0.120	+0.180
R	0	0



(Program example 1)

```
G50 X 280.0 Z 240.0 ;
G00 X 120.0 Z 200.0 T 11 ;
G01 Z 140.0 F 30 ;
      X 160.0 Z 80.0 ;
G03 X 227.7 Z 30.0 I-20.0 K-50.0 ;
G00 X 280.0 T 10 ;
```

The path of the tool nose of tool #1 coincides with the programmed path in this program.

(Program example 2)

The path of the tool nose of tool #2 will coincide with the programmed path when the following changes are made in program example 1.

T11 → T22
T10 → T20

6.1.9 Only T code

When only a T code is specified in a block, the tool is moved by the offset value without a move command. The movement is performed at rapid traverse rate in the G00 mode. It is performed at feedrate in other modes.

When a T code with offset number 0 or 00 is specified by itself, movement is performed to cancel the offset.

(Note 1) *G50X (x) Z (z) T___ ;*

Tool is not moved.

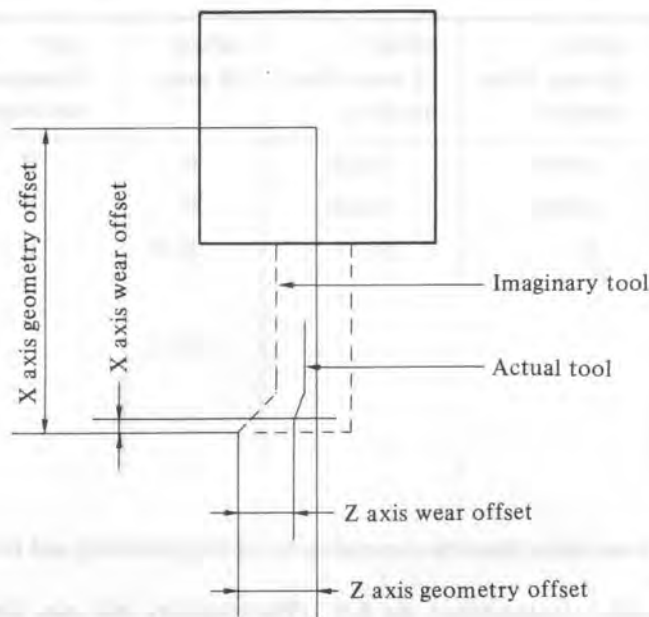
The coordinate system in which the coordinate value of the tool position is (x, z) is set. The tool position is obtained by subtracting the offset value corresponding to the offset number specified in the T code.

(Note 2) *When the offset value corresponding with the offset number being used in the cycle operation is changed from the MDI, the new offset value cannot be used until the T code of the offset number is specified again. But in certain cases, the new offset value is used from the next block by setting the parameter TLCC.*

(Note 3) *The block: G04T___; executes a tool change and effects tool offset for the next block which contains a move command when parameter TLCC = 1. The block: G04T___; executes a tool offset but does not effect tool offset in the following block when parameter TLCC = 0.*

6.2 Tool Geometry Offset and Tool Wear Offset

Optionally, it is possible to divide the tool offset to the tool geometry offset for compensating the tool shape or tool mounting position and the tool wear offset for compensating the tool nose wear.



6.2.1 T code for tool offset

The T code has the following meaning:

T 0000

Geometry offset number & wear offset number

Tool selection

(Note 1) *The geometry offset number can be made common to the tool selection by the setting of the parameter.*

(Note 2) *Maximum 32 pairs can be used for offsets. Maximum 64 pairs option can not be used.*

6.2.2 Tool offset number

Designation of the tool offset number means the selection of the corresponding offset amount and the start of offset.

When the tool offset number is 0, the wear offset amount is zero, and the wear offset is cancelled.

(Generally, the geometry offset is not cancelled. At the next tool selection, the coordinate system is shifted by the difference.)

Set the offset amount corresponding to the offset number in advance in the offset memory by using the MDI unit or with the G10 command.

The compensation amounts corresponding to the designated offset number include the values for the X and Z axes and tool nose radius values. The offset with the X and Z compensation amounts is called the tool offset. Compensation by tool nose radius is called the tool nose radius compensation.

Geometry offset

Geometry offset number	OFGX (X geometry offset number)	OFGZ (Z geometry offset number)	OFGR (R geometry)	OFT (Imaginary tool number)
G 01	10.040	50.020	0	0
G 02	20.060	20.030	0	0
G 03	0	0	0.20	0
G 04	0	0	0	0
.	.	.	.	.
.	.	.	.	.

Wear offset

Wear offset number	OFWX (X wear offset number)	OFWZ (Z wear offset number)	OFWR (R wear)	OFT (Imaginary tool number)
W01	0.040	0.020	0	0
W02	0.060	0.030	0	0
W03	0	0	0.20	0
W04	.	.	.	.
W05	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

The imaginary tool nose radius direction is common to the tool geometry and tool wear offsets and may be designated for either offset.

For the tool nose radius compensation, see 6.3. (The imaginary tool nose direction is also used in tool nose radius compensation.)

Tool offset is valid when the T code is selected and the offset number is not 0. Wear offset is cancelled when the T code is selected and the offset number is 0.

The following values can be set as the offset amount:

Metric input: 0 to ±999.999 mm

Inch input: 0 to ±99.9999 inch

As for X-axis offset amount, diameter or radius designation can be selected with the parameter (ORG). Further, the maximum (WOMAX) and the incremental offset input value (WIMAX) can be limited by the parameter.

(Note) For the incremental offset input, see IV, 5.12.2.

6.2.3 Offset

When the wear offset is performed, the programmed path is offset by the X and Z offset amounts. Namely, the offset amount corresponding to the number designated with the T code is added to or subtracted from the end point of each block.

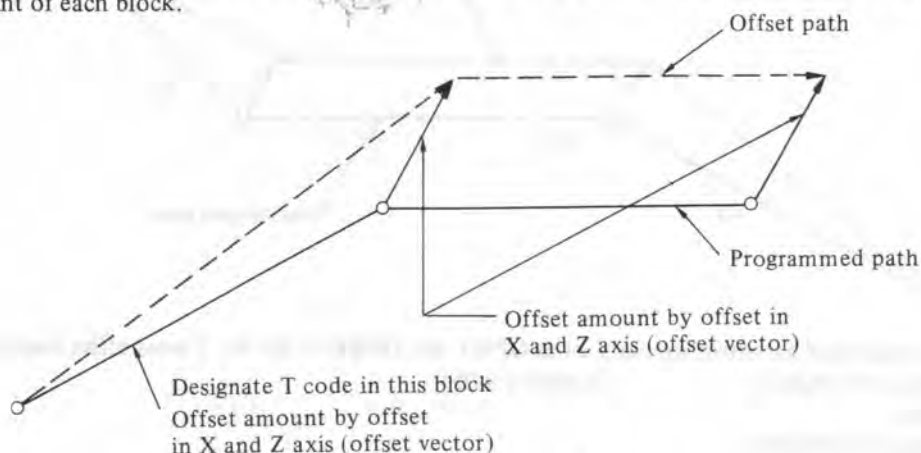


Fig. 6.2.3 (a) Wear offset

With the geometry offset, the coordinate system is shifted by the X and Z offset amounts. Namely, the offset amount corresponding to the number designated with the code is added to or subtracted from the current position.

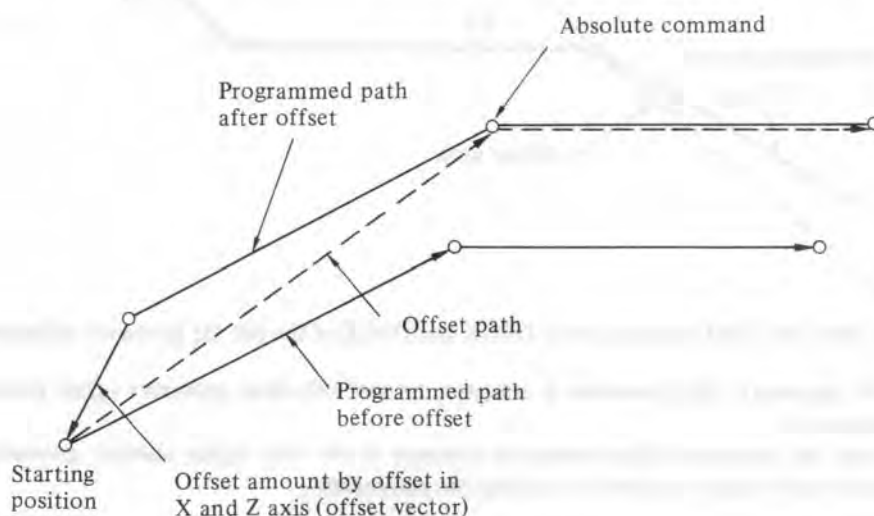


Fig. 6.2.3 (b) Geometry offset

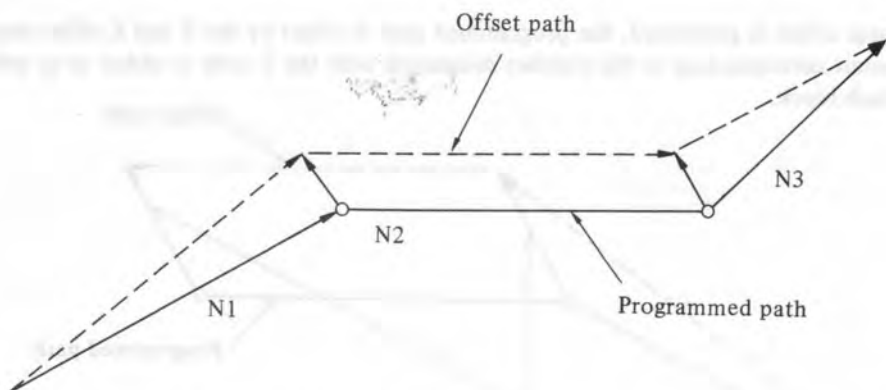
6.2.4 Offset vector

The vector constituted by the X and Z offset amounts, as shown in Fig. 6.2.3 (a), is called the offset vector. Compensation by the X and Z offset amounts is called the compensation by the offset vector.

6.2.5 Offset cancel

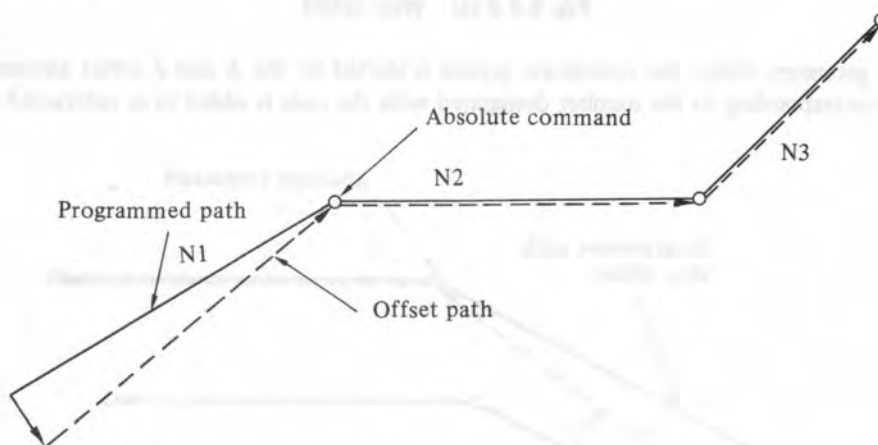
Offset is cancelled when the T code of offset number 0 is selected. At the end of the block in which offset is cancelled, the offset vector is zero.

```
N1 U50.0W100.0T0202 ;      Wear offset
N2 W100.0 ;
N3 U50.0W50.0T0200 ;
```



(Assume that there are offset amounts set at OFWX and OFWZ of the No. 2 wear offset memory.)

N1 X50.Z100.T0202 ; Geometry offset
 N2 Z200 ;
 N3 X200.Z250.T0200 ;



(Assume that there are offset amounts set at OFGX and OFGZ of the No. 02 geometry offset memory.)

- (Note 1) When the geometry offset number is common to tool selection, geometry offset is cancelled with tool selection 0.
- (Note 2) Even when the geometry offset number is common to the wear offset number, geometry offset can be cancelled with offset number 0 by setting the parameter.

6.3 Tool Nose Radius Compensation

It is difficult to produce the compensation necessary to form accurate parts when using only the tool offset function due to tool nose roundness. The tool nose radius compensation function compensates automatically for the above errors.

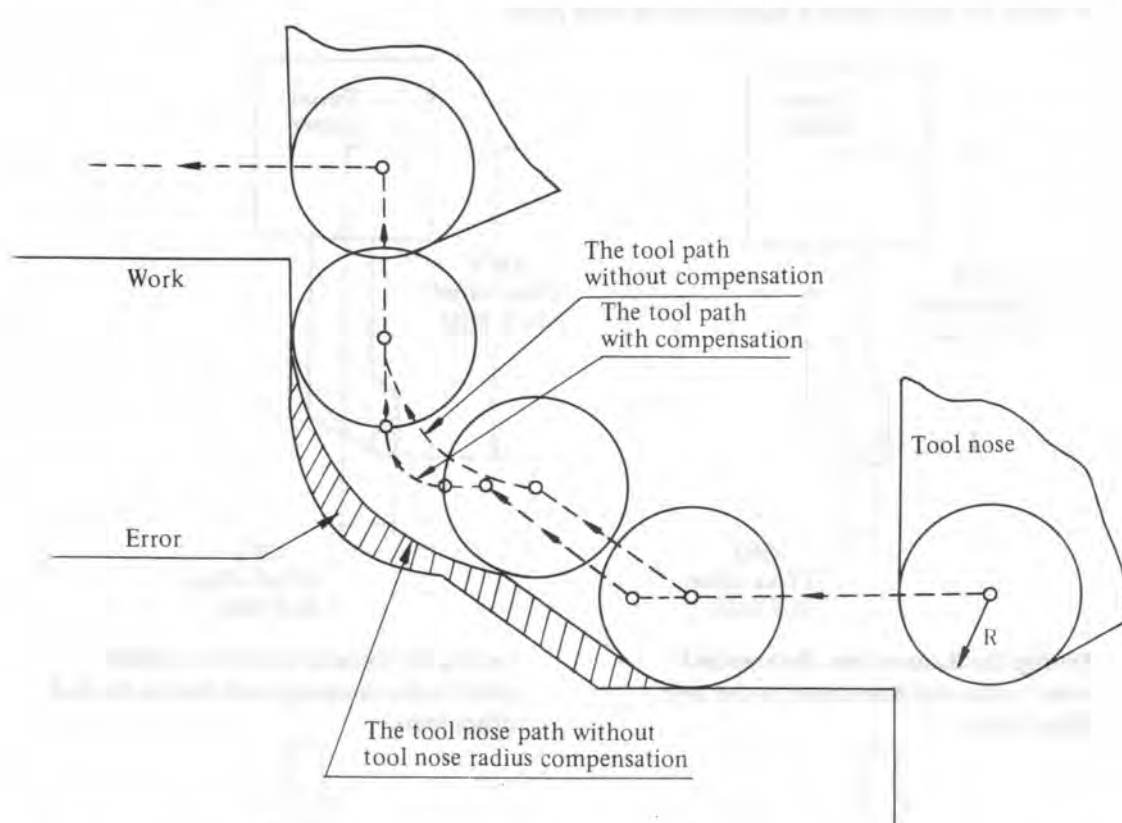
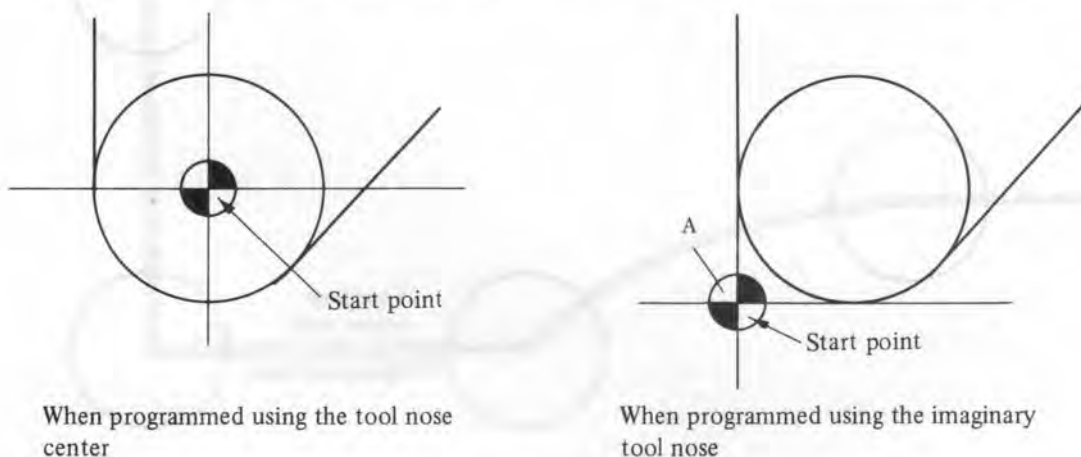


Fig. 6.3 Tool Path with Tool Nose Radius Compensation

6.3.1 Imaginary tool nose

The tool nose at position A in the following figure does not actually exist. The imaginary tool nose is required because it is usually more difficult to set the actual tool nose center to the start point than the imaginary tool nose. (Note) Also when imaginary tool nose is used, the tool nose radius need not be considered in programming.

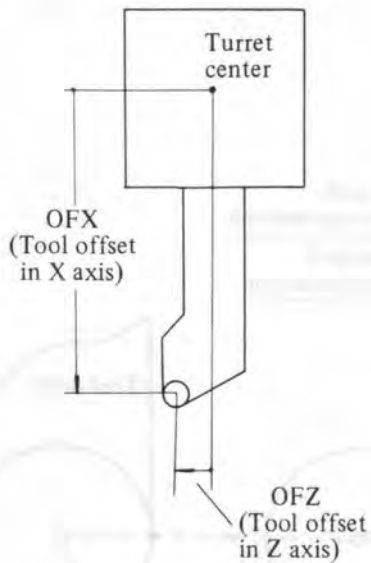
The position relationship when the tool is set to the start point is shown in the following figure.



(Note) In a machine with reference points, a standard point like the turret center can be placed over the start point. The distance from this standard point to the nose radius center or the imaginary tool nose is set as the tool offset value.

Setting the distance from the standard point to the tool nose radius center as the offset value is the same as placing the tool nose radius center over the start point, while setting the distance from the standard point to the imaginary tool nose is the same as placing the imaginary tool nose over the standard point. To set the offset value, it is usually easier to measure the distance from the standard point to the imaginary tool nose than from the standard point to the tool nose radius center.

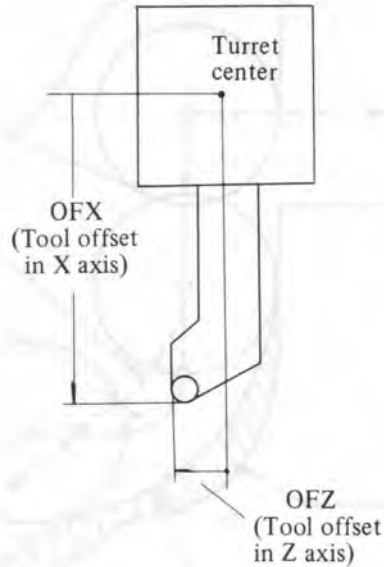
- When the turret center is placed over the start point.



Setting the distance from the standard point to the tool nose center as the tool offset value.



The start point is placed over the tool nose center.



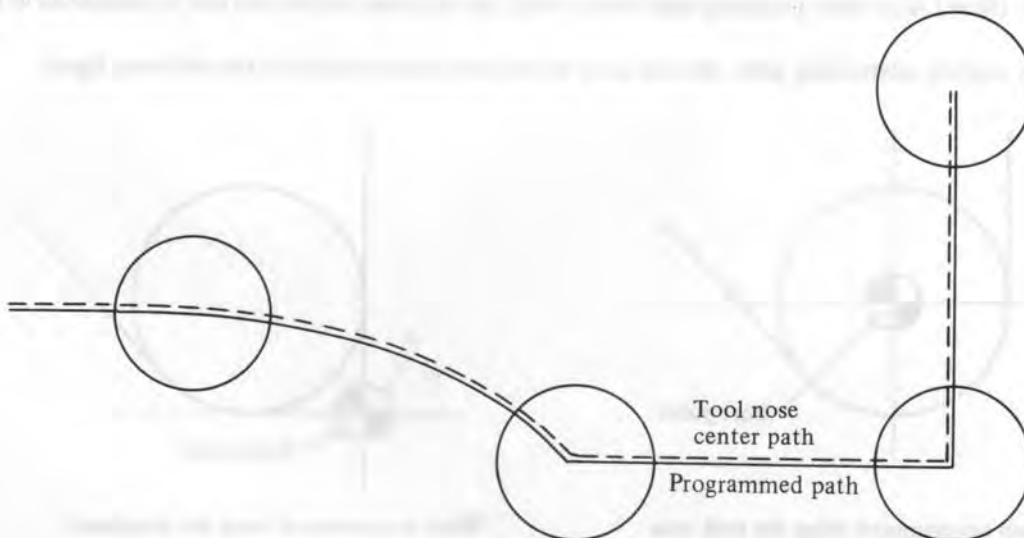
Setting the distance from the standard point to the imaginary tool nose as the tool offset value.



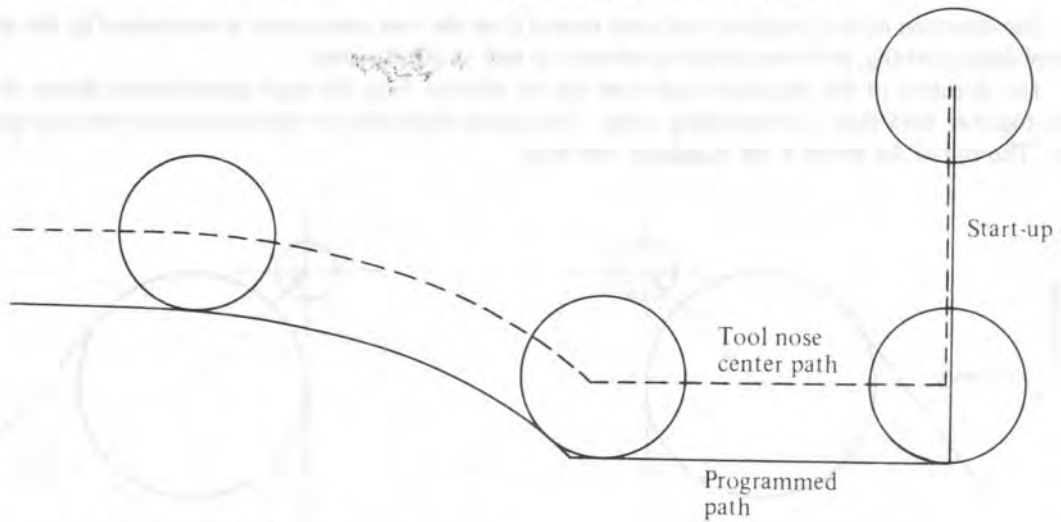
The start point is placed over the imaginary tool nose.

(i) Programming using the tool nose center

Unless tool nose radius compensation is performed the tool nose center path is the same as the programmed path.

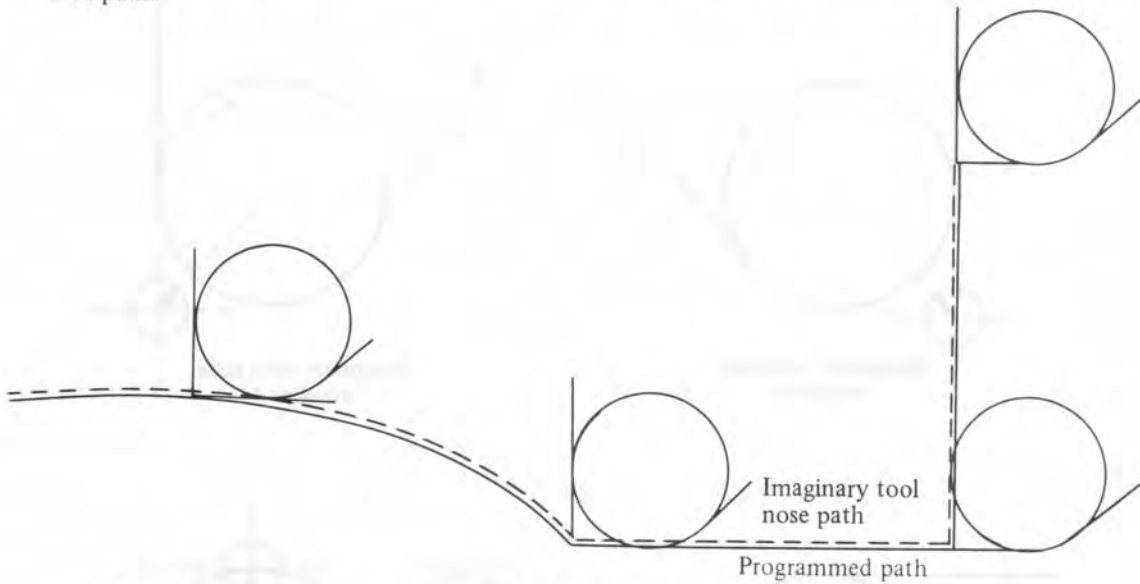


If tool nose radius compensation is used, accurate cutting will be performed.

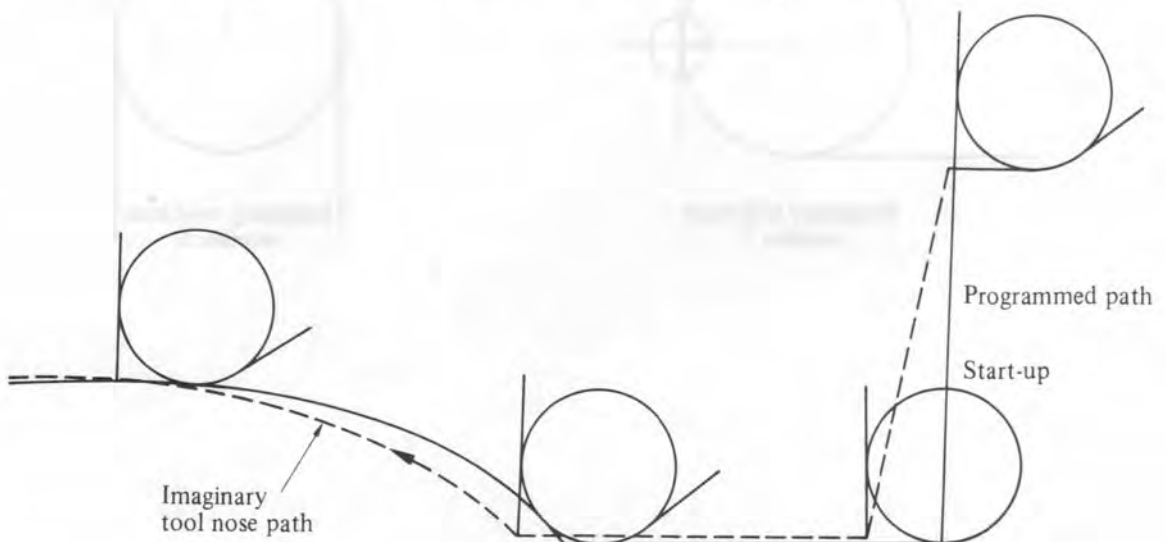


(ii) Programming using imaginary tool nose

Unless tool nose radius compensation is used, the imaginary tool nose path is the same as the programmed path.



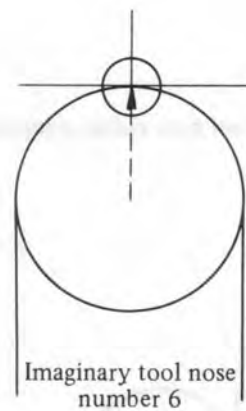
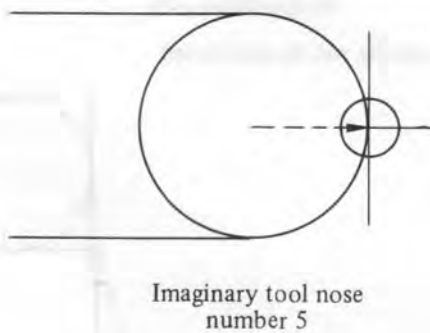
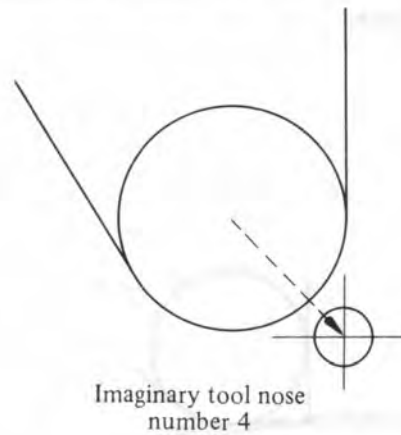
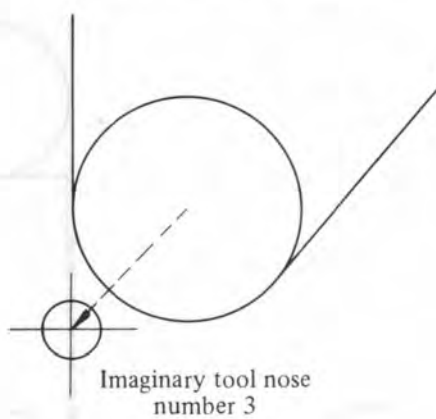
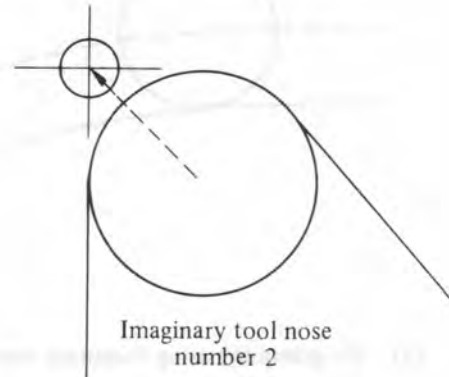
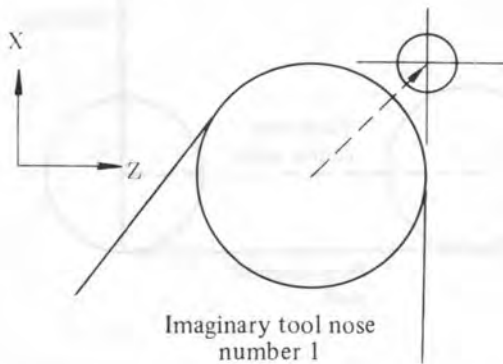
If tool nose radius compensation is used, accurate cutting will be performed.

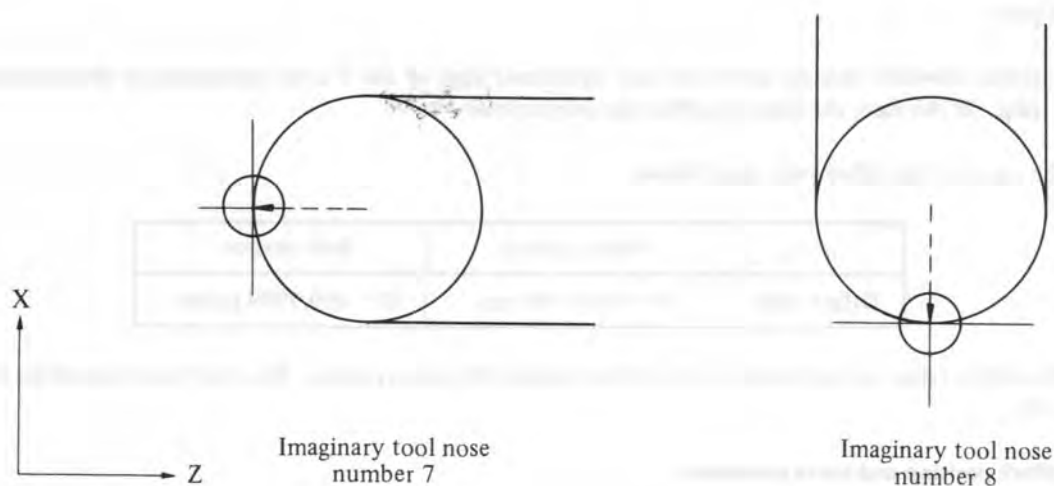


6.3.2 Direction of imaginary tool nose

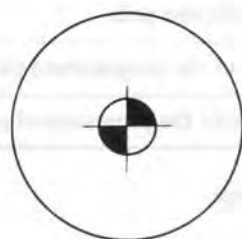
The direction of the imaginary tool nose viewed from the tool nose center is determined by the direction of the tool during cutting, so it must be set in advance as well as offset values.

The direction of the imaginary tool nose can be selected from the eight specifications shown in the figure below together with their corresponding codes. This figure illustrates the relation between the tool and the start point. The end of the arrow is the imaginary tool nose.



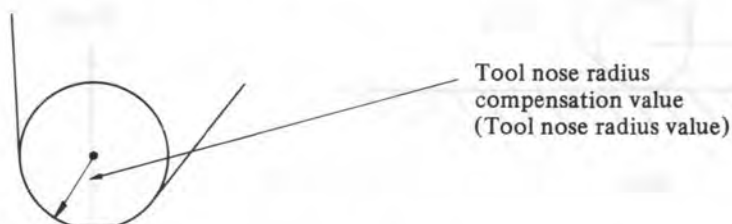


Imaginary tool nose numbers 0 and 9 are used when the tool nose center coincides with the start point. Set imaginary tool nose number to address OFT for each offset number.



Imaginary tool nose number 0 or 9

6.3.3 Setting the tool nose radius compensation value



This value is set from the MDI corresponding to the offset number specified in the lower 2 digits (note) of the 4-digit T code.

T

Offset number	OFX Offset value on X axis	OFZ Offset value on Z axis	OFR Tool nose radius compensation value	OFT Direction of imaginary tool nose
01	0.040	0.020	0.20	1
02	0.060	0.030	0.25	2
⋮	⋮	⋮	⋮	⋮
31	0.050	0.015	0.12	6
32	0.030	0.025	0.24	3

Max. 32 pairs

(Note) Offset numbers may be set in the least significant digit of the T code depending on the parameter setting. In this case, the range of offset numbers is from 1 to 9.

The range of the offset value is as follows:

	Metric system	Inch system
Offset value	0 ~ ±999.999 mm	0 ~ ±99.9999 inches

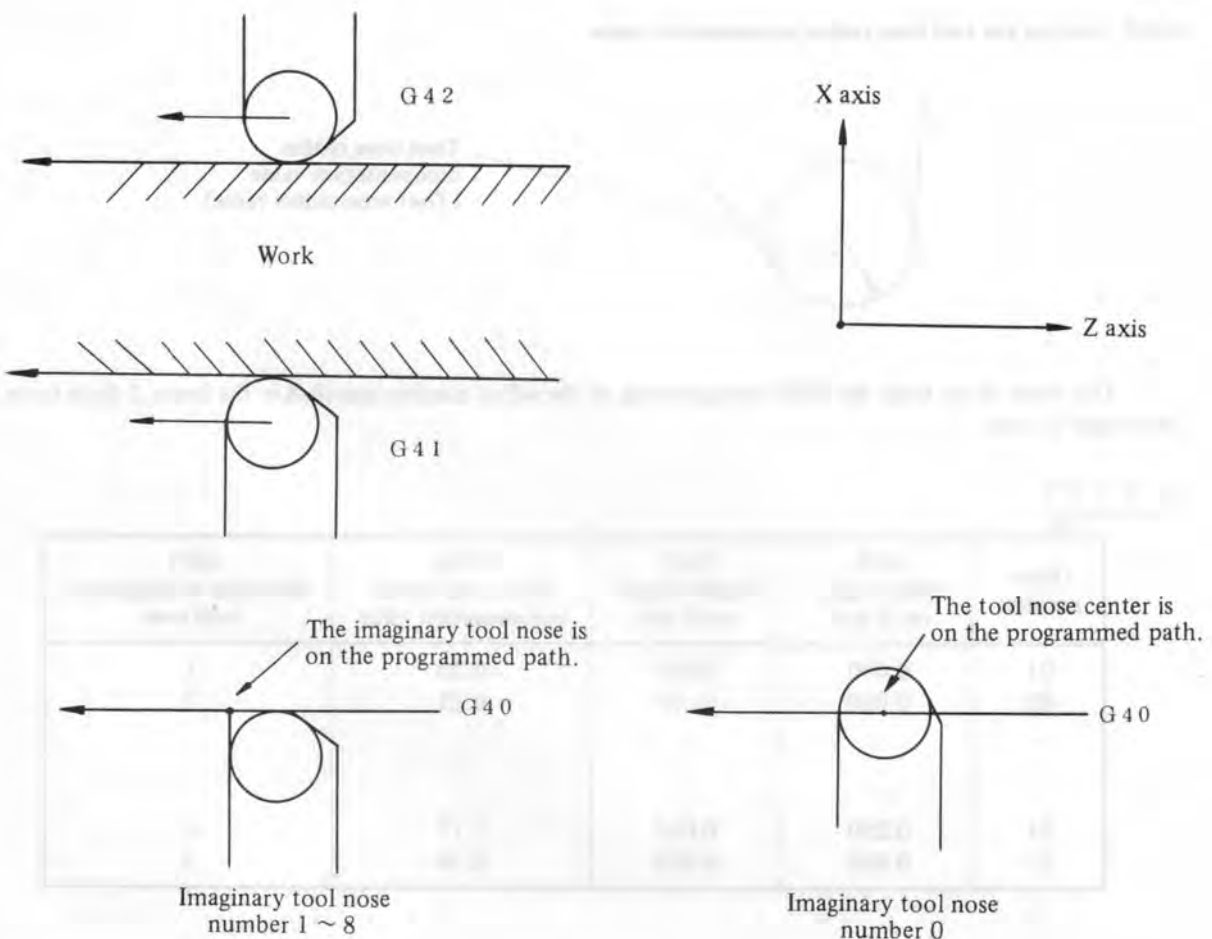
The offset value corresponding to the offset number 00 is always zero. No offset value can be set to offset number 00.

6.3.4 Work position and move command

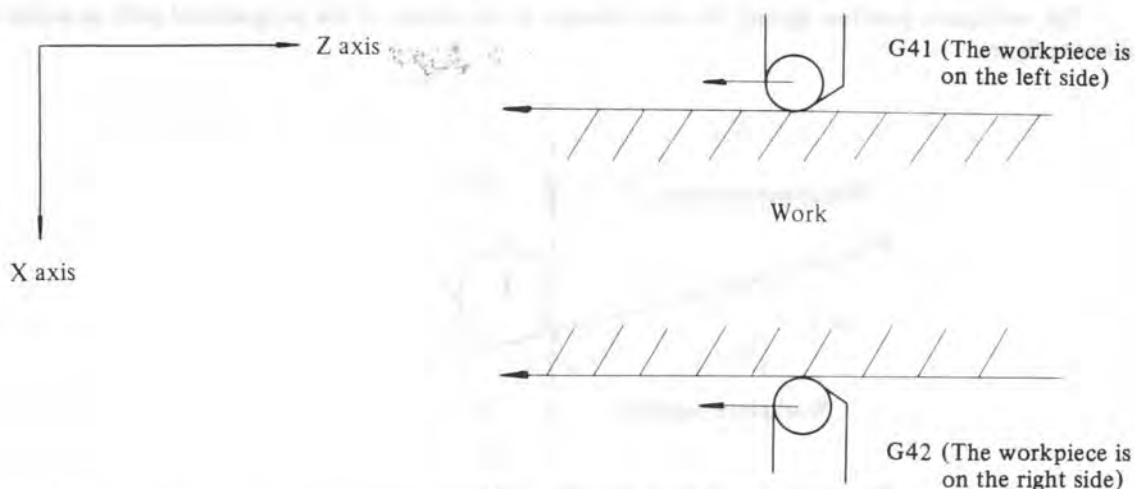
In tool nose radius compensation, the position of the workpiece with respect to the tool must be specified.

G code	Work position	Tool path
G40	(Cancel)	Moving along the programmed path
G41	Right side	Moving on the left side of the programmed path
G42	Left side	Moving on the right side of the programmed path

The tool is offset to the side opposite the side of the workpiece.



The workpiece position can be changed by setting the coordinate system as shown below.



(Note) If the tool nose radius compensation value is negative, the workpiece position is changed.

G40, G41 and G42 are modal.

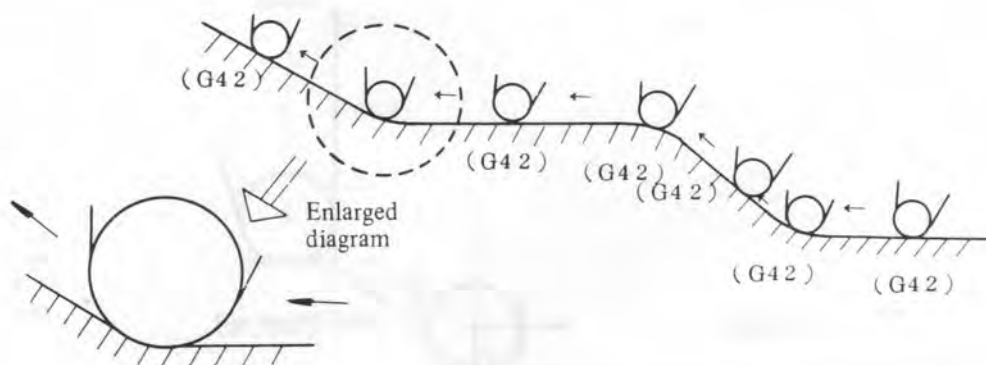
G41	X		Z		:	} G41 mode
	X		Z		:	
	X		Z		:	
G42	X		Z		:	} G42 mode
	X		Z		:	
	X		Z		:	
G40	X		Z		:	} G40 mode
	X		Z		:	
	X		Z		:	

Don't specify G41 while in the G41 mode. If you do, compensation is not same as usual. Don't specify G42 while in the G42 mode for the same reason.

G41 or G42 mode blocks in which G41 or G42 are not specified are expressed by (G41) or (G42) respectively.

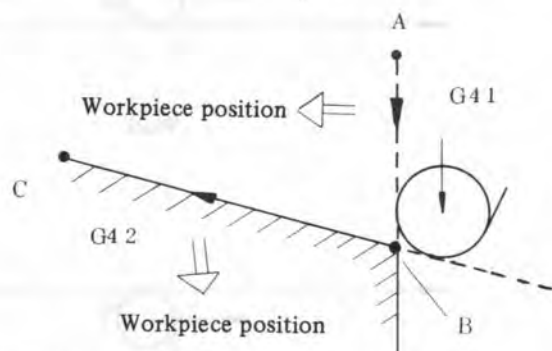
(1) When the workpiece position does not change

When the tool is moving, the tool nose maintains contact with the workpiece.



(2) When the workpiece position changes

The workpiece position against the tool changes at the corner of the programmed path as shown in the following figure.



Programmed path from A to B: G41

Programmed path from B to C: G42

Although the workpiece does not exist on the right side of the programmed path in the above case, the existence of the workpiece is assumed in the movement from A to B. The workpiece position must not be changed in the block next to the start-up block. In the above example, if the block specifying motion from A to B were the start-up block, the tool path would not be the same as the one shown.

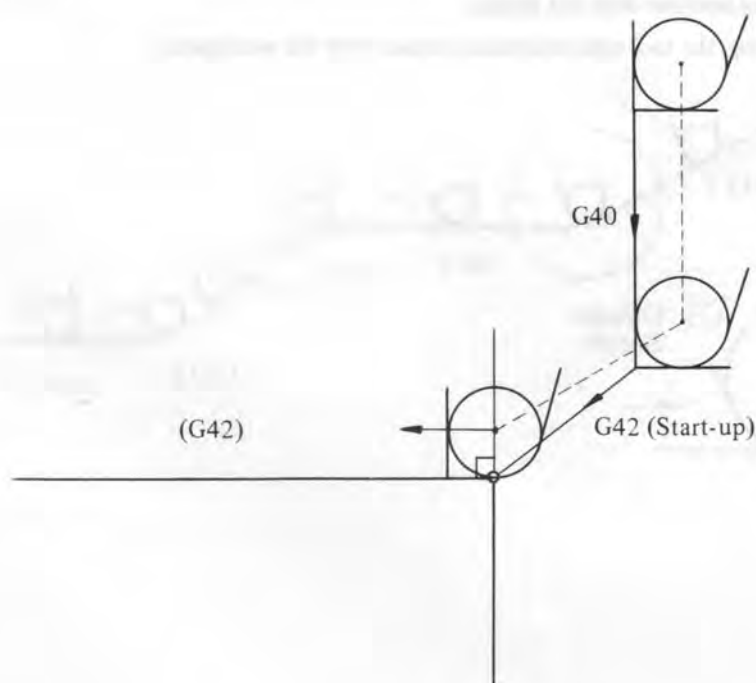
(3) Start-up

The block in which the mode changes to G41 or G42 from G40 is called the start-up block.

G40 _____ ;
 G41 _____ ; (Start-up block)
 _____ ;

Transient tool movements for offset are performed in the start-up block.

In the block after the start-up block, the tool nose center is positioned vertically to the programmed path of that block at the start point.

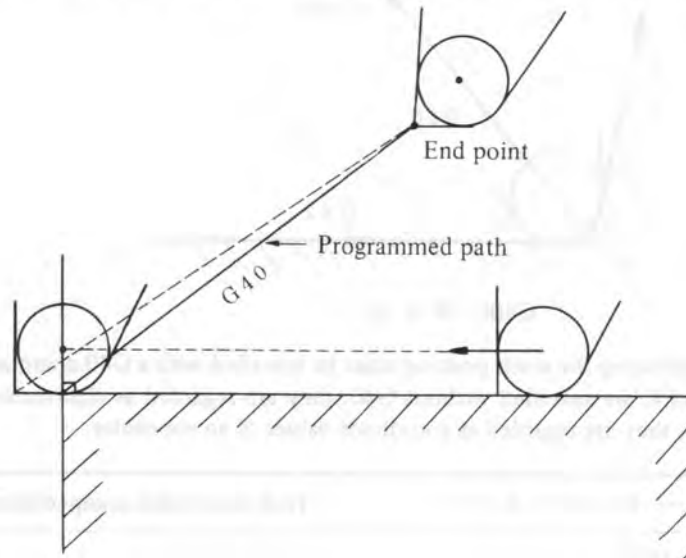


(4) Offset cancel

The block in which the mode changes to G40 from G41 or G42 is called the offset cancel block.

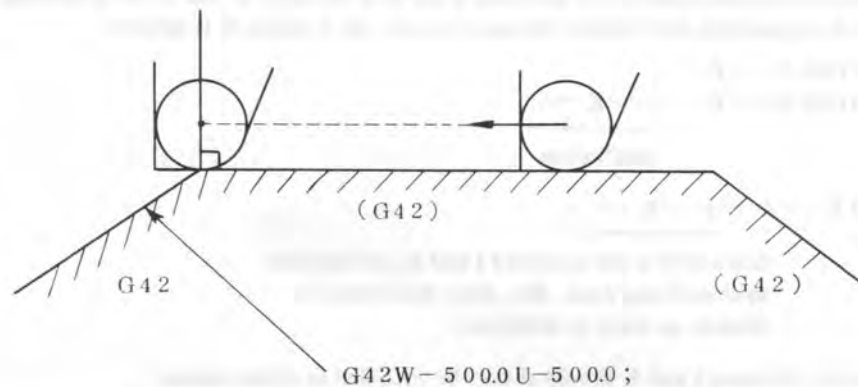
```
G41  
;  
G40 ; (Offset cancel block)
```

The tool nose center moves to a position vertical to the programmed path in the block before the cancel block. The tool is positioned at the end point in the offset cancel block (G40) as shown below.



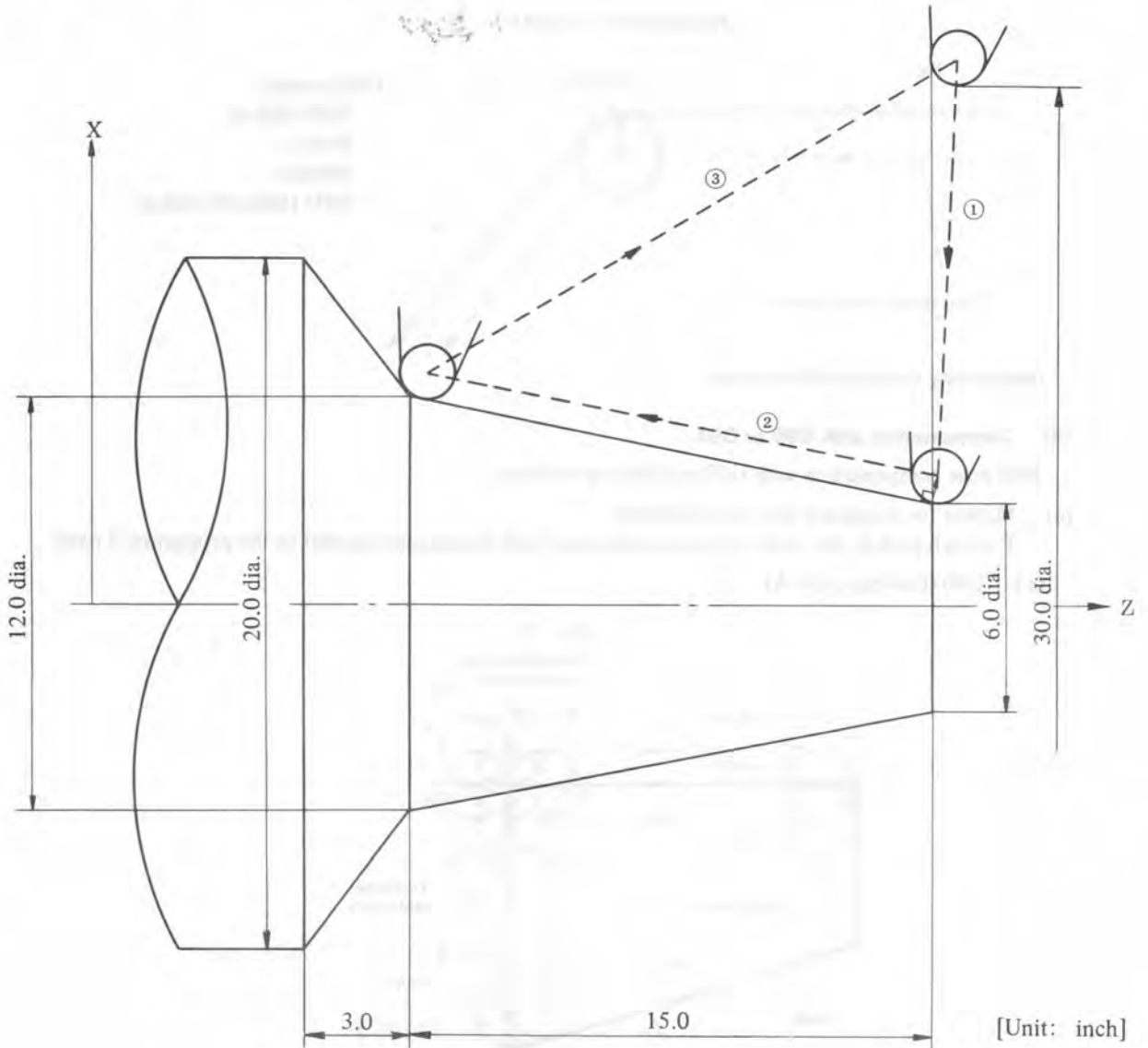
(5) When G41/G42 is specified again in G41/G42 mode

In this case, the tool nose center is positioned vertical to the programmed path of the preceding block at the end point of the preceding block.



In the block that first specifies G41/42, the above positioning of the tool nose center is not performed.

(7) Example



(In G40 mode, radius programming)

- ① G42 G00 X 3.0 ;
- ② G01 X 6.0 W-15.0 F1 ;
- ③ G40 G00 X 15.0 W 15.0 I 4.0 K-3.0 ;

6.3.5 Notes in tool nose radius compensation

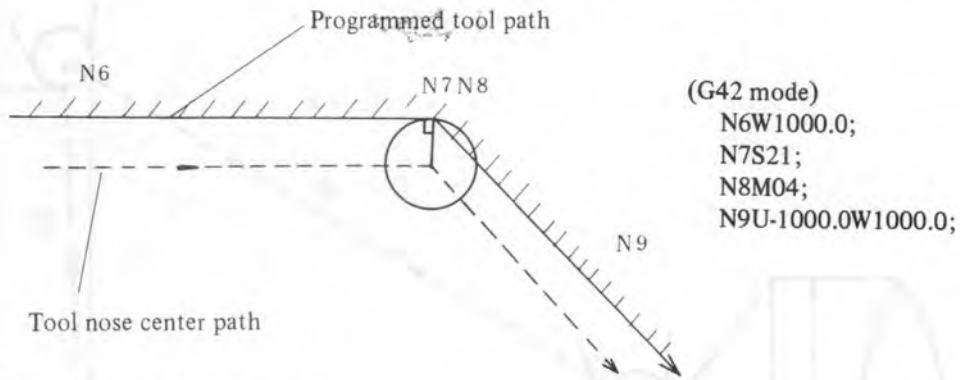
(1) Two or more blocks without a move command should not be programmed consecutively.

Blocks without a move command are:

- ① M05 ; M code output
- ② S21 ; S code output
- ③ G04 X 1000 ; Dwell
- ④ G01 U0 ; Feed distance of zero
- ⑤ G98 ; G code only
- ⑥ G22 X 10000 ; Machining area setting
- ⑦ G10 P 01 X 100 Z 200 R50 Q2 ; Offset change

If two or more of the above blocks are specified consecutively, the tool nose center comes to a position vertical to the programmed path of the preceding block at the end of the preceding block. However, if the no-

movement command is ④ above, the above tool motion is attained only with one block.



Overcutting occurs in this example.

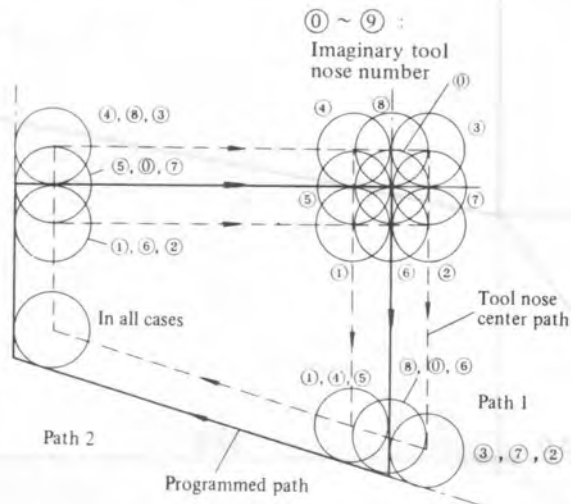
(2) Compensation with G90 or G94

Tool nose compensation with G90 or G94 is as follows:

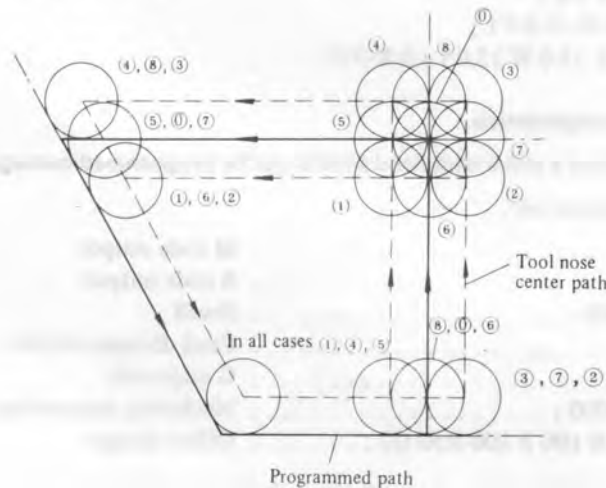
(a) Motion for imaginary tool nose numbers

For each path in the cycle, the tool nose center path is generally parallel to the programmed path.

(i) G90 (Cutting cycle A)

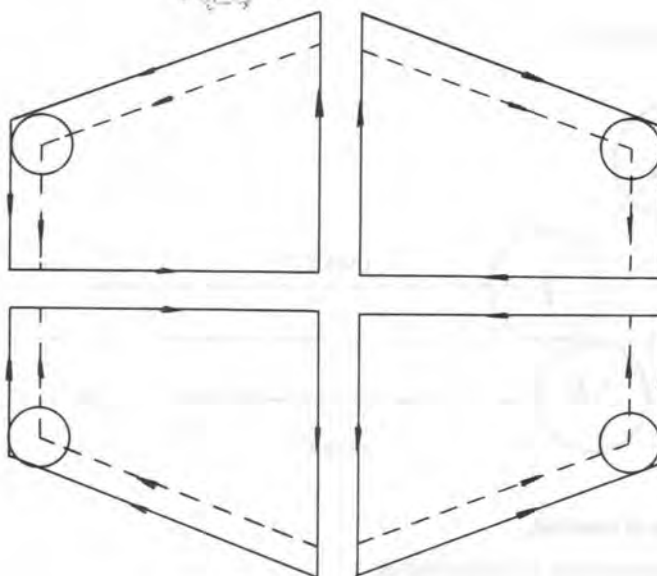


(ii) G94 (Cutting cycle B)

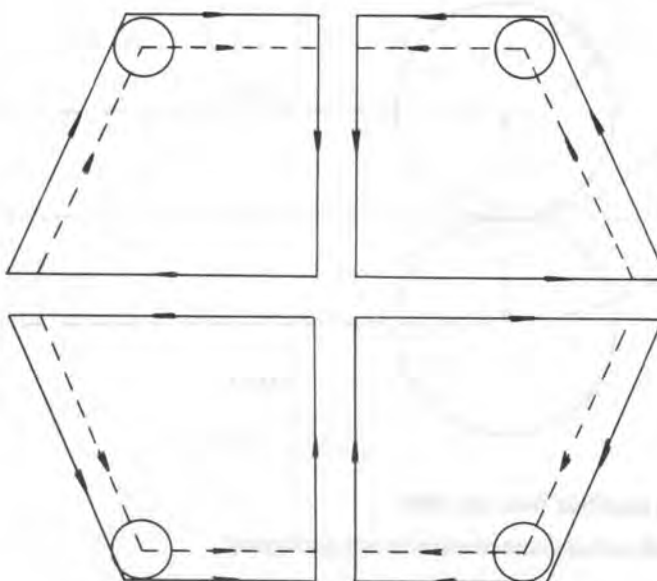


(b) The offset direction is indicated in the figure below regardless of the G41/G42 mode.

(i) G90



(ii) G94



(3) Compensation with G71, G72 or G73

If G71 was specified with tool nose radius compensation, when machining while the tool nose center coincides with the start point, the tool nose radius compensation value is added to the clearance value ΔU and ΔW but the tool nose radius compensation is cancelled during machining.

That is,

$$\Delta U = \Delta U + \text{Tool nose radius compensation value} (\Delta U \approx 0)$$

$$\Delta W = \Delta W + \text{Tool nose radius compensation value} (\Delta W \approx 0)$$

See Section 7.2.1.

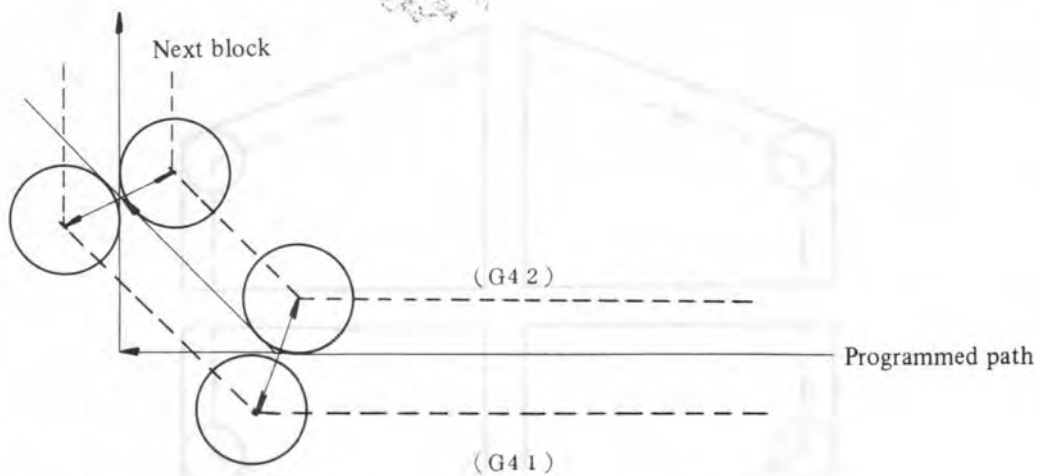
When machining while the imaginary tool nose coincides with the start point, or the clearance ΔU and ΔW are zero, the tool nose radius compensation value is not added to the clearance value.

(4) When G74 to G76 or G92 is specified.

Tool nose radius compensation is not performed in this case.

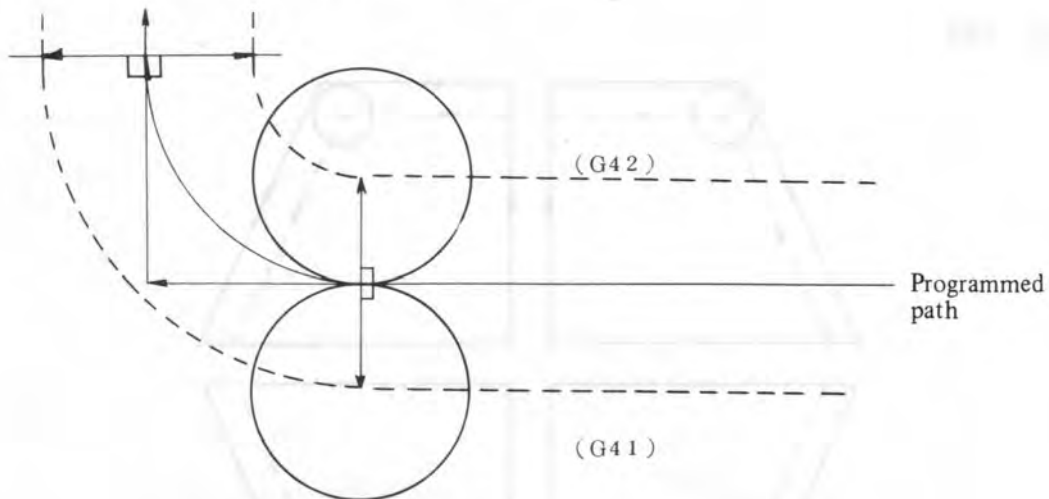
(5) When chamfering is performed.

Movement after compensation is shown below.



(6) When a corner arc is inserted.

Movement after compensation is shown below.

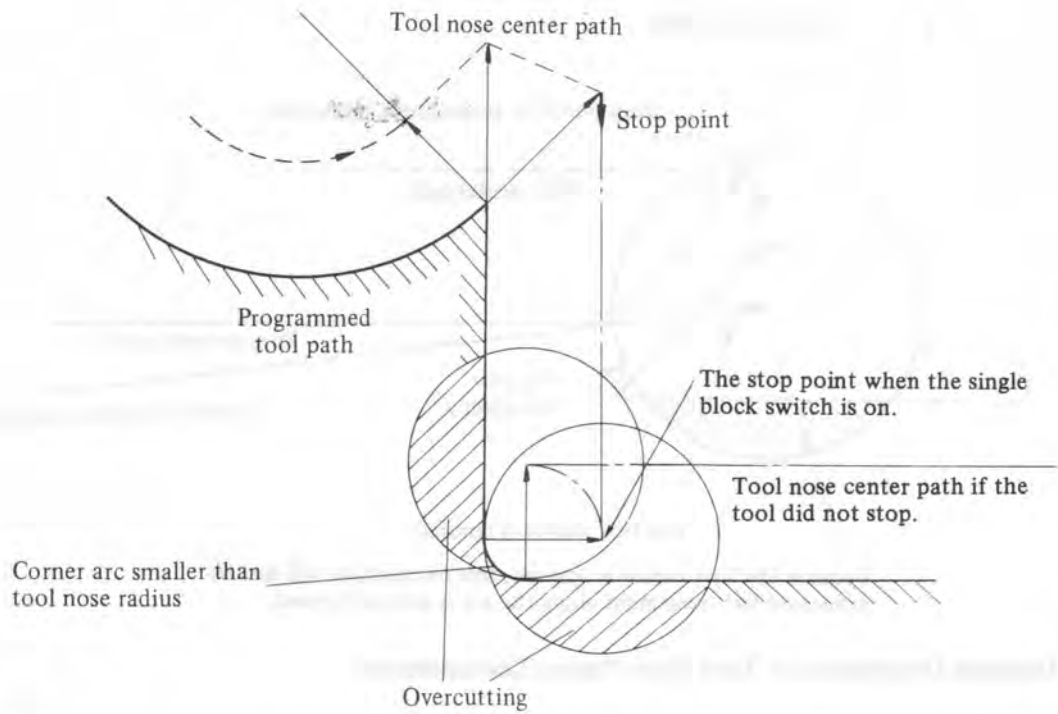


(7) When the block is specified from the DMI.

In this case, tool nose radius compensation is not performed.

(8) When machining at an inside corner smaller than the tool nose radius.

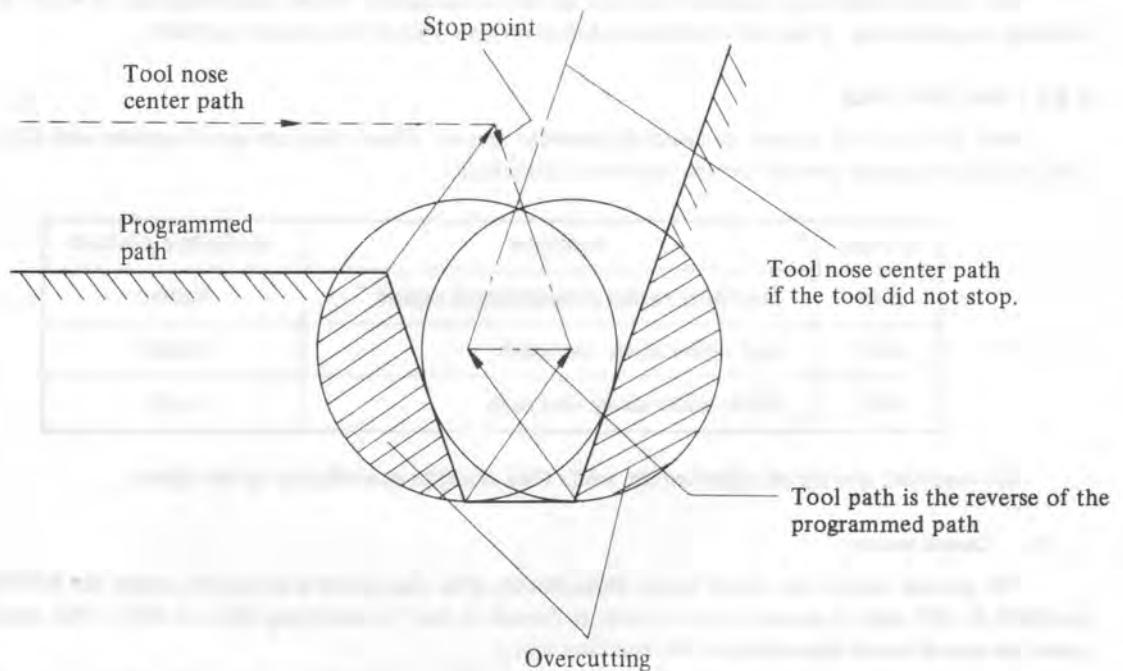
In this case, the inner offset of the tool will result in overcutting. The tool will stop and the alarm (No. 41) is displayed just after starting the preceding block, or terminating the corner movement if it exists. However, if the SINGLE BLOCK SWITCH is on, the tool stops at the end point of the preceding block. (This feature always applies to alarm No. 41.)



(9) Machining a groove smaller than the tool nose diameter.

When tool nose radius compensation forces the path of the tool nose center to move in a direction opposite to the programmed direction, overcutting will result.

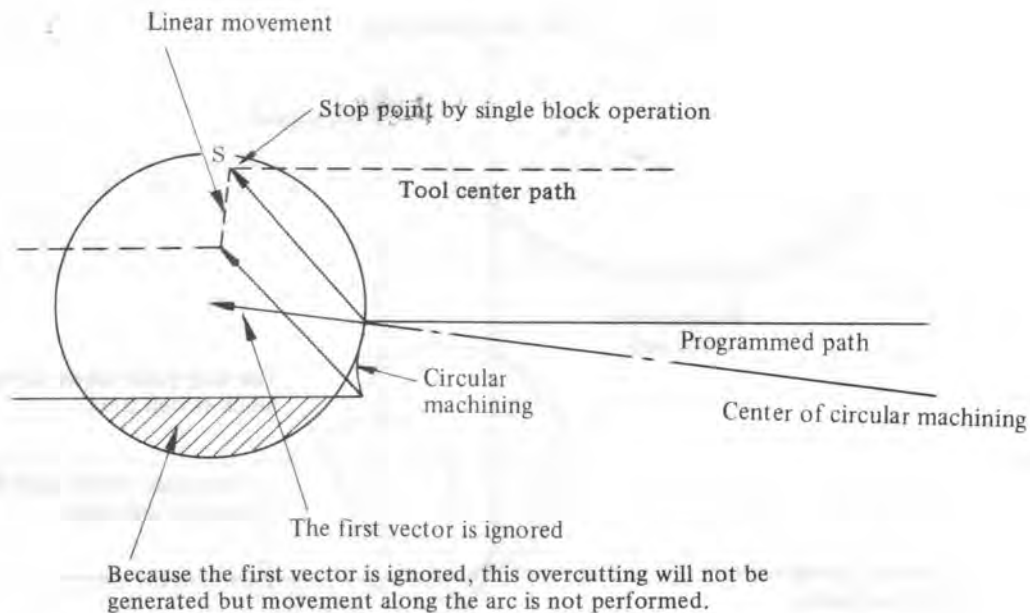
In this case, the alarm (No. 41) is displayed just after starting the preceding block (or after the corner movement if it exists) and the motion stops.



(10) When machining a step smaller than the tool radius

When a program contains a step smaller than the tool radius and this step is an arc, the path of the center of the tool may travel in the reverse of the programmed direction. In this case, the first vector is ignored, and the tool moves linearly to the second vector position. The tool may stop at this point by the single block operation. If not machining in the single block mode, cycle operation will continue.

If the step is specified with a line, the offset is properly performed without generating an alarm. (However, uncut parts remain.)



6.4 Detailed Description of Tool Nose Radius Compensation

6.4.1 Tool nose R center offset vector

The tool nose R center offset vector is a two dimensional vector equal to the offset value specified in a T code, and is calculated in the NC. Its direction changes block by block according to tool movement. This offset vector (simply called vector hereinafter) is internally created by the control unit as required for proper offsetting and to calculate a tool path with exact offset (by tool nose radius) from the programmed path. This vector is deleted by resetting.

The vector always accompanies the tool as the tool advances. Proper understanding of vector is essential to accurate programming. Read the description below on how vectors are created carefully.

6.4.2 G40, G41, G42

G40, G41 or G42 is used to delete or generate vectors. These codes are used together with G00, G01, G02, G03 or G32 to specify a mode for tool motion (offsetting).

G code	Function	Workpiece position
G40	Tool nose radius compensation cancel	Neither
G41	Left offset along tool path	Right
G42	Right offset along tool path	Left

G41 and G42 specify an offset mode, while G40 specifies cancellation of the offset.

(1) Cancel mode

The system enters the cancel mode immediately after the power is turned on, when the RESET button on the MDI & DPL unit is pushed or a program is forced to end by executing M02 or M30. (The system may not enter the cancel mode depending on the machine tool.)

In the cancel mode, the vector is set to zero, and the path of the center of tool nose coincides with the programmed path. A program must end in cancel mode. If it ends in the offset mode, the tool cannot be positioned at the end point, and the tool stops at a location the vector length away from the end point.

(2) Start-up

When a block which satisfies all the following conditions is executed in cancel mode, the system enters the offset mode. Control during this operation is called start-up.

- (i) G41 or G42 is contained in the block, or has been specified to set the system in G41 or G42 mode.
- (ii) The offset number for tool nose radius compensation is not 00.

(iii) X, Z, U or W moves is specified in the block and the move distance is not zero.

A circular command (G02 or G03) is not allowed in start up. If specified, alarm No. 34 will occur.

Two blocks are read in during start up. The first block is executed, and the second block is entered into the tool nose radius compensation buffer (this data cannot be displayed).

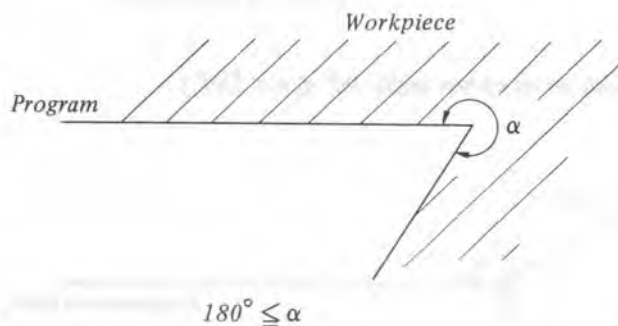
In the single block mode, two blocks are read and the first block is executed, then the machine stops.

In subsequent operations, two blocks are read in advance, so the NC has the block currently being executed, and the next two blocks.

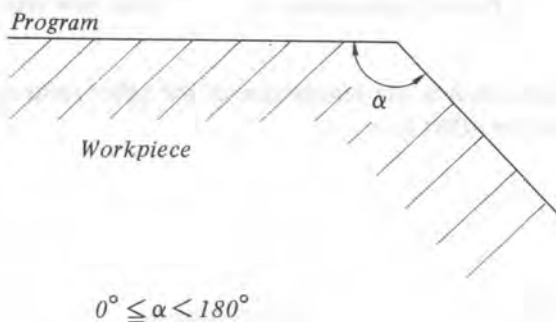
(Note) The meaning of "inner-side" and "outer-side" used later is as follows:

An angle of intersection created by two blocks of move commands is referred to as "inner-side" when it is over 180° and "outer-side", when it is from 0° to 180° as measured from the workpiece.

① Inner side

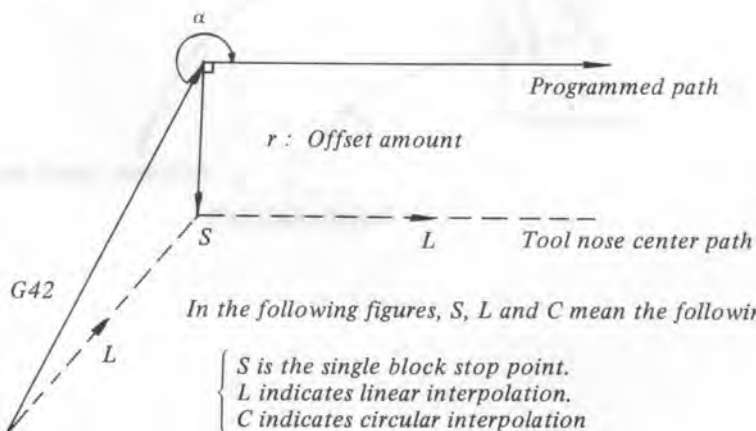


② Outer side

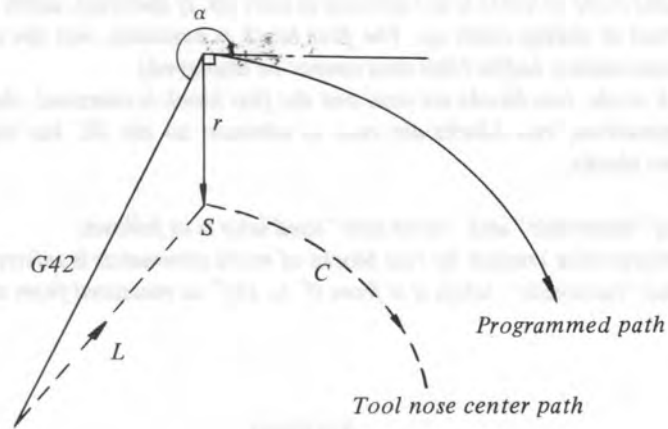


(a) Machining an inner wall ($180^\circ \leq \alpha$)

(i) Linear → Linear

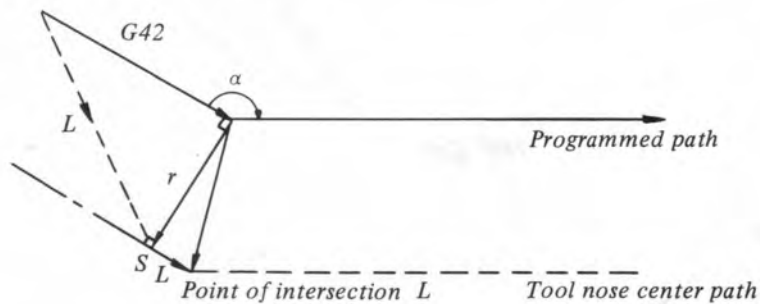


(ii) Linear $\rightarrow$ Circular



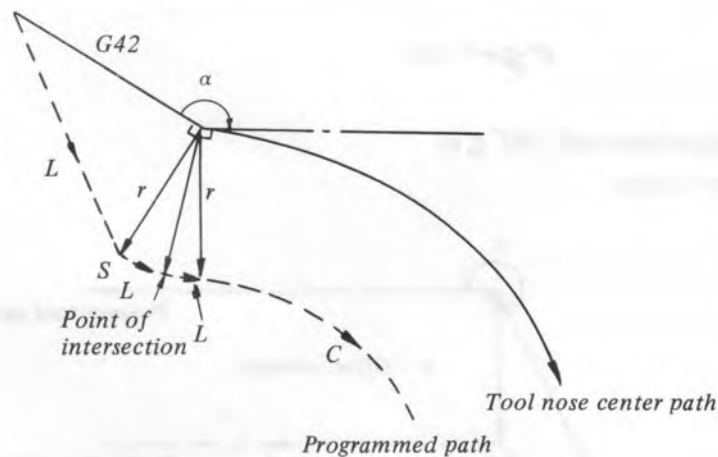
(b) Machining an outer wall (at an obtuse angle, $90^\circ \leq \alpha < 180^\circ$)

(i) Linear $\rightarrow$ Linear



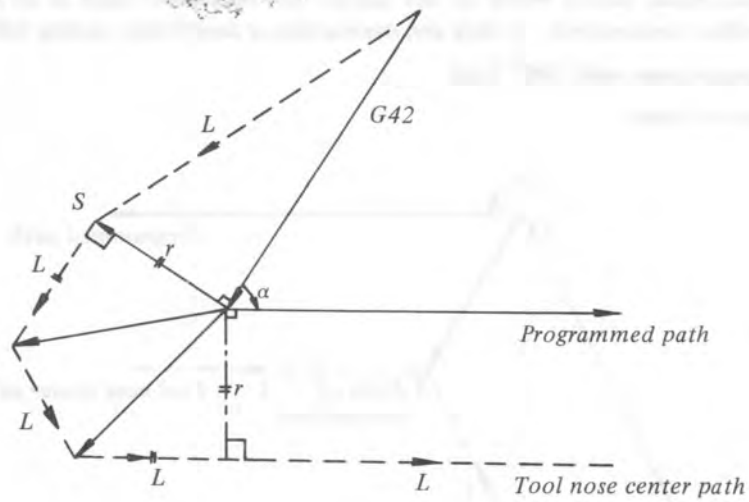
(Note) The point of intersection is the intersection of the offset paths of the two blocks where the programmed paths are offset by r .

(ii) Linear $\rightarrow$ Circular

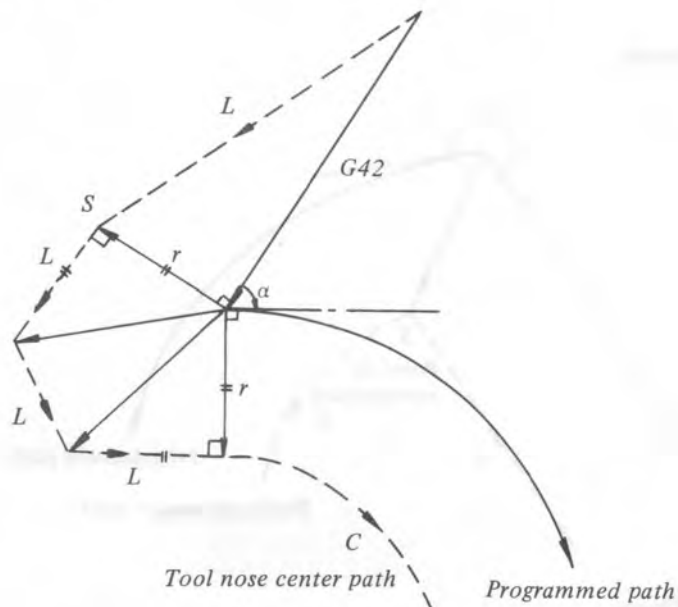


(c) Machining an outer wall (at an acute angle $\alpha < 90^\circ$)

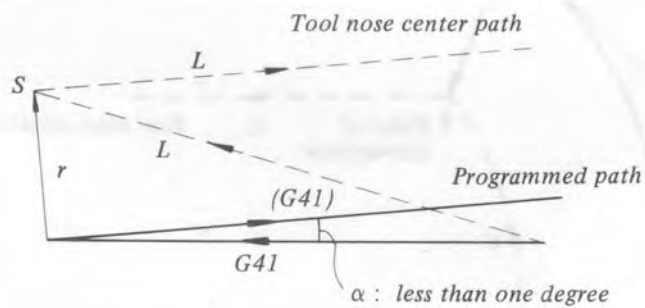
(i) Linear $\rightarrow$ Linear



(ii) Linear $\rightarrow$ Circular



(d) When machining an outer wall at a sharp corner (less than 1 degree) in linear-linear motion ($\alpha < 1^\circ$):

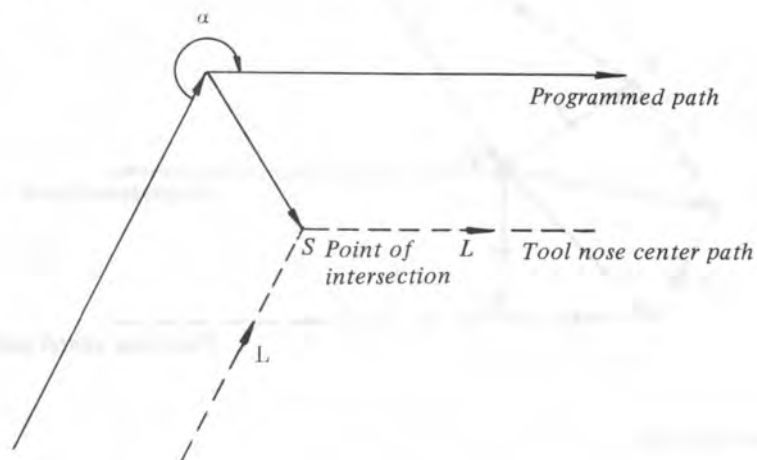


(3) Offset mode

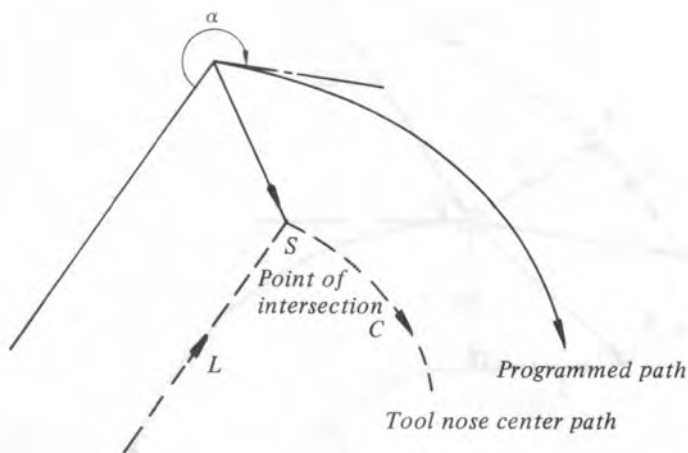
In the offset mode, tool offset is provided even during positioning, as well as linear and circular interpolation. In the offset mode, blocks which do not specify tool movement (such as an M function or dwell block) must not be specified consecutively. If they are, overcutting or insufficient cutting will result.

(a) Machining an inner wall ($180^\circ \leq \alpha$)

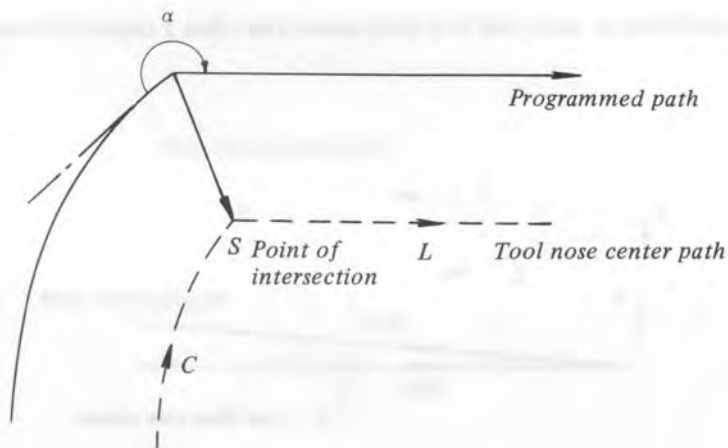
(i) Linear $\rightarrow$ Linear



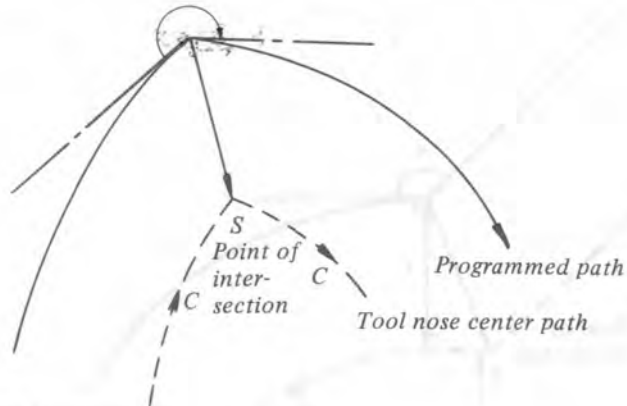
(ii) Linear $\rightarrow$ Circular



(iii) Circular $\rightarrow$ Linear

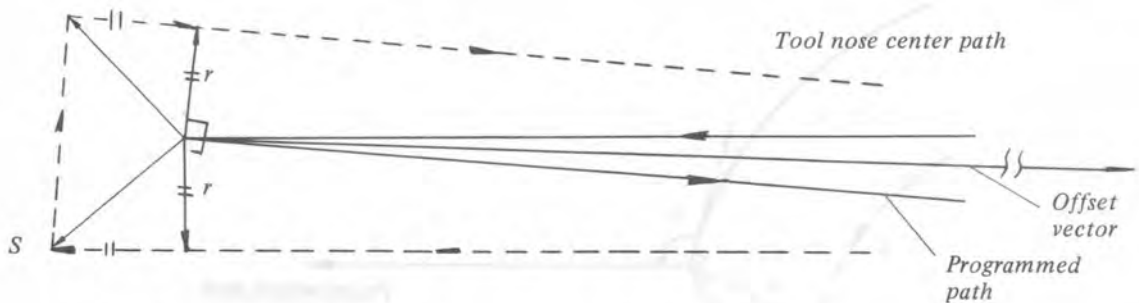


(iv) Circular $\rightarrow$ Circular



(v) When machining the inner wall at a sharp corner (less than 1 degree) and the offset vector is abnormally enlarged:

(i) Linear $\rightarrow$ Linear



Consider the following in the same way:

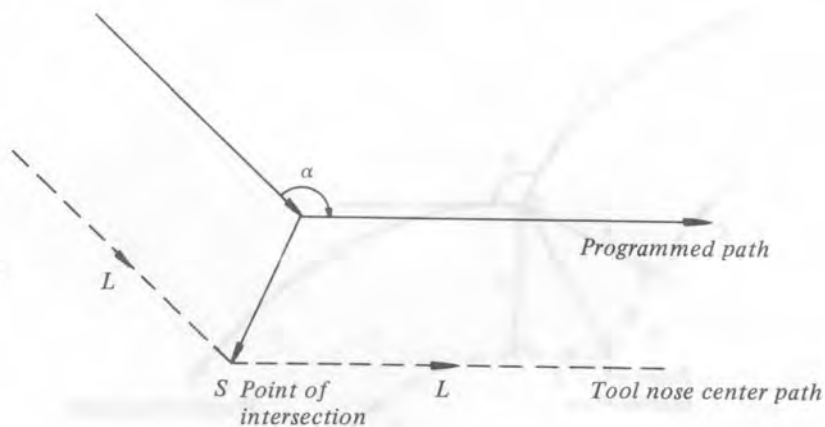
(ii) Circular $\rightarrow$ Linear

(iii) Linear $\rightarrow$ Circular

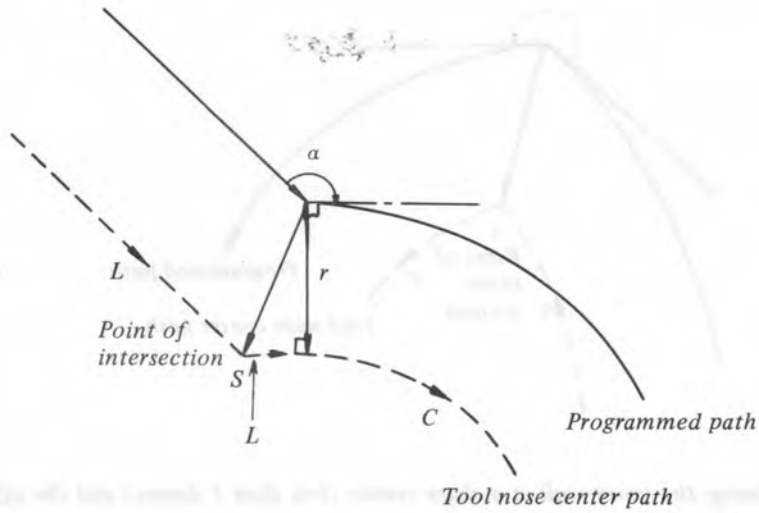
(iv) Circular $\rightarrow$ Circular

(b) Machining an outer wall (at an obtuse angle, $90^\circ \leq \alpha < 180^\circ$)

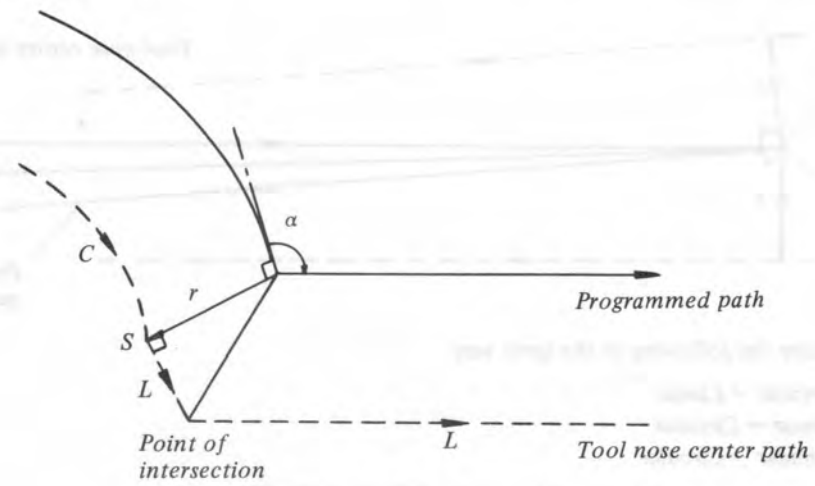
(i) Linear $\rightarrow$ Linear



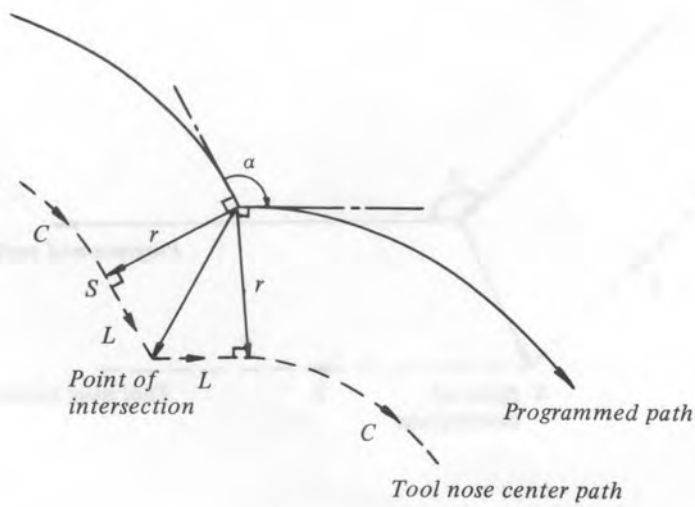
(ii) Linear $\rightarrow$ Circular



(iii) Circular $\rightarrow$ Linear

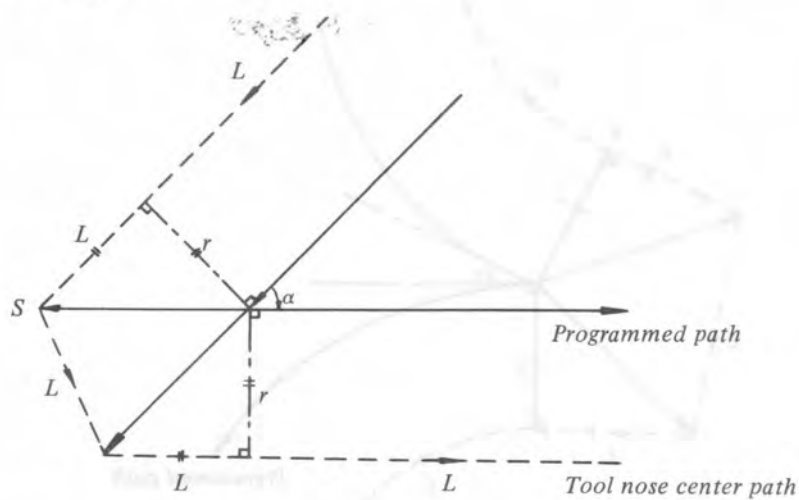


(iv) Circular $\rightarrow$ Circular

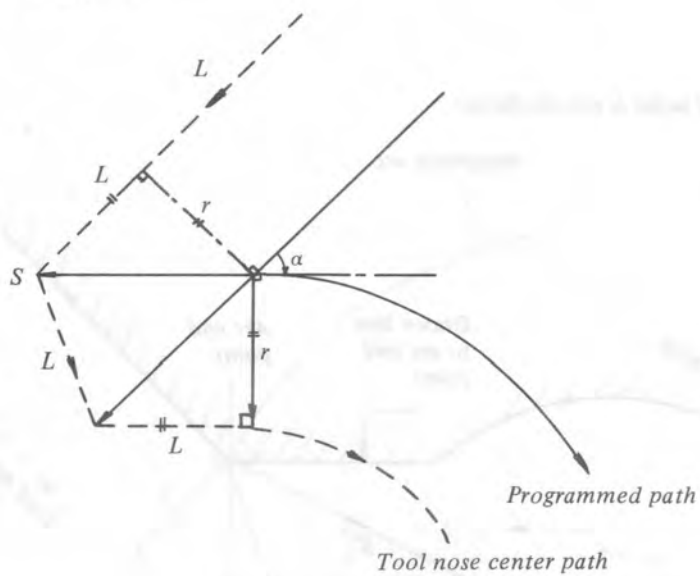


(c) Machining an outer wall (at an acute angle $\alpha < 90^\circ$)

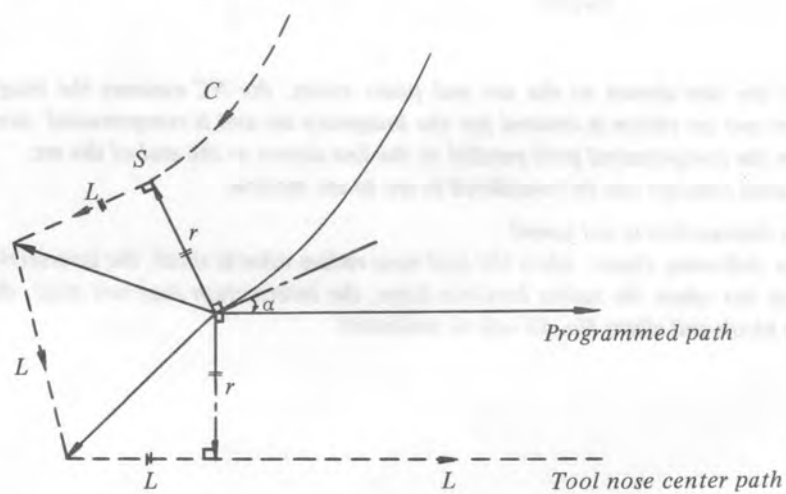
(i) Linear $\rightarrow$ Linear



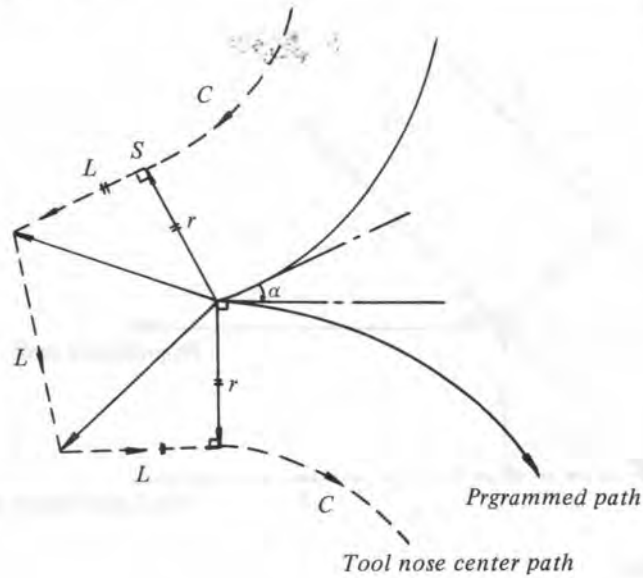
(ii) Linear $\rightarrow$ Circular



(iii) Circular $\rightarrow$ Linear

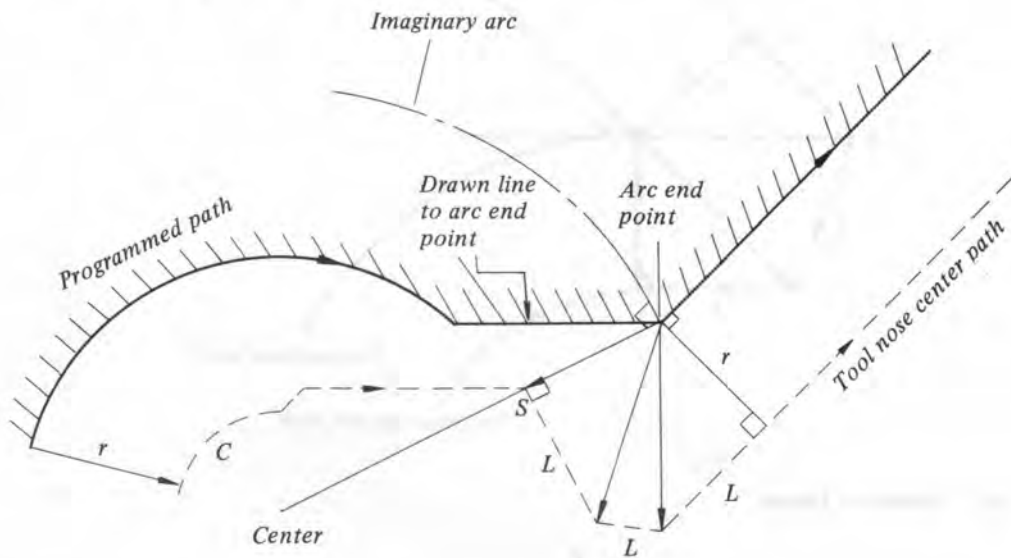


(iv) Circular $\rightarrow$ Circular



(d) Exceptional case

(i) When an arc end point is not on the arc

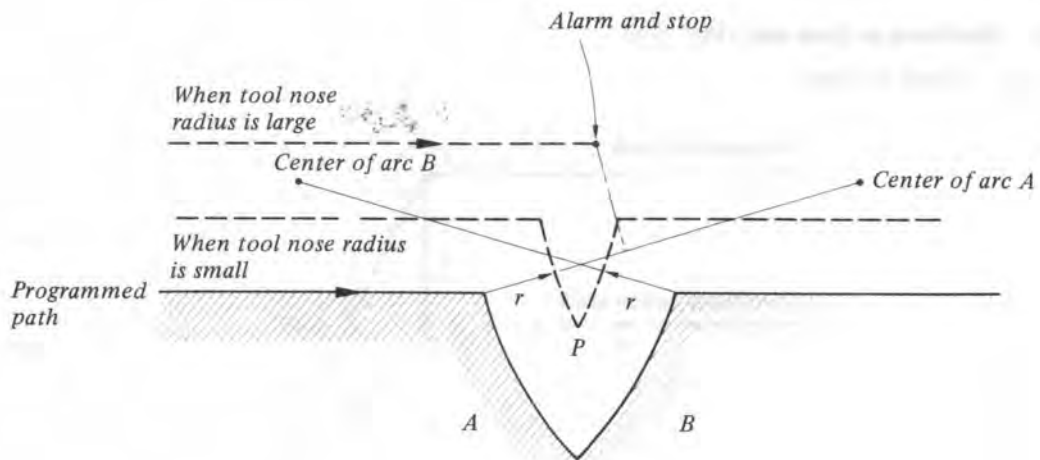


When the line drawn to the arc end point exists, the NC assumes the imaginary arc shown in the above figure and an vector is created for the imaginary arc and is compensated. Accordingly, the tool path differs from the compensated path parallel to the line drawn to the end of the arc.

The same concept can be considered in arc to arc motion.

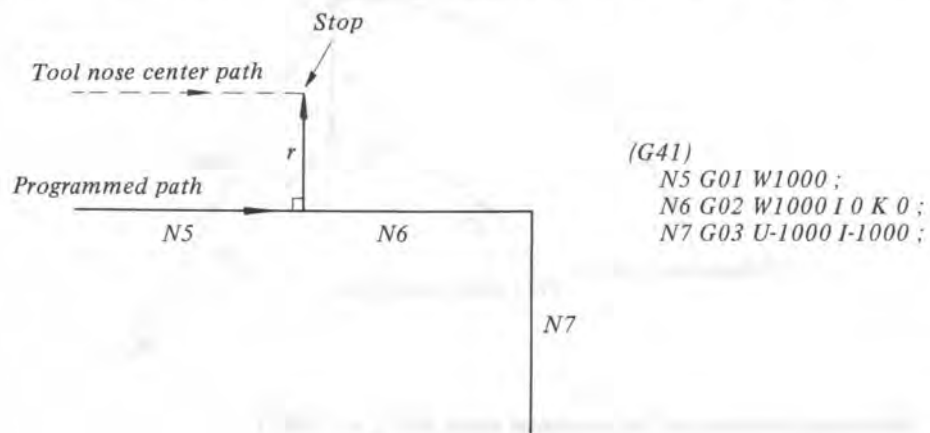
(ii) When intersection is not found

In the following figure, when the tool nose radius value is small, the intersection of the offset path of an arc exists but when the radius becomes large, the intersection may not exist, the tool stops at the end of previous block and alarm No. 33 will be indicated.



If the radius is small, point P of intersection of offset path exists, but when the radius is large, it is not found.

- (iii) When an arc, on which the center and start point or the center and end point are the same point is specified, alarm No. 38 is indicated and the tool stops at the end of the earlier block.



(4) Offset cancel

When a block which satisfies one of the following conditions is executed in the offset mode, the system enters the cancel mode. The control for this operation is called the offset cancel.

- G40 is specified.
- 00 is specified as the offset number of tool nose radius compensation.

Offset cancellation must not be specified in a circular command (G02, G03). If specified, alarm No. 34 is indicated.

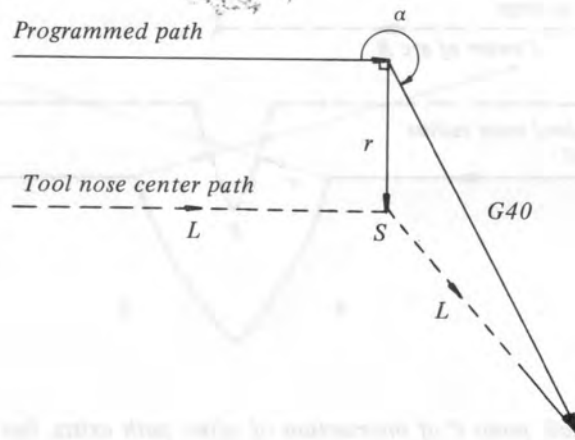
During offset cancellation, one block is read in and two blocks [including a block stored in the tool nose radius compensation buffer (this data cannot be displayed)] are executed.

In single block mode, one block is read in and one block is executed and execution stops. Then pushing the START button executes one block without reading in the next block.

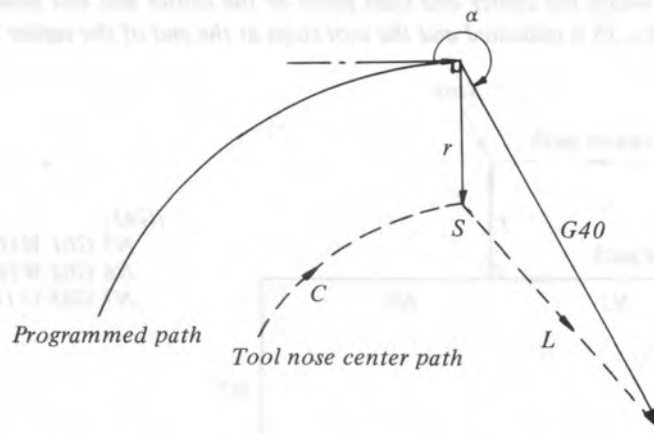
Then, the system enters the cancel mode and only the block to be executed next is stored in the buffer storage. No data is read into the tool nose radius compensation buffer.

(a) Machining an inner wall ($180^\circ \leq \alpha$)

(i) Linear $\rightarrow$ Linear

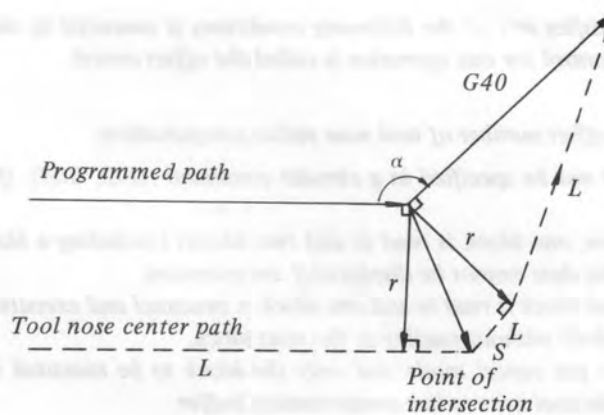


(ii) Circular $\rightarrow$ Linear

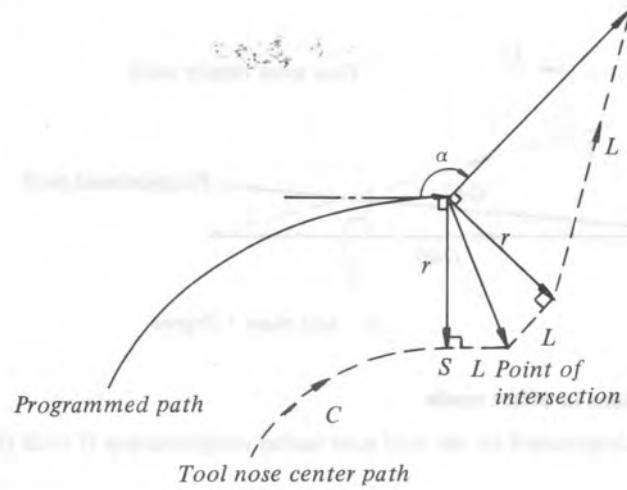


(b) Machining an outer wall (at an obtuse angle, $90^\circ \leq \alpha < 180^\circ$)

(i) Linear $\rightarrow$ Linear

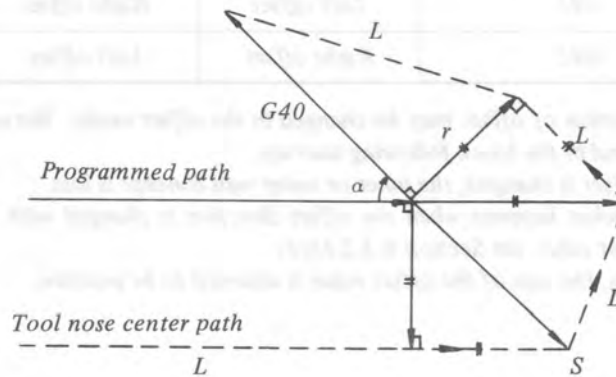


(ii) Circular $\rightarrow$ Linear

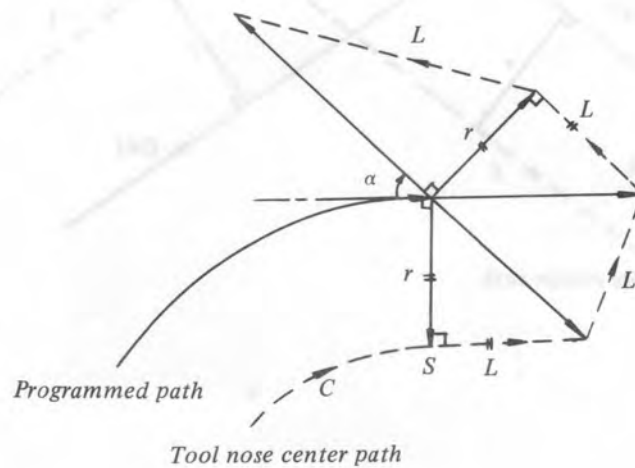


(c) When machining an outer wall (at an acute angle $\alpha < 90^\circ$)

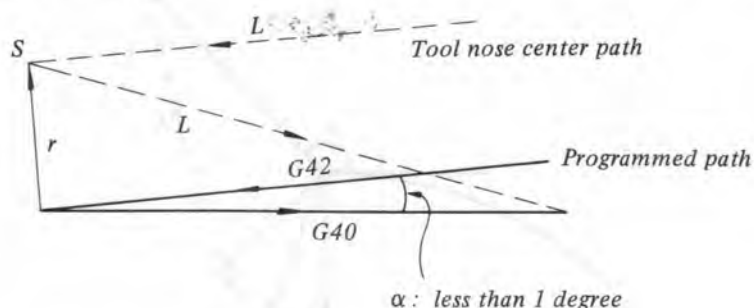
(i) Linear $\rightarrow$ Linear



(ii) Circular $\rightarrow$ Linear



- (d) When machining an outer wall at a sharp corner of less than 1 degree in linear-linear motion ($\alpha < 1^\circ$), an inner wall is assumed.



(5) Change of offset directions in offset mode

The direction of offset is determined by the tool nose radius compensation G code (G41, G42) and the sign of the offset value.

G code	Sign of offset value	
	+	-
G41	Left offset	Right offset
G42	Right offset	Left offset

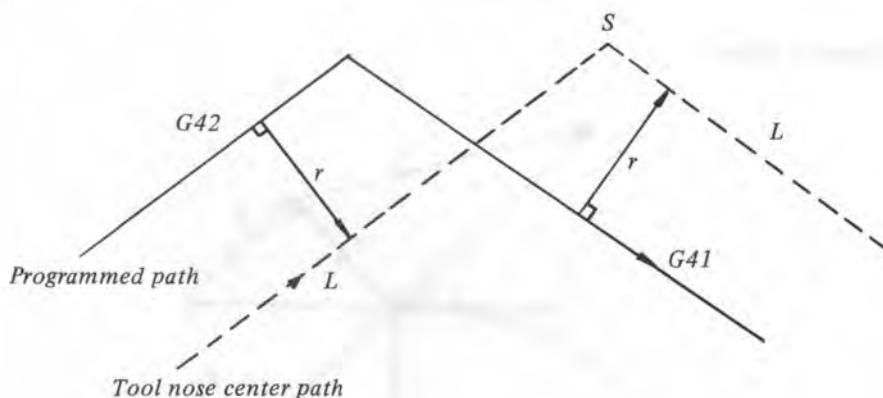
In special cases, the direction of offset may be changed in the offset mode. But offset direction cannot be changed in the start-up block and in the block following start-up.

When the direction of offset is changed, the inner or outer wall concept is lost.

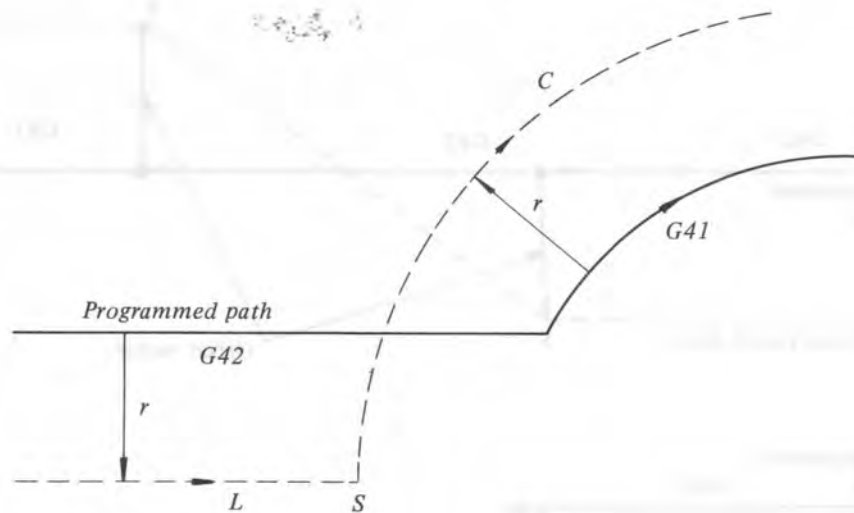
The following explains what happens when the offset direction is changed with G41 or G42. When it is changed by the sign of the offset value, see Section 6.3.2.16(3).

In the following examples, the sign of the offset value is assumed to be positive.

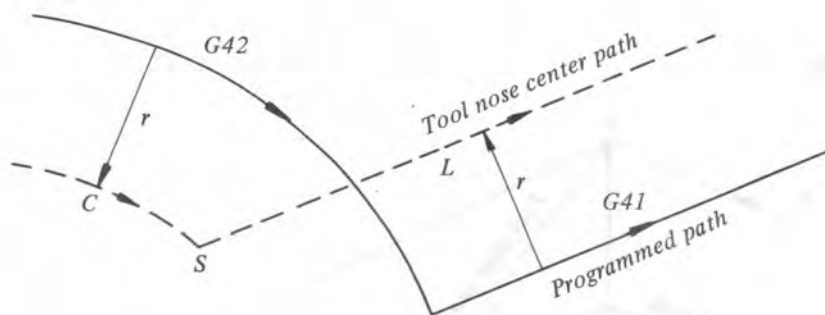
- (i) Linear → Linear



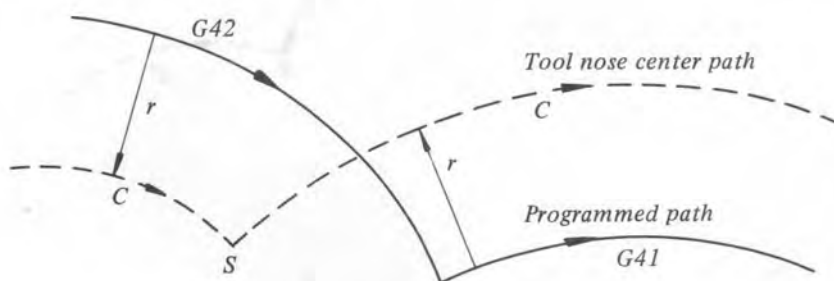
(ii) Linear $\rightarrow$ Circular



(iii) Circular $\rightarrow$ Linear



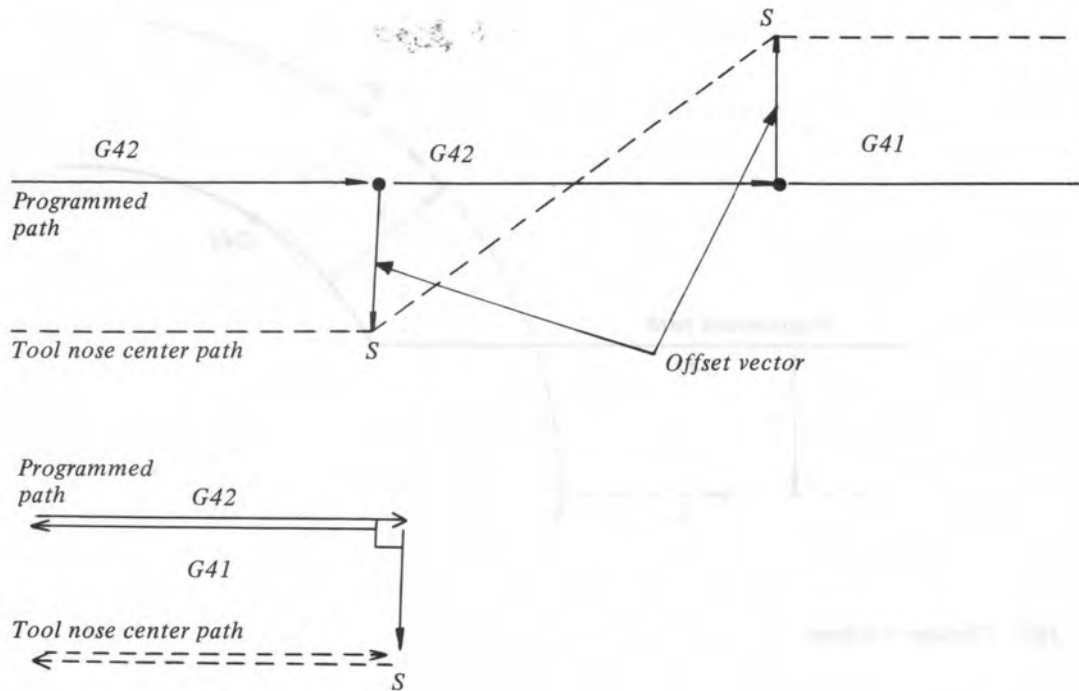
(iv) Circular $\rightarrow$ Circular



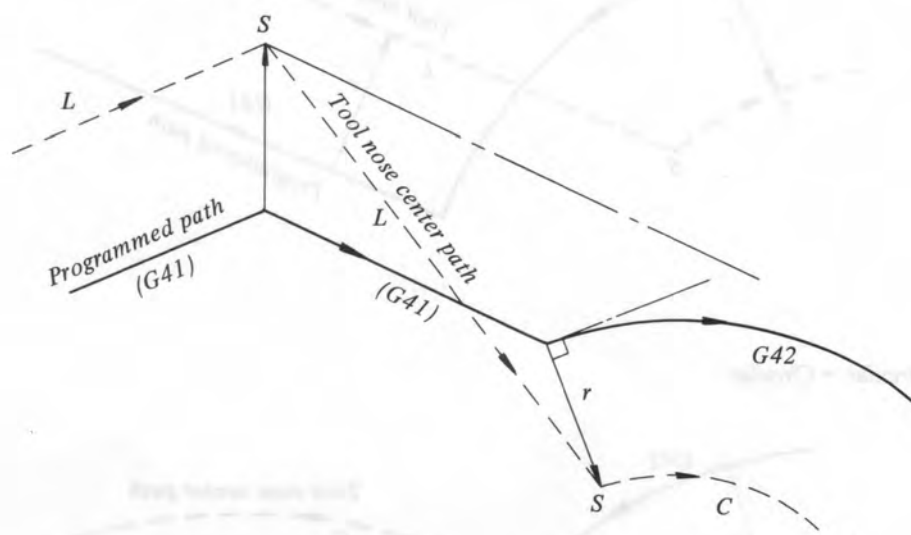
(v) When an intersection is not required if offset is normally performed.

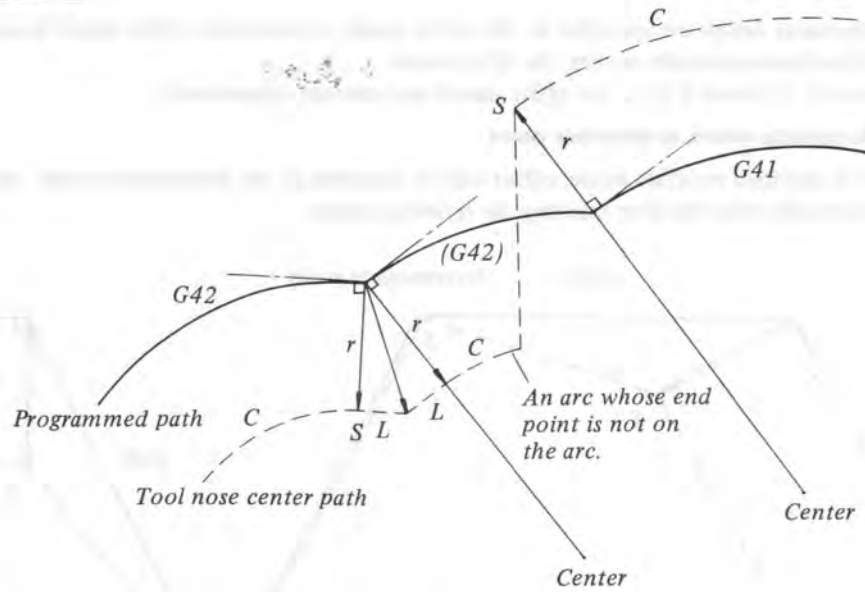
When changing the offset direction in block A to block B using G41 and G42, if intersection with the offset path is not required, the vector normal to block B is created at the start point of block B.

Linear → Linear



Linear → Circular





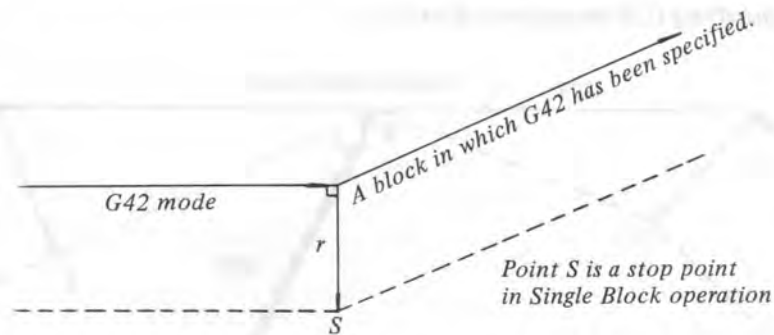
(6) Tool nose radius compensation G code in the offset mode

The offset vector can be set to make a right angle with the tooling direction in an earlier block, regardless of whether an inner or outer wall is being machined, by specifying the tool nose radius compensation G code (G41, G42) in the offset mode.

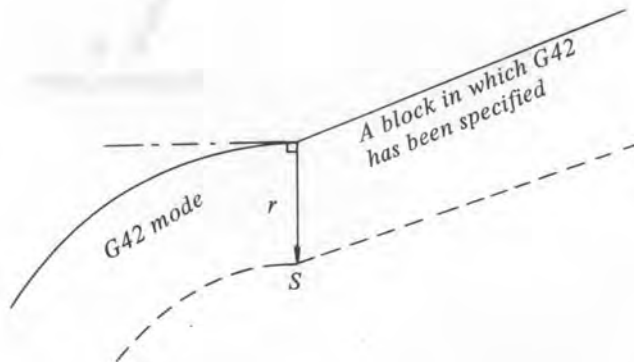
If this code is included in a circular command, correct circular motion will not be obtained.

When the direction of offset changes because of a tool nose radius compensation G code (G41, G42) specification, see Section 6.3.2.5.

Linear → Linear



Circular → Linear



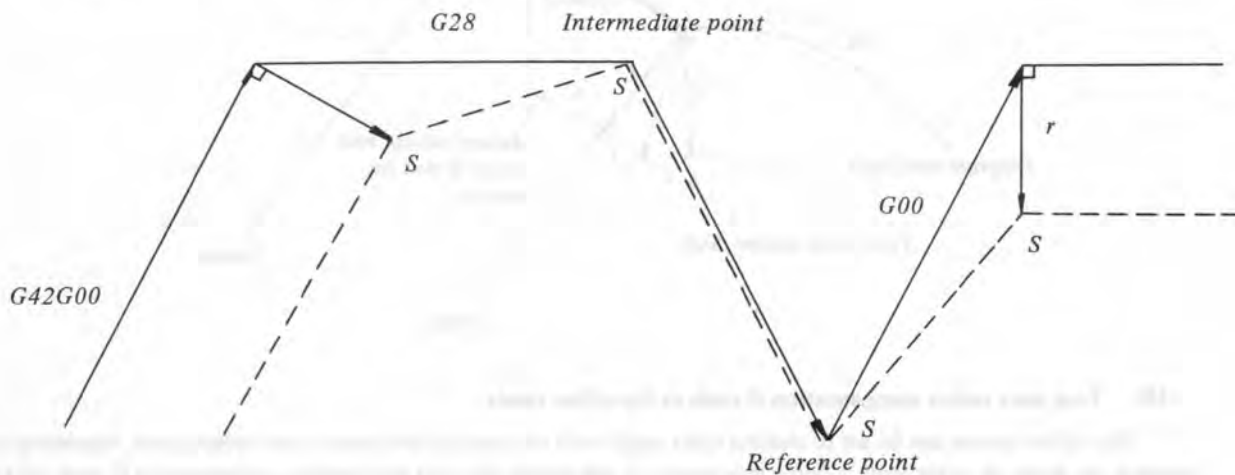
(7) Temporary offset cancel

If the commands below are specified in the offset mode, a temporary offset cancel is executed, and thereafter, the system will automatically restore the offset mode.

See Sections 6.3.2.4 and 6.3.2.2 for offset cancel and start-up, respectively.

(a) G28 Automatic return to reference point

If G28 is specified in offset mode, offset will be cancelled at the intermediate point, and the offset mode will be automatically restored after reaching the reference point.

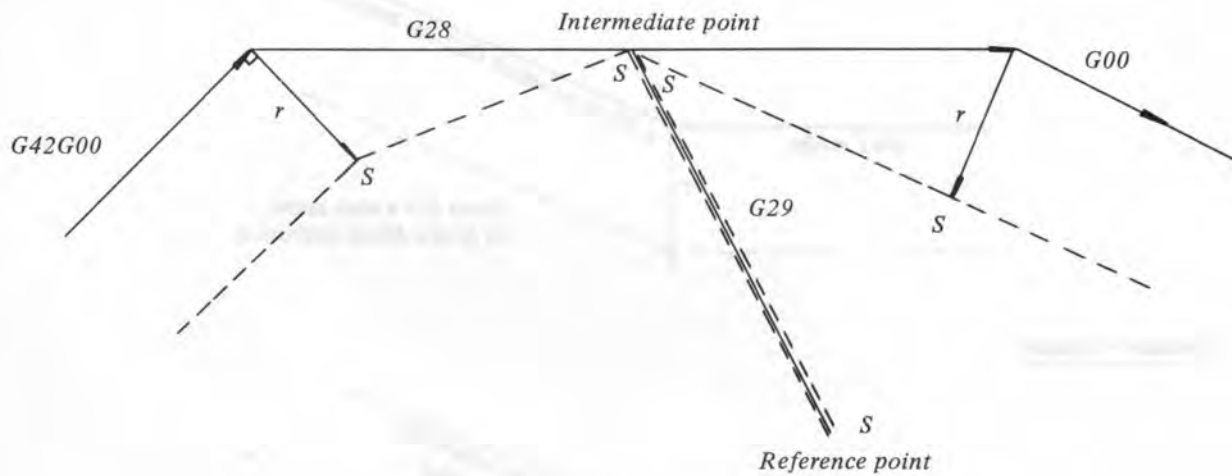


Point S is a stop point in Single Block operation

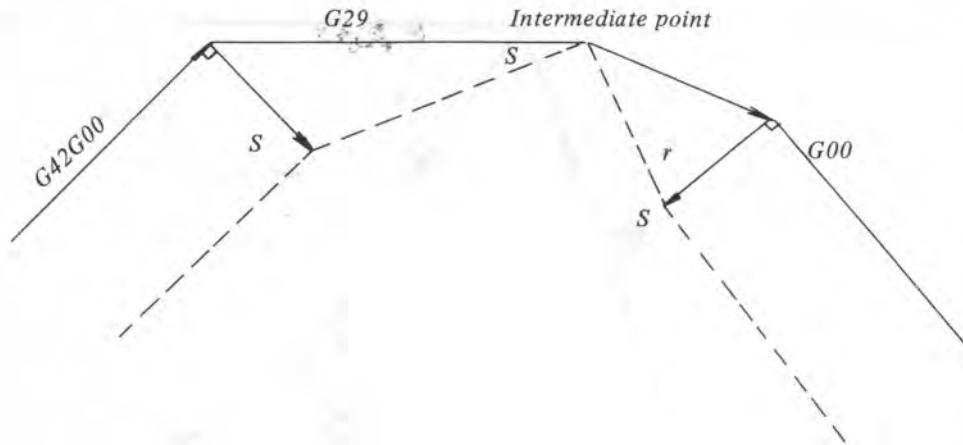
(b) G29 Automatic return from reference point

If G29 is specified in the offset mode, offset will be cancelled at the intermediate point, and the offset mode will be restored automatically in the next block.

(i) When specifying G29 immediately after G28



(ii) When the G29 command does not immediately follow G28

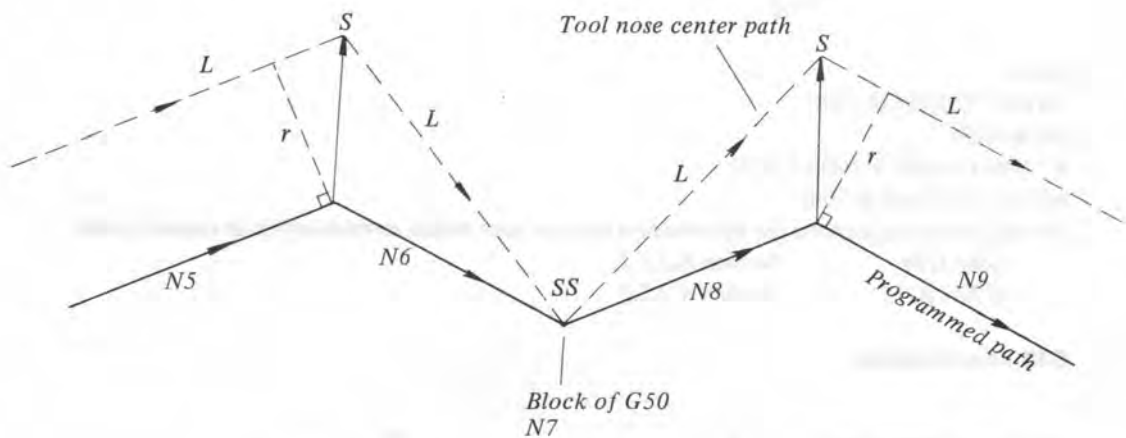


(8) Command to temporarily delete the vector

When the following commands are specified in the offset mode, the offset vector is temporarily deleted, then the offset mode will be automatically restored.

In this case, the offset cancel motion is not made but the center of the tool nose goes to the programmed point from the top of the vector at the intersection of the offset paths. When the system enters the offset mode, the center of the tool path goes directly to the top of the vector at the intersection of the offset paths.

(a) G50 Programming of absolute zero point



(G41 mode)

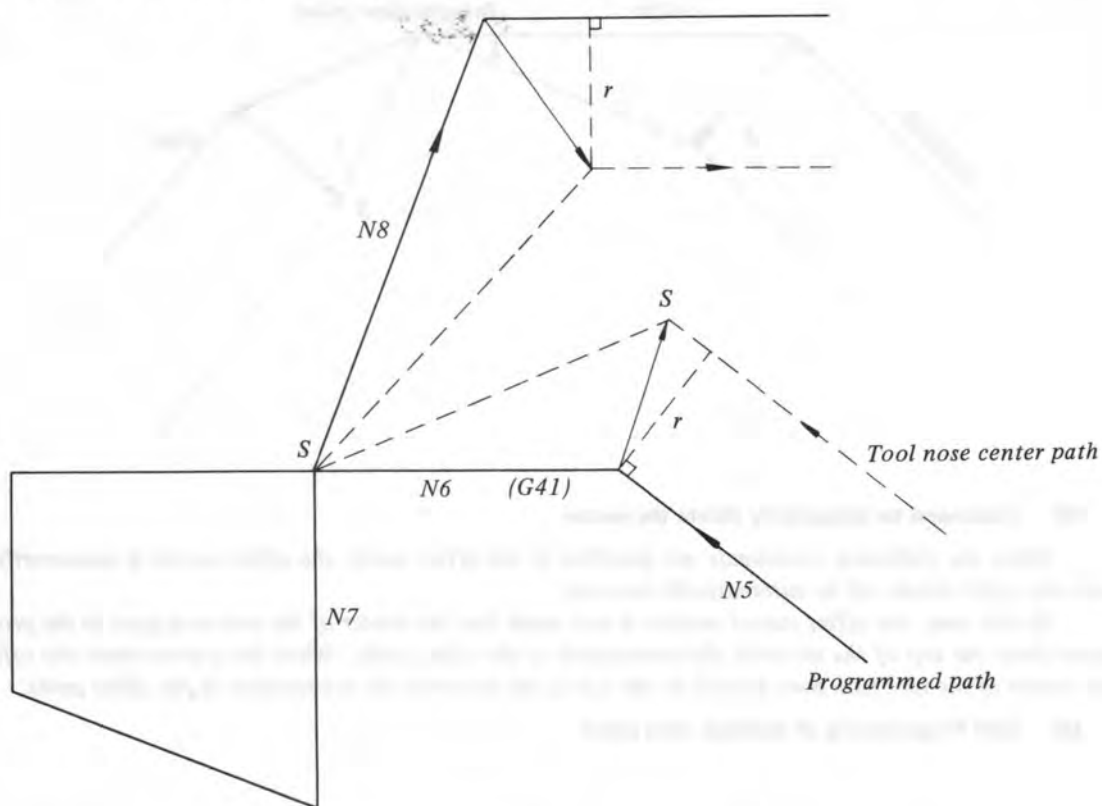
N5 G01 U 3000 W 7000 ;

N6 U-3000 W 6000 ;

N7 G50 X 1000 Z 2000 ;

N8 G01 X 4000 Z 8000 ;

- (b) G90, G92, G94 Canned cycles
G71 – G76 Multiple repetitive cycles



(G42)

N5 G01 U 5000 W-6000 ;

N6 W-8000 ;

N7 G90 U-6000 W-8000 I-3000 ;

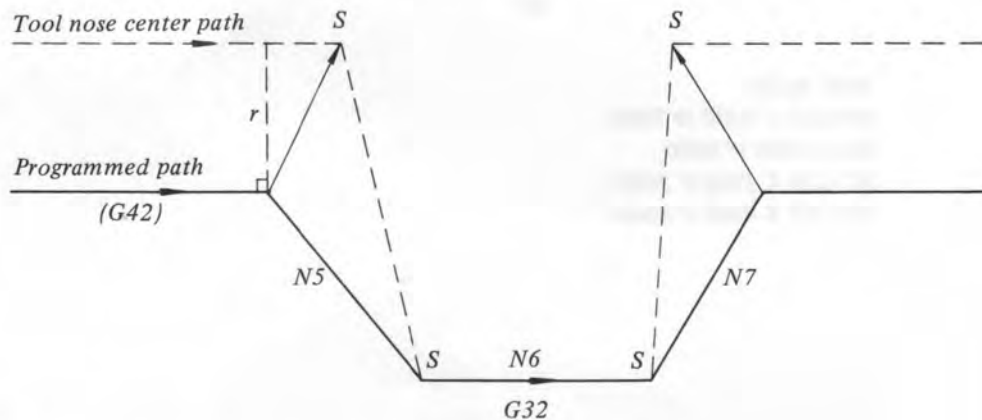
N8 G01 U 12000 W 5000 ;

See the following sections for information on tool nose radius compensation in canned cycles.

G90, G94 Section 6.2.5.2

G70 – G76 Section 6.2.5.3

- (c) G32 Thread cutting



The tool nose radius compensation is not in effect in a G32 block.

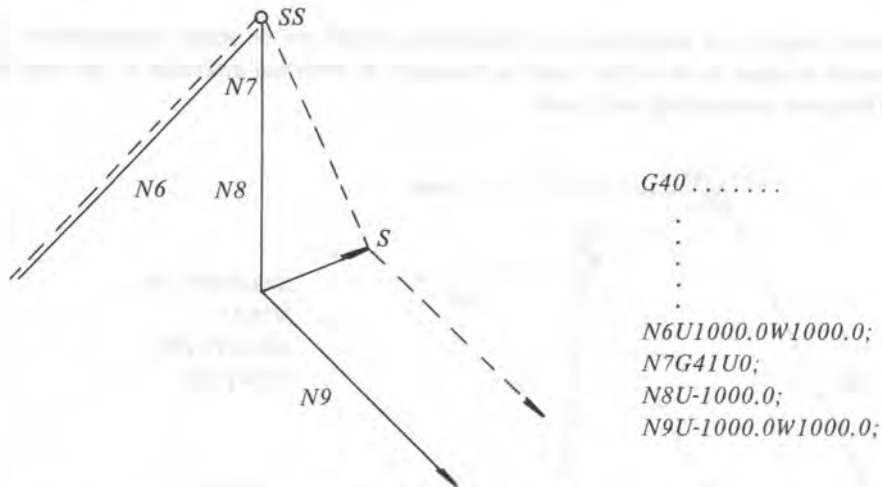
(9) A block not specifying tool movement

The following blocks specify no tool movement. In these blocks, a tool will not move even if tool nose radius compensation is actuated.

- ① M05 ; M code output
- ② S21 ; S code output
- ③ G04X1000 ; Dwell
- ④ G01U0 ; Move distance is zero
- ⑤ G98 ; G code only
- ⑥ G22X1000000 ; Work area setting
- ⑦ G10P01X10Z20R10Q1 ; Offset value change

(a) When specified at start-up

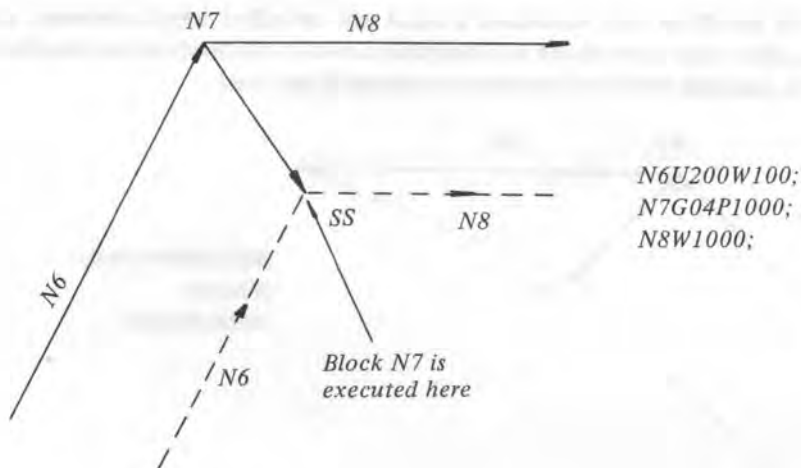
If a block not specifying tool movement is input at start-up, the offset vector is not produced.



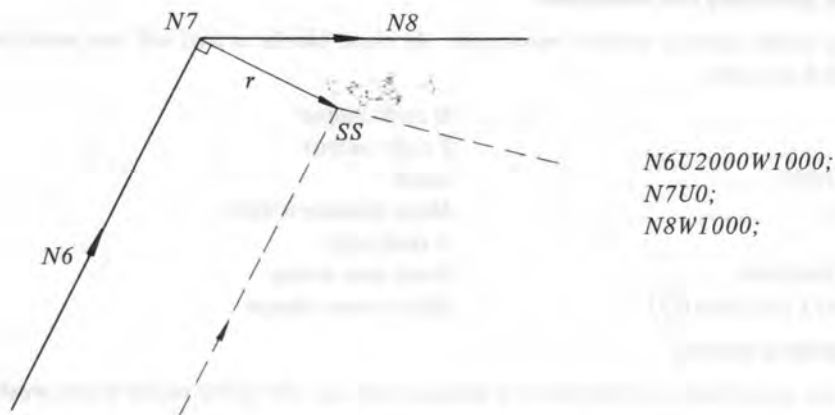
(b) When specified in offset mode

When a block not specifying tool movement is input in the offset mode, the vector and tool nose center path are the same as if the block were not specified. (See Section 6.3.2.3.)

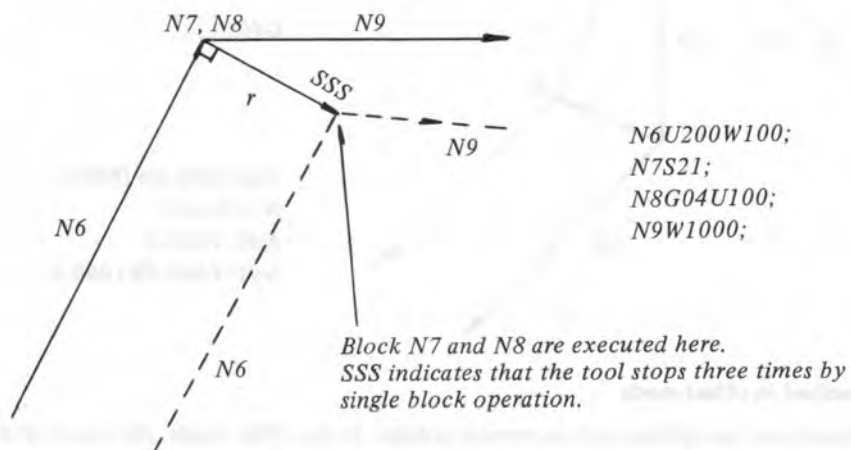
This block is executed at the single block stop point.



However, when a block not specifying tool movement is of type ④ (the move distance is zero), even if the block is specified singly, tool motion is the same as if more than one block not specifying tool movement were input consecutively. (This will be explained later.)

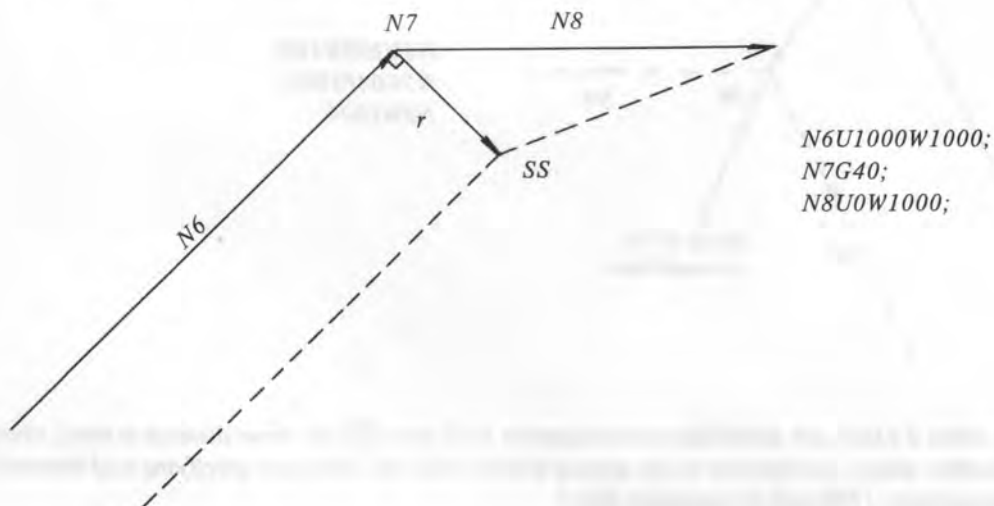


Two or more blocks not specifying tool movement should not be input consecutively. If they are, a vector whose length is equal to the offset value is produced in a normal direction to the tool motion in the earlier block. Therefore, overcutting may result.

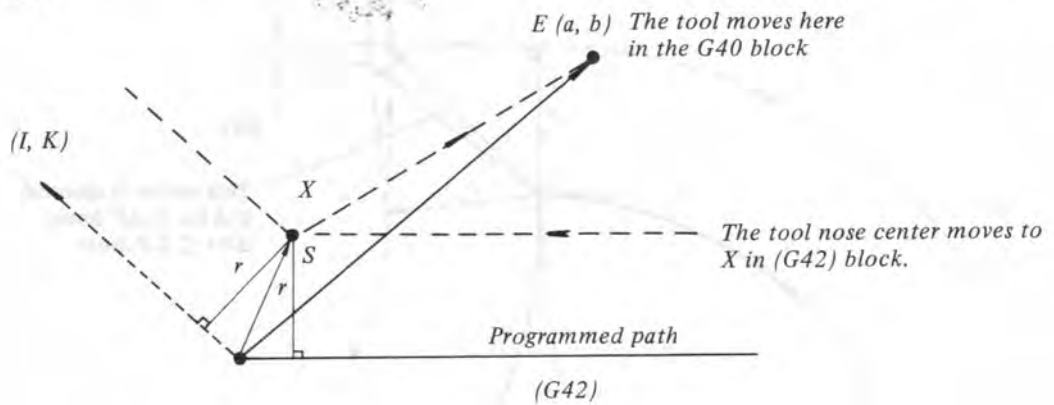


(c) When specified with offset cancel command

When a block not specifying tool movement is input with an offset cancel command, a vector whose length is equal to the offset value is produced in a direction normal to the tool motion specified in the earlier block, and this vector is cancelled when the next move command is executed.



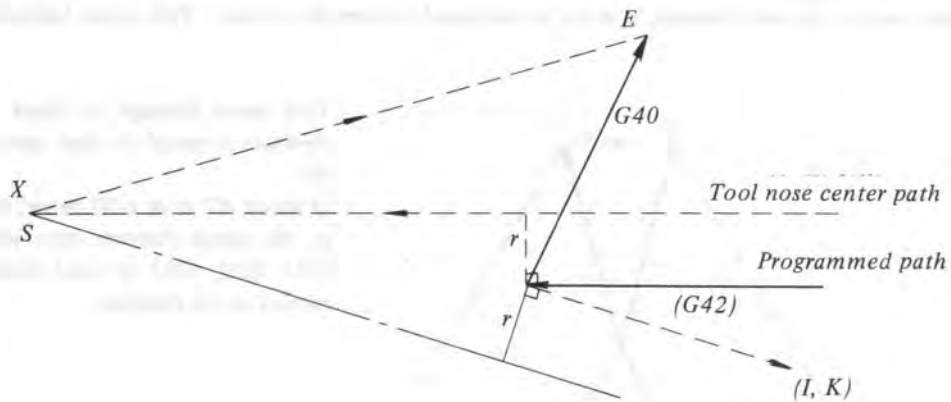
- (10) When a block includes G40 and I ___ K ___; and the mode of the earlier block is G41 or G42 NC assumes that movement from the end point of the earlier block has been specified in the direction of (I, K) (I, K)



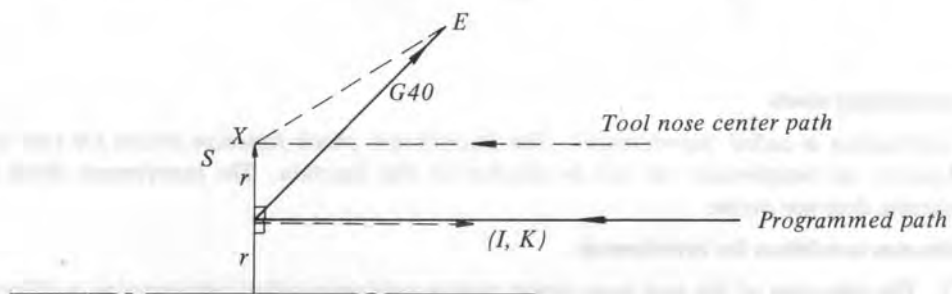
(G42 mode)

G40X a Z b I ___ K ___;

In this case, note that the NC obtains an intersection with the tool path regardless of whether inner or outer wall machining is specified.



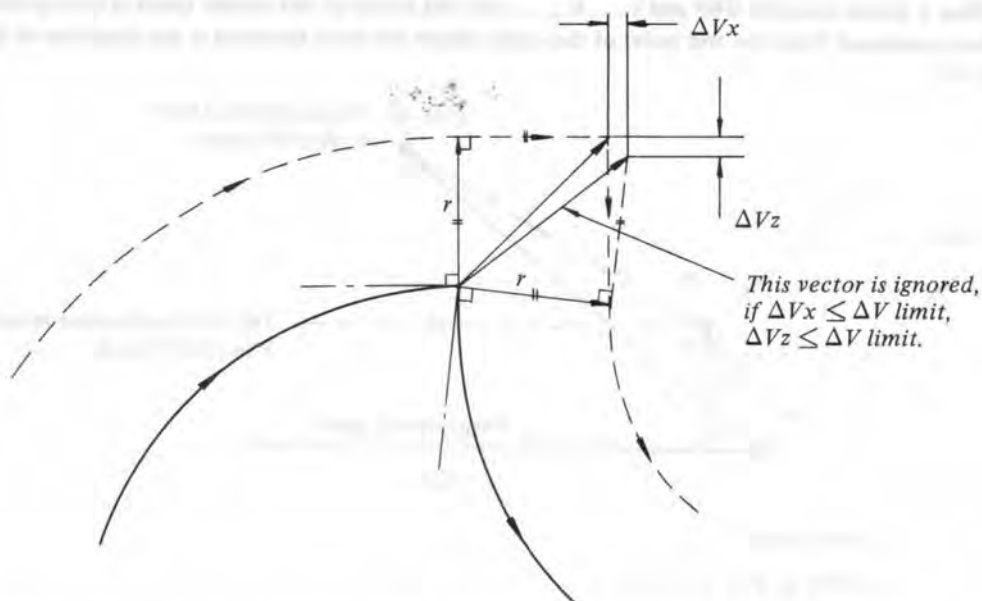
When an intersection can not be obtained, the tool moves to a position normal to the programmed path at the end of the earlier block.



(11) Corner movement

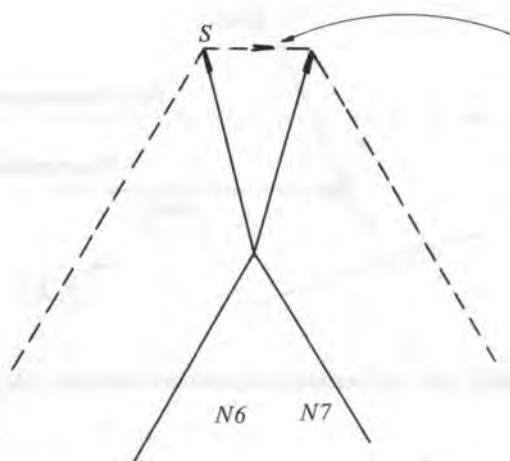
When two or more vectors are produced at the end of a block, the tool moves linearly from one vector to another.

If these vectors almost coincide, the corner movement is not performed and the latter vector is ignored.



If $V_x \leq \Delta V_{\text{limit}}$ and $V_z \leq \Delta V_{\text{limit}}$, the latter vector is ignored. The ΔV_{limit} is preset in the NC unit. The value of ΔV_{limit} is specified by parameter. (CRCDL, parameter number 69)

If these vectors do not overlap, a move is provided to turn the corner. This move belongs to the latter block.



This move belongs to block N7, thus, the feedrate is equal to that specified in block N7.

If block N7 is in G00 mode, the tool moves at the rapid traverse rate, while if it is in G01, G02, G03 or G32 mode, the tool is moved at the feedrate.

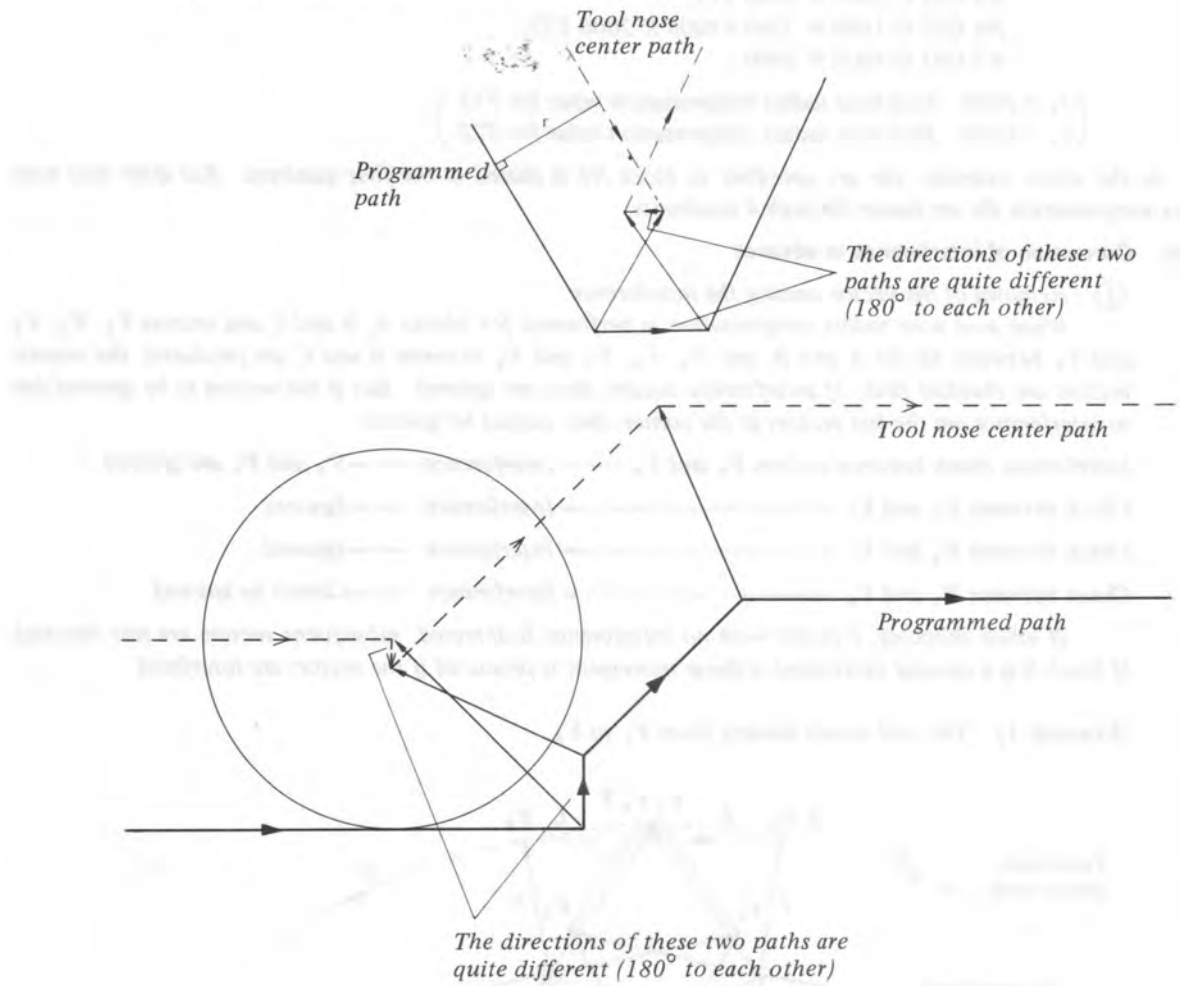
(12) Interference check

Tool overcutting is called 'interference'. The interference check function checks for tool overcutting in advance. However, all interference can not be checked by this function. The interference check is performed even if overcutting does not occur.

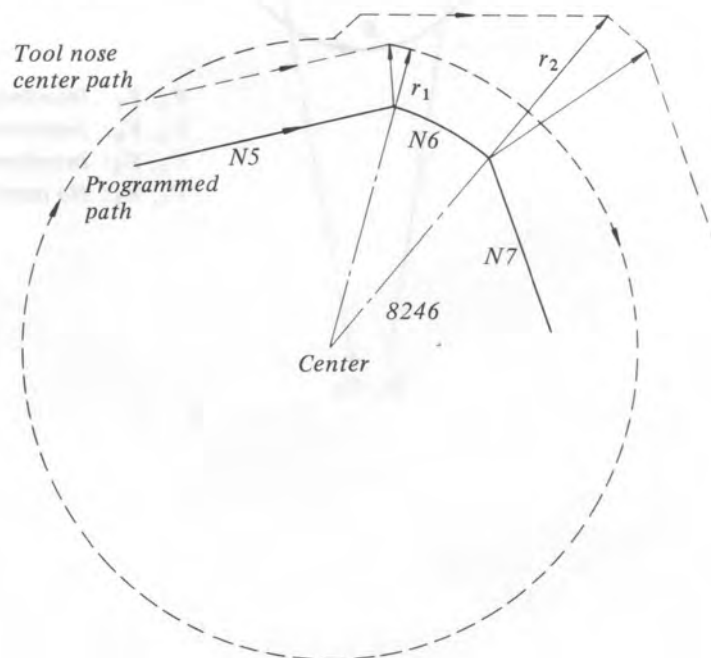
(a) Reference conditions for interference

- ① The direction of the tool nose center path in tool nose radius compensation is different from that of the programmed path. (From 90 degrees to 270 degrees between these paths)
- ② In addition to the above condition, the angle between the start point and end point on the tool nose center path is quite different from that between the start point and end point on the programmed path in circular machining. (More than 180 degrees)

Example of condition ①



Example of condition ②



(G41 mode)
N5 G01 U 2000 W 8000 T11;
N6 G02 U-1600 W 3200 I-8000 K-2000 T22;
N7 G01 U-5000 W 2000 ;

$\left(\begin{array}{l} r_1 = 2000 \text{ Tool nose radius compensation value for T11} \\ r_2 = 6000 \text{ Tool nose radius compensation value for T22} \end{array} \right)$

In the above example, the arc specified in block N6 is placed in the first quadrant. But after tool nose radius compensation the arc passes through 4 quadrants.

(b) Correction of interference in advance

① Removal of the vector causing the interference

When tool nose radius compensation is performed for blocks A, B and C and vectors V_1, V_2, V_3 and V_4 between blocks A and B, and V_5, V_6, V_7 and V_8 between B and C are produced, the nearest vectors are checked first. If interference occurs, they are ignored. But if the vectors to be ignored due to interference are the last vectors at the corner, they cannot be ignored.

Interference check between vectors V_4 and V_5 —→ Interference —→ V_4 and V_5 are ignored

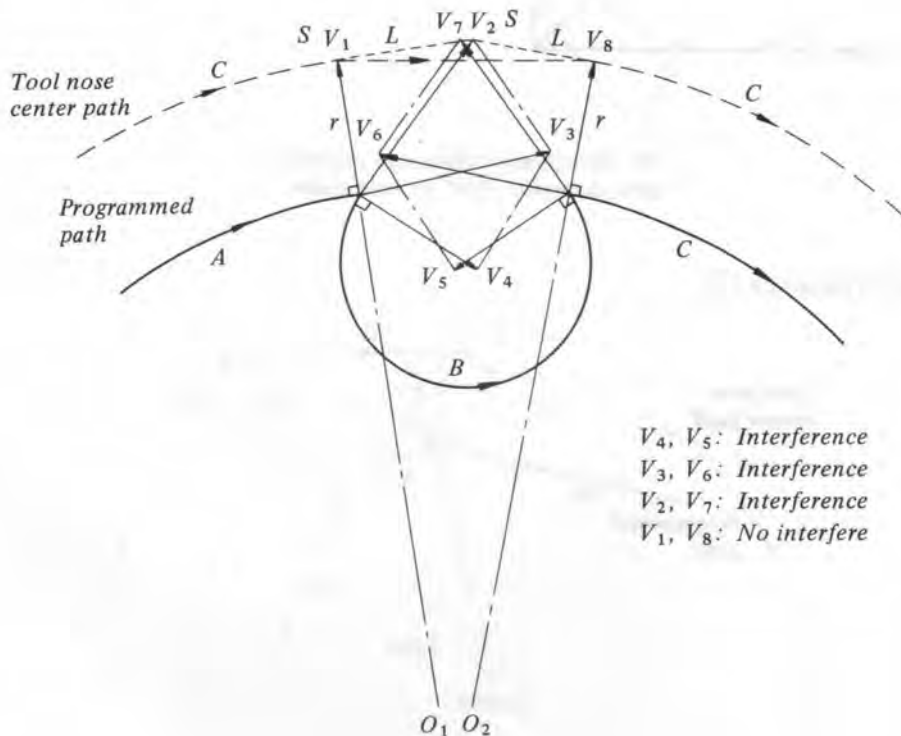
Check between V_3 and V_6 —→ Interference —→ Ignored

Check between V_2 and V_7 —→ Interference —→ Ignored

Check between V_1 and V_8 —→ Interference —→ Cannot be ignored

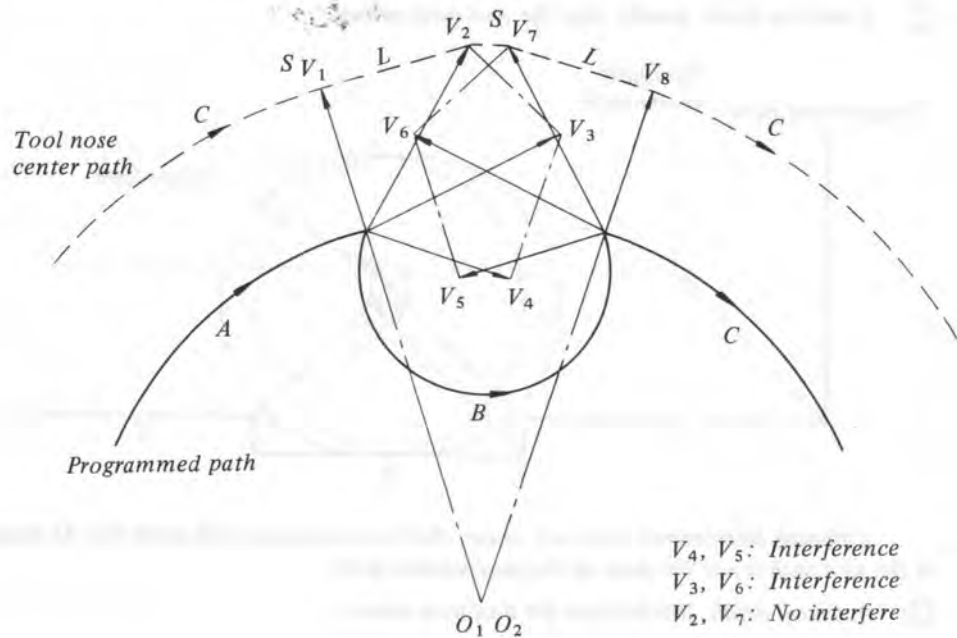
If while checking, a vector with no interference is detected, subsequent vectors are not checked. If block B is a circular movement, a linear movement is produced if the vectors are interfered.

(Example 1) The tool moves linearly from V_1 to V_8 .



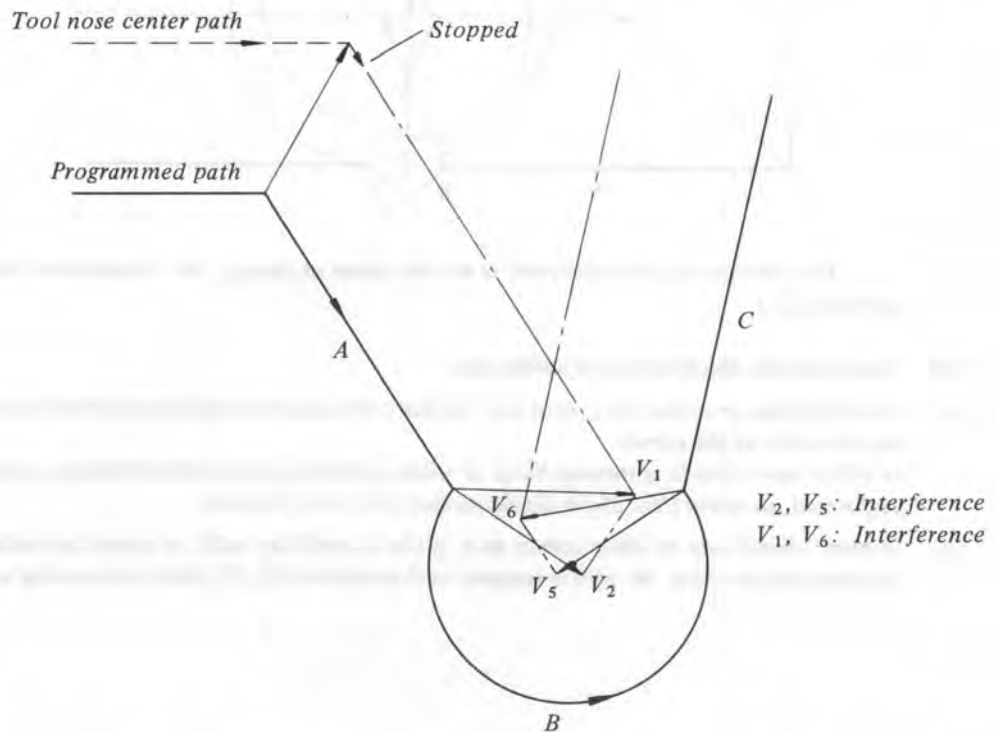
(Example 2) The tool moves linearly as follows:

Tool path: $V_1 \rightarrow V_2 \rightarrow V_7 \rightarrow V_8$



② If the interference occurs after correction ①, the tool is stopped with an alarm.

If the interference occurs after correction ① or if there are only one pair of vectors from the beginning of checking and the vectors interfere, the alarm (No. 41) is displayed and the tool is stopped immediately after execution of the preceding block. (If the block is executed by the single block operation, the tool is stopped at the end of the block.)

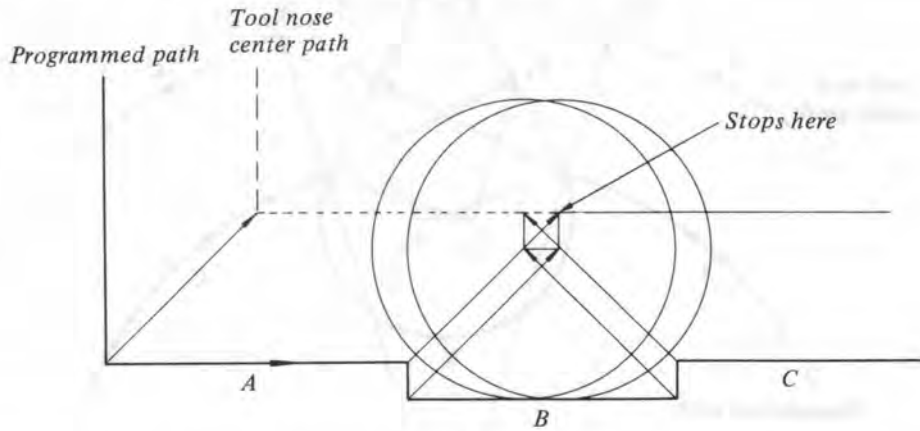


After ignoring vectors V_2 and V_5 because of interference, interference also occurs between vectors V_1 and V_6 . The alarm is displayed and the tool is stopped.

(c) Checking is performed although interference does not actually occurs.

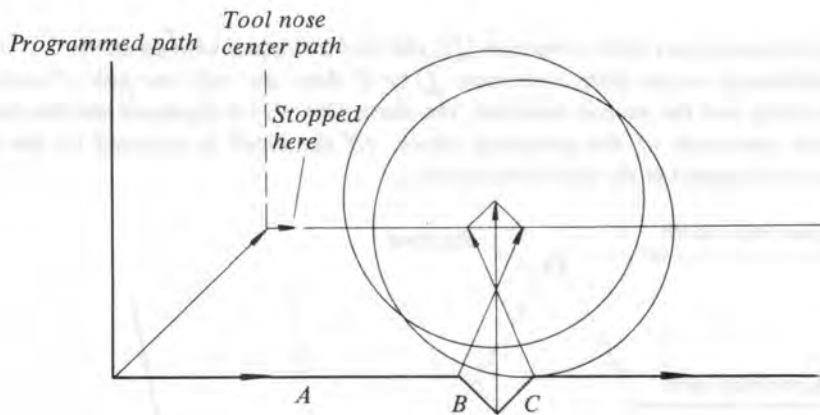
There are many examples including the following:

- ① A shallow depth, smaller than the tool nose radius.



Although interference does not occur, the tool is stopped with alarm No. 41 because the direction of the tool path is not the same as the programmed path.

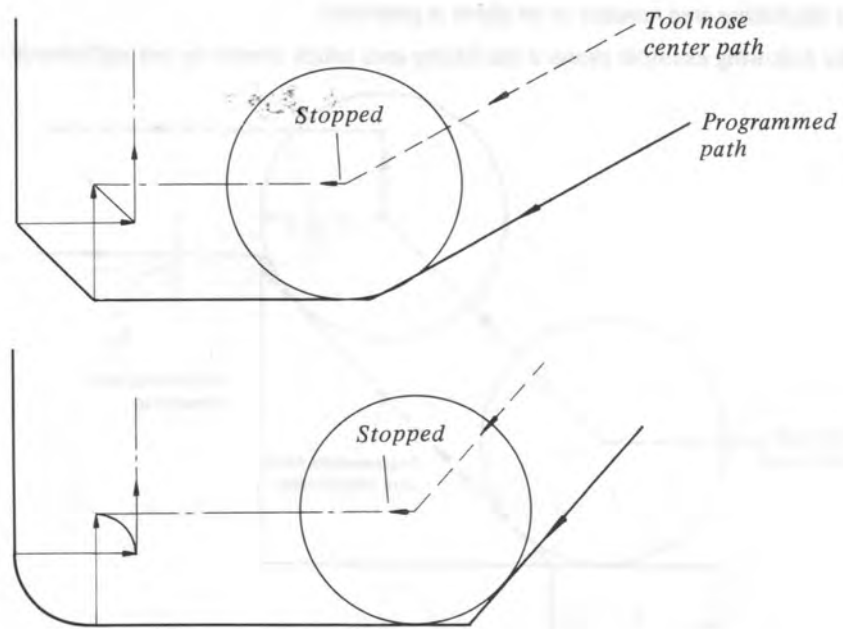
- ② A groove depth, smaller than the tool nose radius.



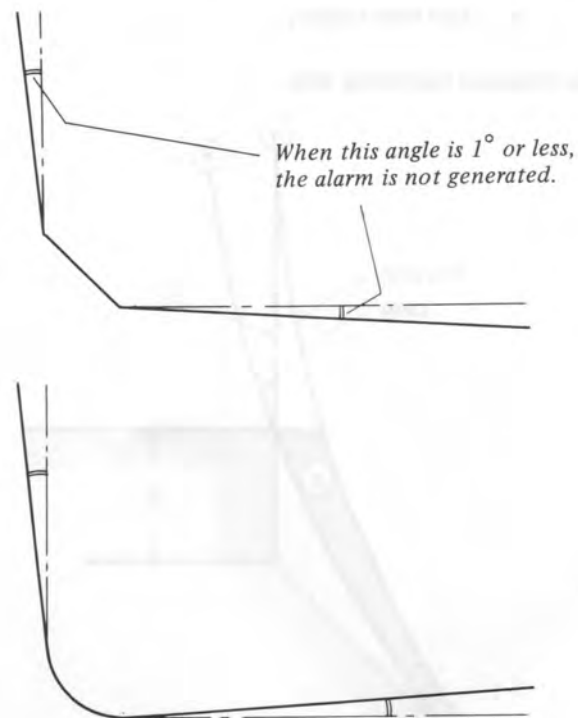
The direction of the tool path is not the same as that of the programmed path. (The same as in example ①.)

(13) Correction in chamfering and corner arcs

- (a) In chamfering or corner arcs, tool nose radius compensation only be performed when an ordinary intersection exists at the corner.
In offset cancel mode, a start-up block or when exchanging the offset direction, compensation cannot be performed, an alarm (No. 39) is displayed and the tool is stopped.
- (b) In inner chamfering or inner corner arcs, if the chamfering value or corner arc value is smaller than the tool nose radius value, the tool is stopped with an alarm (No. 39) since overcutting will occur.

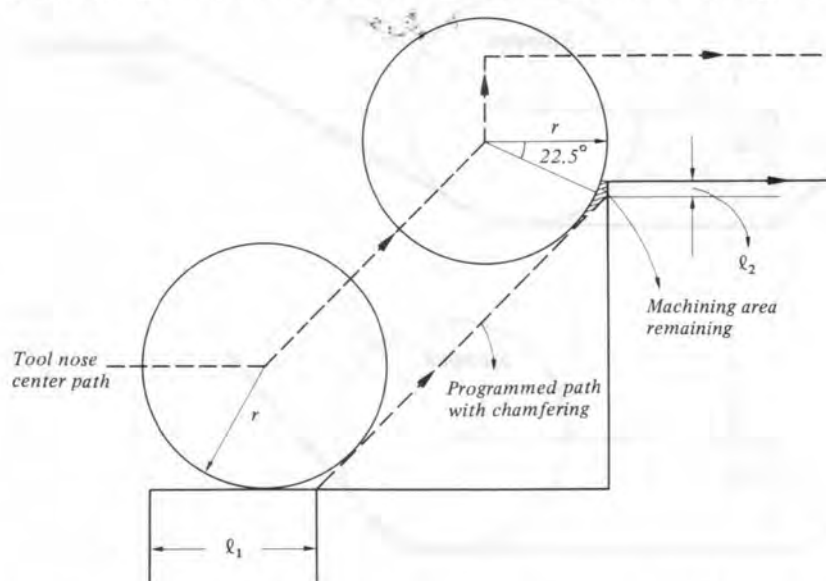


- (c) The valid inclination angle of the programmed path in the blocks before and after the corner is 1 degree or less so that the alarm (No. 52, 54) generated by the calculating error of tool nose radius compensation does not occur.



(d) When machining area remains or an alarm is generated.

(i) The following example shows a machining area which cannot be cut sufficiently.

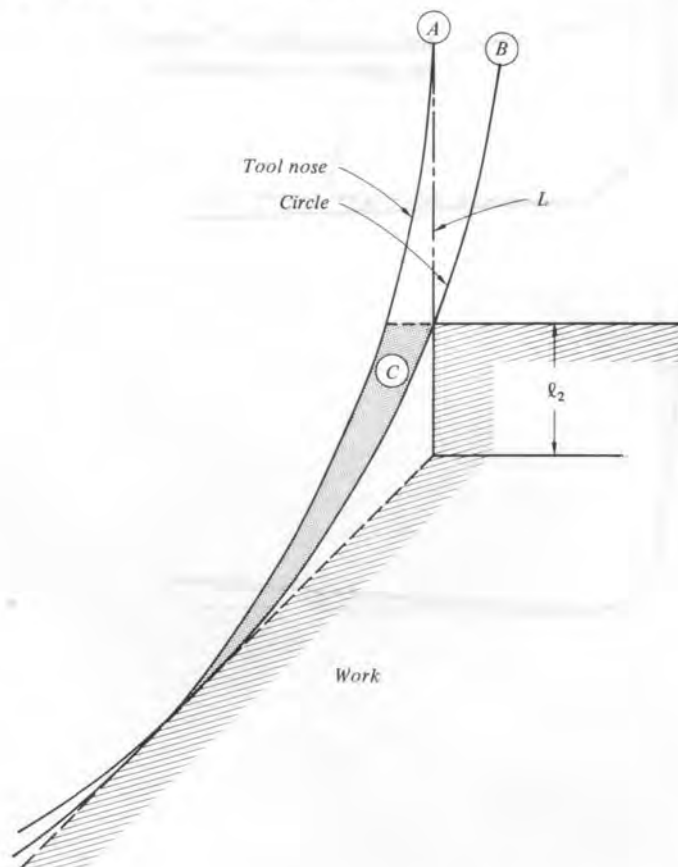


In inner chamfering, if the portion of the programmed path that is not a part of the chamfering (in the above figure l_1 or l_2) is in the following range, insufficiently cut area will exist.

$$0 \leq l_1 \text{ or } l_2 < r \tan 22.5^\circ$$

(r : tool nose radius.)

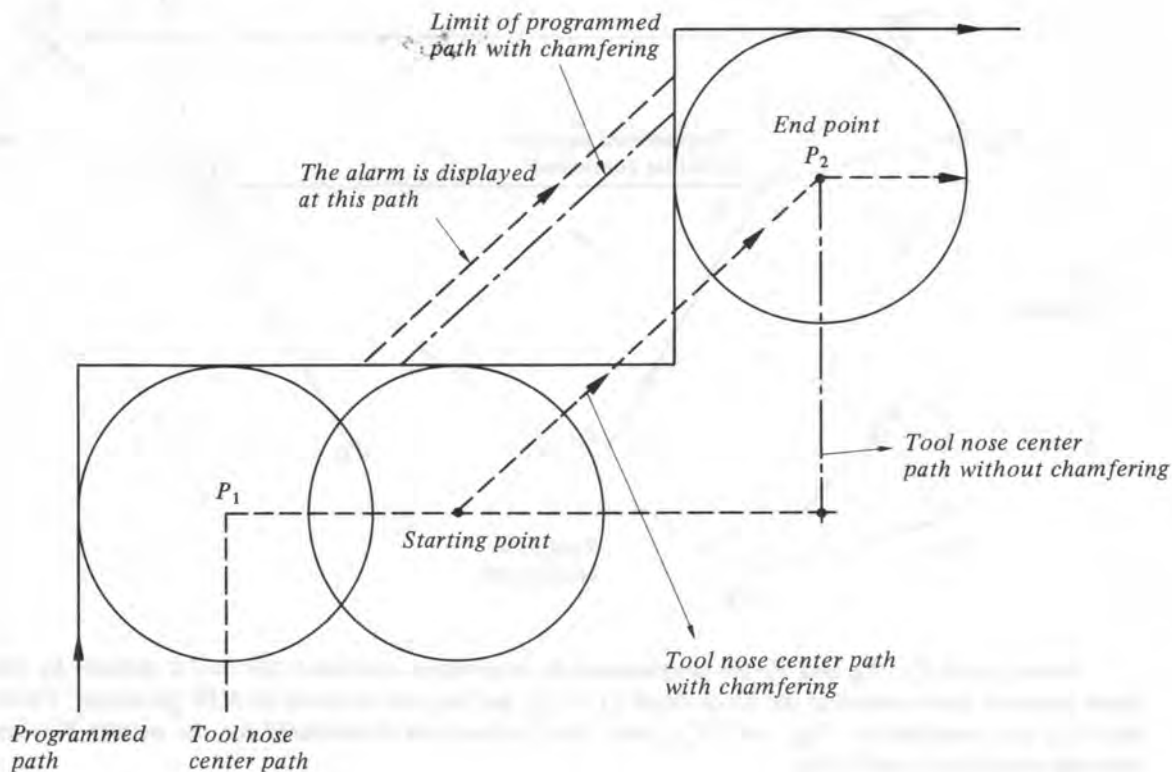
Enlarged view of the remained machining area.



Although the tool should be positioned at (B) in the above figure, the tool is positioned at (A) (the tool nose is tangent to line L).

Thus, area (C) is not machined.

(ii) Alarm No. 52 or 55 is displayed in the following cases.



In outer chamfering with an offset, a limit is imposed on the programmed path.

In the above example, when the start or the end point of the tool nose center path during chamfering coincides with the intersection points (P_1 or P_2) without chamfering respectively, outer chamfering is limited.

If the chamfering value is more than the limit value as specified, alarm No. 52 or 55 will be displayed.

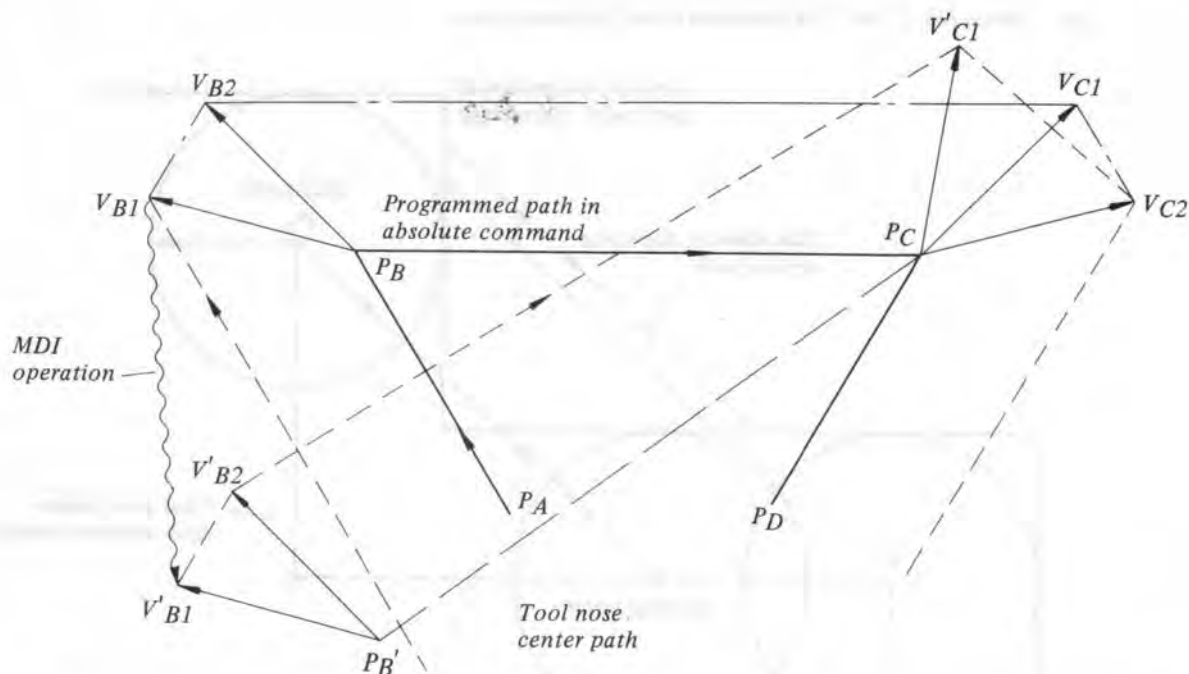
(14) Input command from the MDI

Compensation is not performed for commands input from the MDI.

However, when automatic operation using the NC tape composed of absolute commands is temporarily stopped by the single block function, MDI operation is performed, then automatic operation starts again, the tool path is as follows:

In this case, the vectors at the start point of the next block are translated and the other vectors are produced by the next two blocks.

Therefore, from point P_c , compensation is accurately performed.



When points P_A , P_B and P_C are programmed in an absolute command, the tool is stopped by the single block function after executing the block from P_A to P_B and the tool is moved by MDI operation. Vectors V_{B1} and V_{B2} are translated to V'_{B1} , and V'_{B2} and offset vectors are recalculated for the vectors V_{C1} and V_{C2} between block P_B - P_C and P_C - P_D .

However, since vector V'_{B2} is not calculated again, compensation is accurately performed from point P_C .

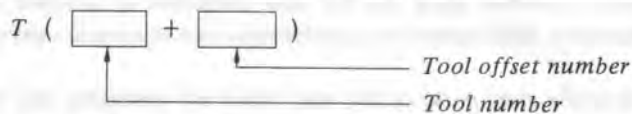
(15) Manual operation

When manual operation is to be performed in tool nose radius compensation mode, see note of Chapter IV Section 4.4.4.

(16) General precautions for offset operations

(a) Offset amount specification

Numbers specified after address T have the following meanings.



Either format of $T(2+1)$ or $T(2+2)$ can be set. The offset value can be specified by the tool offset number in the lower 1 or 2 digits of the T code. The offset value (tool nose radius) corresponding to the tool offset number is used in this case.

Once specified, offset numbers remain valid until another number is specified or the offset mode is cleared.

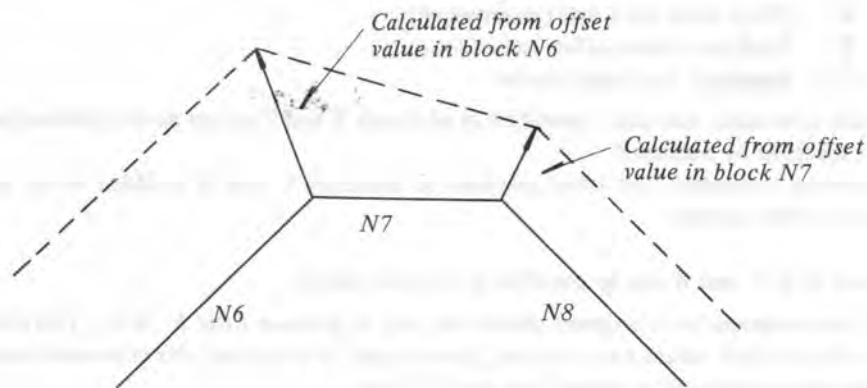
In addition to indicating the offset value for tool nose radius compensation, the offset number is used to specify the offset amount used for tool offset.

See Chapter III Section 6.1.

(b) Changing the offset value

In general the offset value is changed in cancel mode, or when changing tools. If the offset value is changed in offset mode, the vector at the end point of the block is calculated for the new offset value.

The imaginary tool number and tool offset number are changed in the same way.



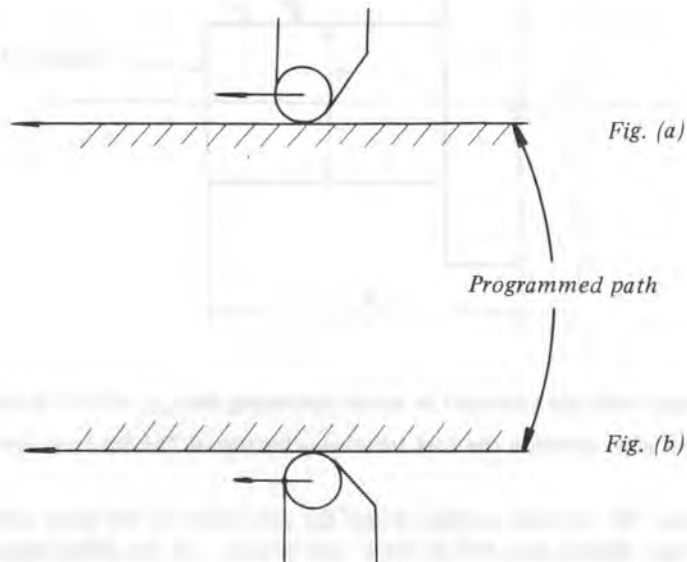
When some vectors are produced between blocks N6 and N7, the vector at the end point of the present block is calculated using the offset value of the block N6.

(c) The polarity of the offset amount and the tool nose center path.

When a negative offset value is specified, the program is executed for the figure which is created by exchanging G41 for G42 or G42 for G41 in the process sheet.

A tool machining an inner wall will machine the outer profile, and tool machining the outer profile will machine the inner wall.

An example is shown below. In general, NC machining is programmed assuming a positive offset value. When a program specifies a tool path as shown in (a), the tool will move as shown in (b) if a negative offset is specified. The tool in (b) will move as shown in (a) when the sign of the offset value is reversed.



(Note) When the sign of the offset value is reversed, the offset vector of the tool nose is reversed but the imaginary tool nose direction does not change.

Therefore, do not reverse the sign of the offset value when starting the machining meeting the imaginary tool nose to the start point.

6.5 Input of Offset Value by Programming (G10)

Offset values can be input by a program using the following command:

G10P_____X_____Z_____R_____Q_____; or

G10P_____U_____W_____R_____Q_____;

P : Offset number

X : Offset value on X axis (absolute)

Z : Offset value on Z axis (absolute)

U : Offset value on X axis (incremental)

- W : Offset value on Z axis (incremental)
 R : Tool nose radius offset value (absolute)
 Q : Imaginary tool nose number

In an absolute command, the values specified in addresses X and Z are set as the offset value corresponding to the offset number specified by address P.

In an incremental command, the value specified in addresses U and W is added to the current offset value corresponding to the offset number.

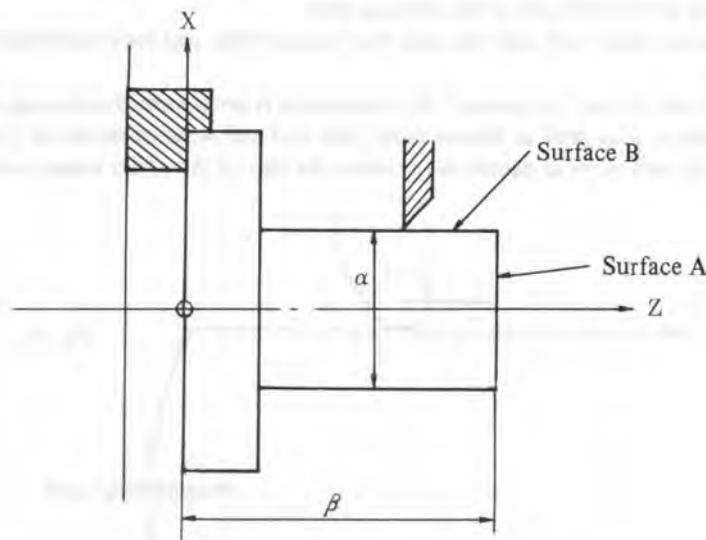
(Note 1) Addresses X, Z, U and W can be specified in the same block.

(Note 2) Use of this command in a program allows the tool to advance little by little. This command can also be used to input offset values one at a time from a tape by specifying this command successively instead of inputting these values one at a time from the MDI unit.

6.6 Direct Input of Measured Offset Value

The following convenient method can be used to set the offset value as the difference between a standard point, such as the tip of the standard tool or the turret center, and the tip of the tool actually used.

A work coordinate system is assumed to have been set in advance. (See Sections 3.5 and 5.8.)



- (1) Cut the workpiece with the tool used in actual machining along surface A in manual mode.
- (2) Release the tool after pressing the tool measuring button on the machine operator's panel and stop spindle rotation.
- (3) Measure distance "β" between surface A and the zero point in the work coordinate system, and set it to the offset number desired plus 100 as the Z offset value. (If the offset value is to be set to offset number 03, set it to offset number 103.)
- (4) Cut the workpiece along surface B in manual mode.
- (5) Release the tool after pressing the tool measuring button on the machine operator's panel and stop spindle rotation.
- (6) Measure distance "α" and set it to the offset number desired plus 100 as the X offset value.

Repeat the above operation for each tool used.

If the coordinate value of the tool at surface B is 105.0, the measured distance α is 104.0 and 104.0 is set to offset number 103 of address X, the offset value 1.0 is automatically set to offset number 03 of address X.

(Note 1) The distance "α" should always be set as a diameter value.

6.7 Automatic Tool Offset

When a tool is moved to the measurement position by execution of a command given to NC, the NC automatically measures the difference between the current coordinate value and the coordinate value of the command measurement position and uses it as the offset amount for the tool. When the tool is moved to the measurement position in offset mode and an amount to be further offset is generated as the result of this measurement, the current offset amount is further offset.

(1) Coordinate system

When moving the tool to a position for measurement, the coordinate system must be set in advance. G50XxZz; sets the coordinate system (the work coordinate system for programming is used in common).

(2) Movement to measurement position

A movement to a measurement position is performed by specifying G36X x_a or G37Z z_a in the MDI, Tape, or Memory mode. In this case, the measurement position should be x_a or z_a (absolute command).

Execution of this command moves the tool at the rapid traverse rate toward the measurement position, lowers the feedrate halfway, then continues to move it until the approach end signal from the measuring instrument is issued. When the tool tip reaches the measurement position, the measuring instrument sends a signal to the NC which stops the tool.

(3) Offset

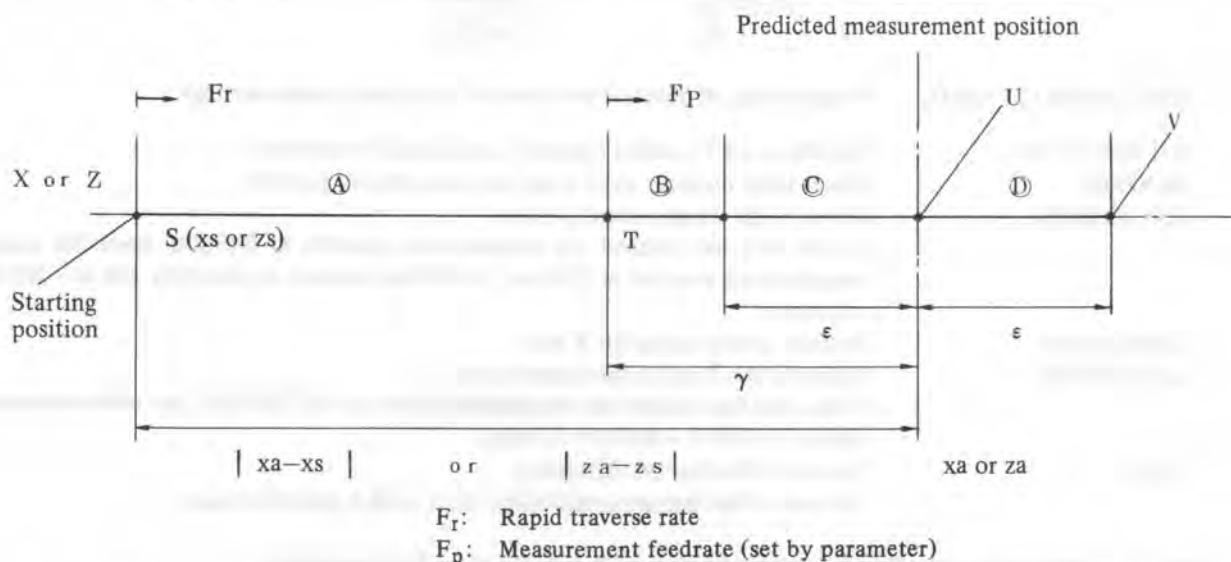
The current tool offset amount is further offset by the difference between the coordinate value (α or β) when the tool has reached the measurement position and the value of x_a or z_a specified in G36X x_a or G37Z z_a .

Offset amount x = current offset amount x + ($\alpha - x_a$)

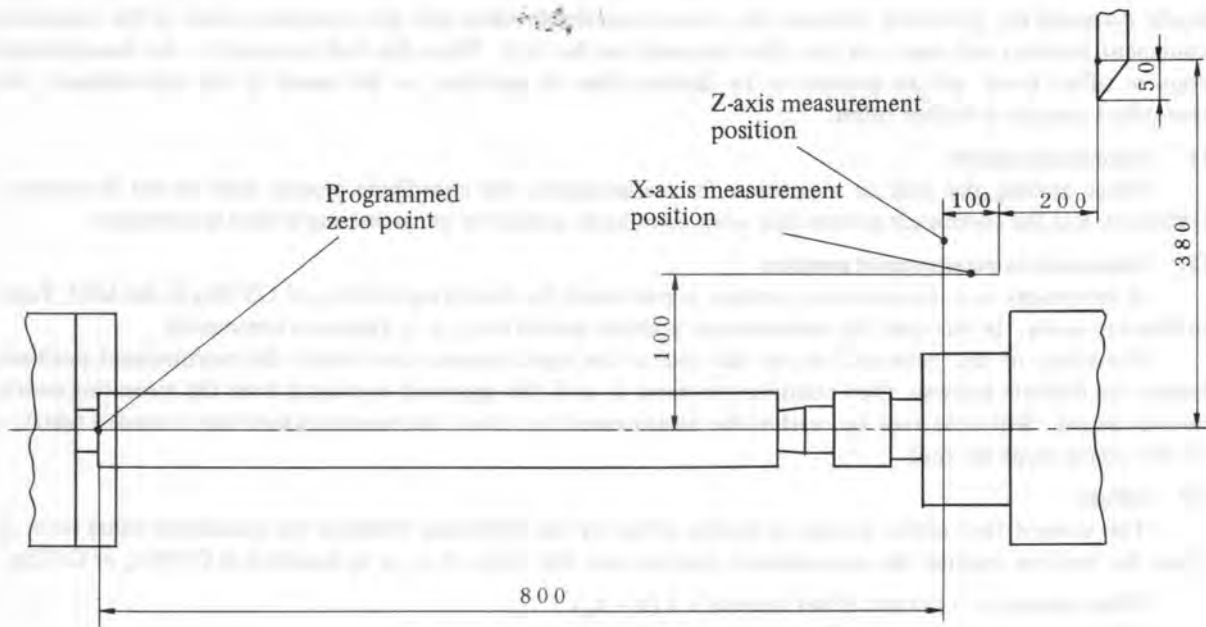
Offset amount z = current offset amount z + ($\beta - z_a$)

These offset amounts can also be altered from the MDI keyboard.

(4) Feedrate and alarm



The tool, when moving from the starting position toward the measurement position predicted by x_a or z_a in G36 or G37, is fed at the rapid traverse rate across area (A). Then the tool stops at point T ($x_a - \gamma_x$ or $z_a - \gamma_z$) and moves at the measurement feedrate set by parameter across areas (B), (C), and (D). If the approach end signal turns on during movement across area (B), alarm 80 is generated. If the approach end signal does not turn on before point V, the tool stops at point V and alarm 80 is generated.



	Offset amount (Before measurement)	Offset amount (After measurement)
X	100000	98000
Z	0	4000

G50 X760000 Z110000;

Programming of absolute zero point (Coordinate system setting)

S01 M03 T0101;

Specifies tool T1, offset number 1, and spindle revolution

Z850000;

Moves some distance away from the measurement position

G36 X200000;

Moves to the measurement position.

(If the tool has reached the measurement position at X19800; since the correct measurement position is 200 mm, the offset amount is altered by $198.0 - 200.0 = -2.0$ mm.)

G00X204000;

Retracts a little along the X axis.

G37 Z800000;

Moves to the Z-axis measurement position.

(If the tool has reached the measurement position at Z804000; the offset amount is altered by $804.0 - 800.0 = 4.0$ mm.)

T0101;

Further offsets by the difference.

The new offset becomes valid when the T code is specified again.

(Note 1) When there is no T-code command before G36 or G37, alarm 81 is generated.

(Note 2) When a T code is specified in the same block as G36 or G37, alarm 82 is generated.

(5) Offset number

There are 16 offset numbers (01 to 16). This can be optionally extended to 32.

(Note 3) Measurement speed, γ , and ϵ are set as parameters. These values have already been set by machine tool builder.

(Note 4) ϵ and γ must be positive numbers such that $\gamma > \epsilon$.

(Note 5) The approach end signal is monitored every 2 ms. The following measuring error is generated:

$$ERR_{\max.} = Fm \times 1/60 \times 2/1000$$

where

$ERR_{\max.}$: maximum measuring error (mm)

Fm : measurement feedrate (mm/min.)

When $F_m = 100 \text{ mm/min.}$, for example, $ERR_{\max.} = 0.003 \text{ m}$

(Note 6) Cancel the tool nose radius compensation before G36 or G37 is specified.

(Note 7) When a manual movement is inserted into a movement at a measurement feedrate, return the tool to the position before the inserted manual movement for restart.

(Note 8) The tool stops a maximum of 16 ms after the approach end signal is detected. But the value of the position when the approach end signal was detected (not the value when the tool stopped) is used for α and β (see (3)) used to determine the offset amount. The overrun for 16 ms is:

$$Q_{\max.} = F_m \times 1/60 \times 16/1000$$

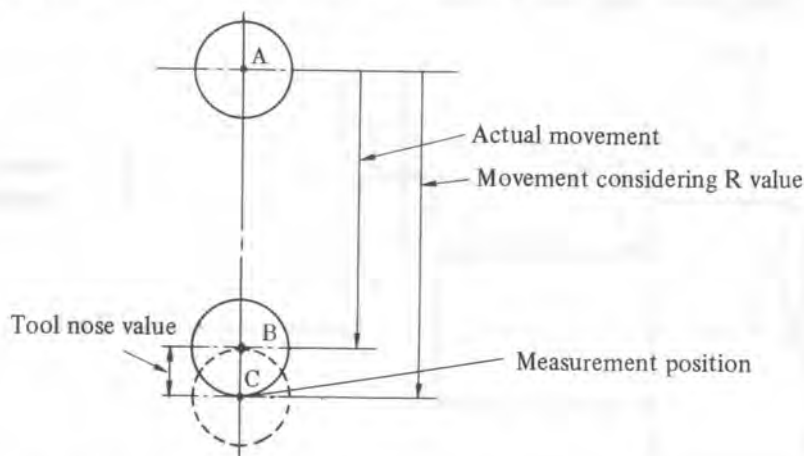
where

$Q_{\max.}$: maximum overrun (mm)

F_m : measurement feedrate (mm/min)

(Note 9) When using the optional tool nose radius compensation function, the tool offset amount is determined considering the value of tool nose R. Make sure that tool nose radius value is set correctly.

Example: When the tool nose center coincides with the start point.



The tool actually moves from point A to point B, but the tool offset amount is determined assuming that the tool moves to point C considering the R value.

(Note 10) When using this function, set parameter 008-TLCC to 0.

TCLL 0: The offset amount, when altered, becomes valid with specification of the next T code.

6.8 External Tool Compensation

This function changes the tool compensation value from external devices. When inputting the tool compensation value from an external device, the input data is added to the offset value of the offset number specified in the program. If so specified by an input signal, the input data can be made the offset value.

If the machine is installed with an automatic measuring instrument for tools and workpiece, offset values can be corrected by inputting errors from the correct values to the NC using this function.

Since programs, operations, functions and limitations differ with the machine tool builder, be sure to operate the tool according to the manual issued by machine tool builder.

(Note 1) If a tool compensation value is input from an external device when offset number 0 has been specified in a program, that is when offset is cancelled, the work coordinate system may be shifted by the value input depending on the machine tool. Refer to the machine tool manual.

7. MACHINING CYCLE FUNCTION

For repetitive machining peculiar to turning, such as metal removal in rough cutting, a series of paths usually specified in a range of three to several dozen blocks, can be specified in one block. In addition, only the values to be exchanged need to be specified for repetition, and programs using this cycle are very simple and useful.

The machining cycle is usually one of the following two types.

- (i) Canned cycle
- (ii) Multiple repetitive cycle

Note that the drawings in the examples below are for diameter programming. In radius programming, change parameters $U/2$ or $X/2$ to U or X respectively.

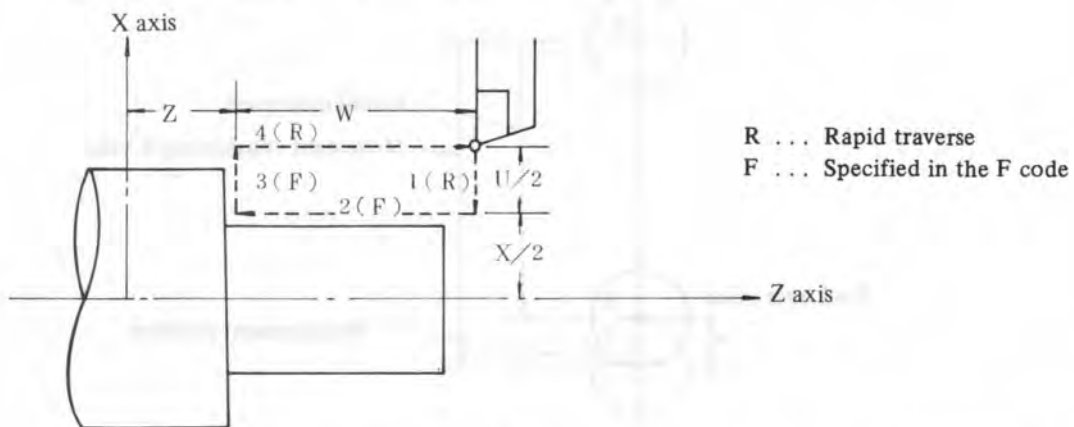
7.1 Canned Cycle (G90, G92, G94)

There are three canned cycles, G90, G92, and G94.

7.1.1 Cutting cycle A, G90

- (1) The straight cutting cycle is specified in the following command.

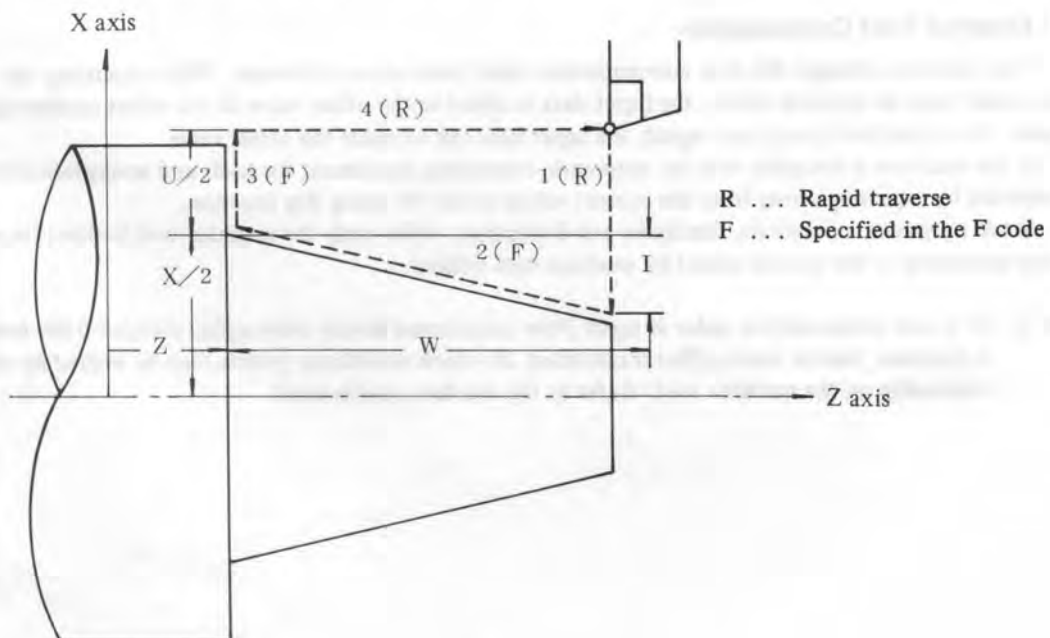
$G90X(U) \text{ --- } Z(W) \text{ --- } F \text{ --- } ;$



In incremental programming, the sign of the numbers following address U and W depend on the direction of paths 1 and 2. In the above example, the signs of U and W are negative. In single block mode, operations 1, 2, 3 and 4 are performed by pressing the cycle start button once.

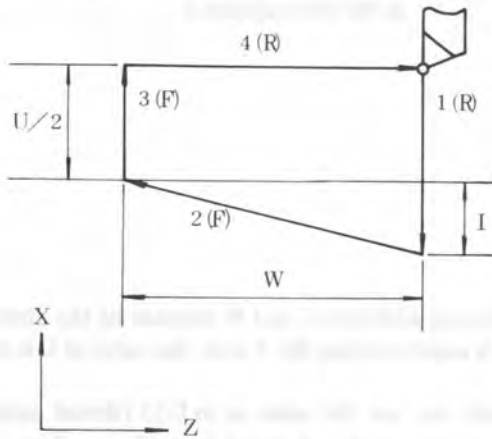
- (2) The taper cutting cycle is specified in the following command.

$G90X(U) \text{ --- } Z(W) \text{ --- } I \text{ --- } F \text{ --- } ;$

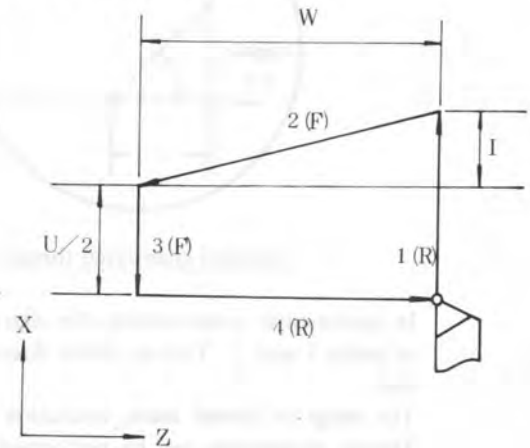


In incremental programming, the relationship between the signs of the numbers following addresses U, W and I, and tool paths is as follows:

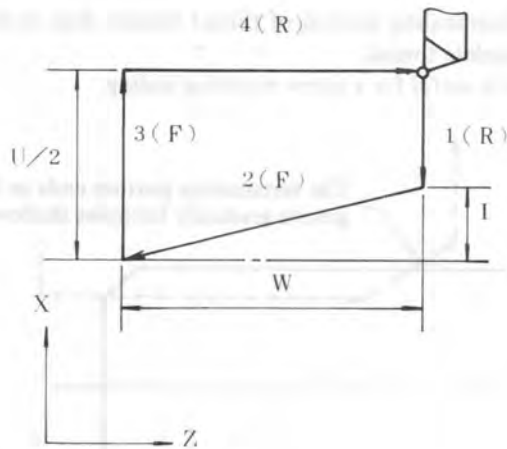
(i) $U < 0, W < 0, I < 0$



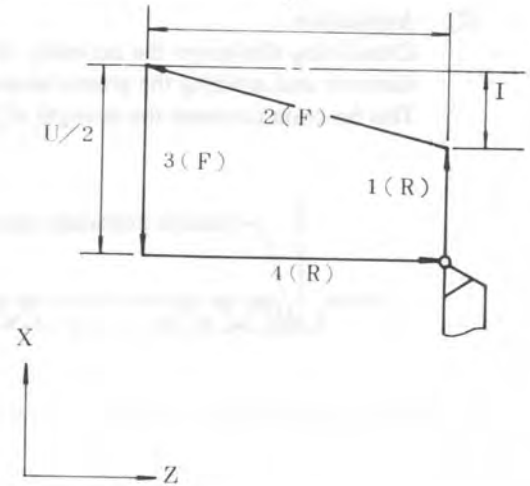
(ii) $U > 0, W < 0, I > 0$



(iii) $U < 0, W < 0, I > 0$ at $|I| \leq \frac{|U|}{2}$



(iv) $U > 0, W < 0, I < 0$ at $|I| \leq \frac{|U|}{2}$

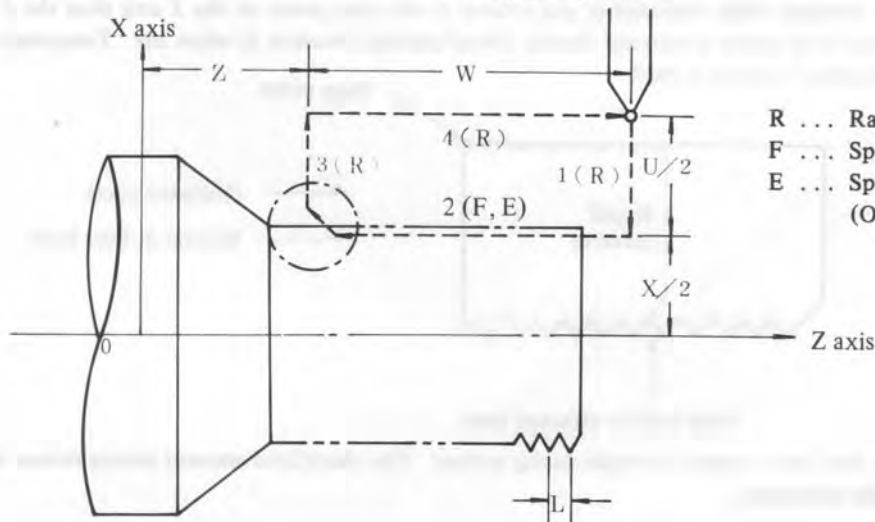


7.1.2 Thread cutting cycle, G92

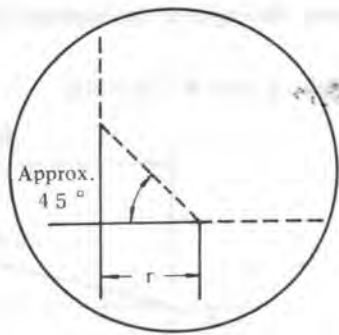
(1) Straight thread cutting is specified in the following command.

G92 X(U) — Z(W) — $\left\{ \begin{array}{l} F \text{ — } \\ E \text{ — } \end{array} \right\}$;

Lead (L) is specified.



R ... Rapid traverse
F ... Specified in the F code
E ... Specified in the E code (Option)



(The chamfered angle in the left figure is 45 degrees or less because of the delay in the servo system.)

Detailed chamfered thread

In incremental programming, the sign of numbers following addresses U and W depends on the direction of paths 1 and 2. That is, if the direction of path 1 is negative along the X axis, the value of U is negative.

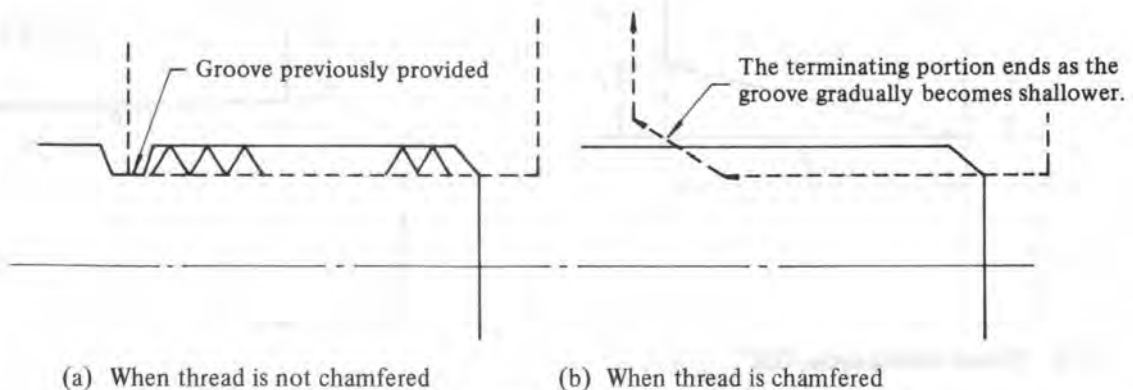
The range of thread leads, limitation of spindle speed, etc. are the same as in G32 (thread cutting). Thread chamfering can be performed in this thread cutting cycle. A signal from the machine tool, initiates thread chamfering. The chamfering distance is specified in a range from 0.1L to 3.1L in 0.1L increments by parameter. (In the above expression, L is the thread lead.)

In the single block mode, operations 1, 2, 3 and 4 are performed by pressing the cycle start button once.

(2) Application

Chamfering eliminates the necessity of making the terminating portion of thread thinner than its inner diameter and avoiding the groove because of an incomplete thread.

This function increases the strength of the screw, and is useful for a screw requiring sealing.



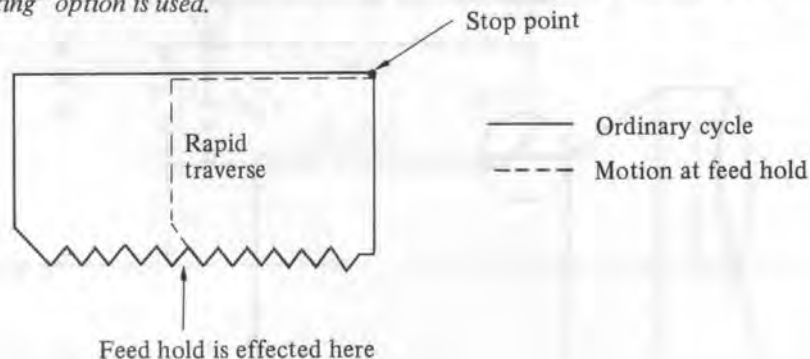
(a) When thread is not chamfered

(b) When thread is chamfered

(Note 1) Notes on thread cutting are the same as for thread cutting in G32.

However, a stop by executing a feed hold during cycle 2 causes the tool to stop after completion of rapid traverse motion on path 3 of the thread cutting cycle.

(Note 2) The tool retreats while chamfering and returns to the start point on the X axis then the Z axis, as soon as the feed hold status is entered during thread cutting (motion 2) when the "Temporary Stop during Thread Cutting" option is used.

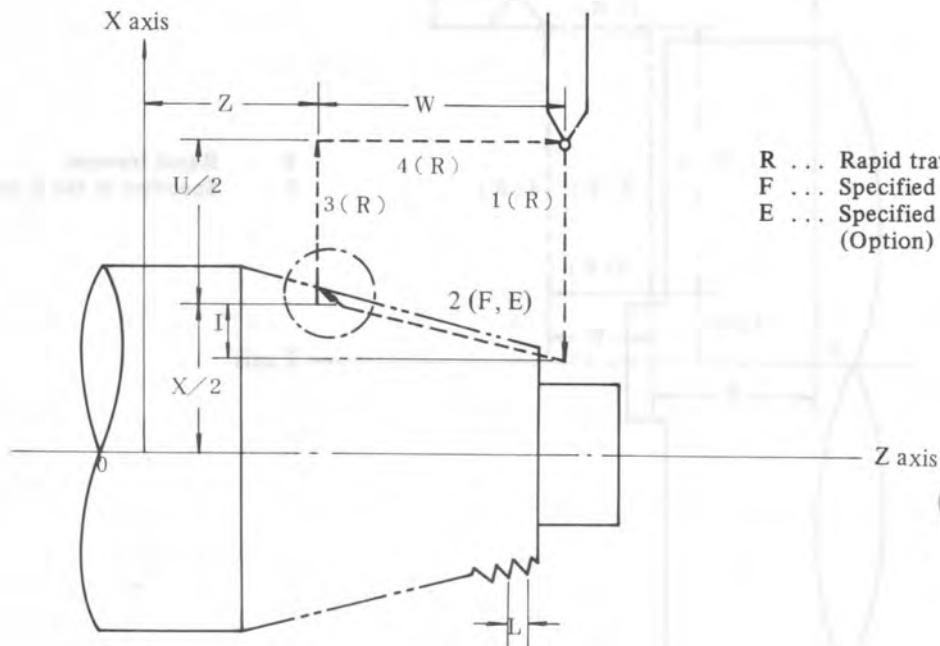


Another feed hold cannot be made during retreat. The chamfered amount during retreat is the same as that at the end point.

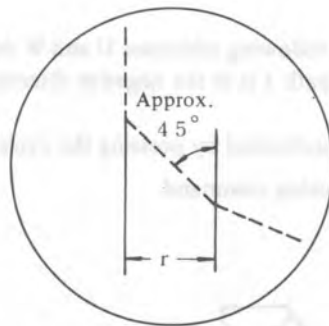
(3) Taper thread cutting is specified in the following command.

G92X(U) — Z(W) — I — { F — }
E — };

Lead (L) is specified.



R ... Rapid traverse
F ... Specified in the F code
E ... Specified in the E code
(Option)



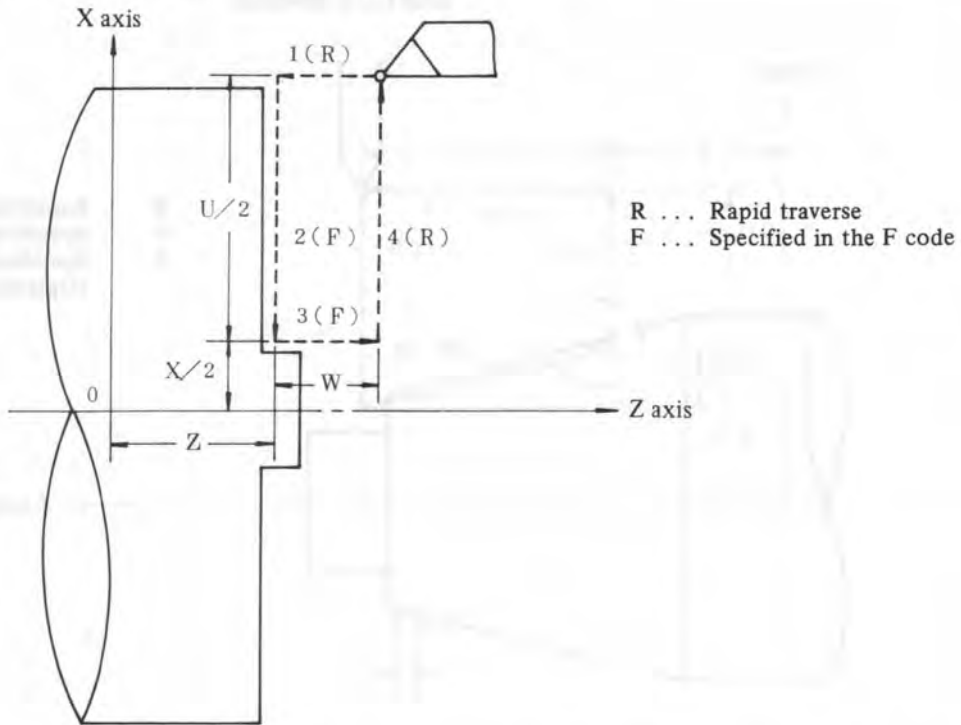
Detailed chamfered thread

(The chamfered angle in the left figure is 45 degrees or less because of the delay in the servo system.)

7.1.3 Cutting cycle B, G94

- (1) The face cutting cycle is specified in the following command.

G94X(U) — Z(W) — F — ;

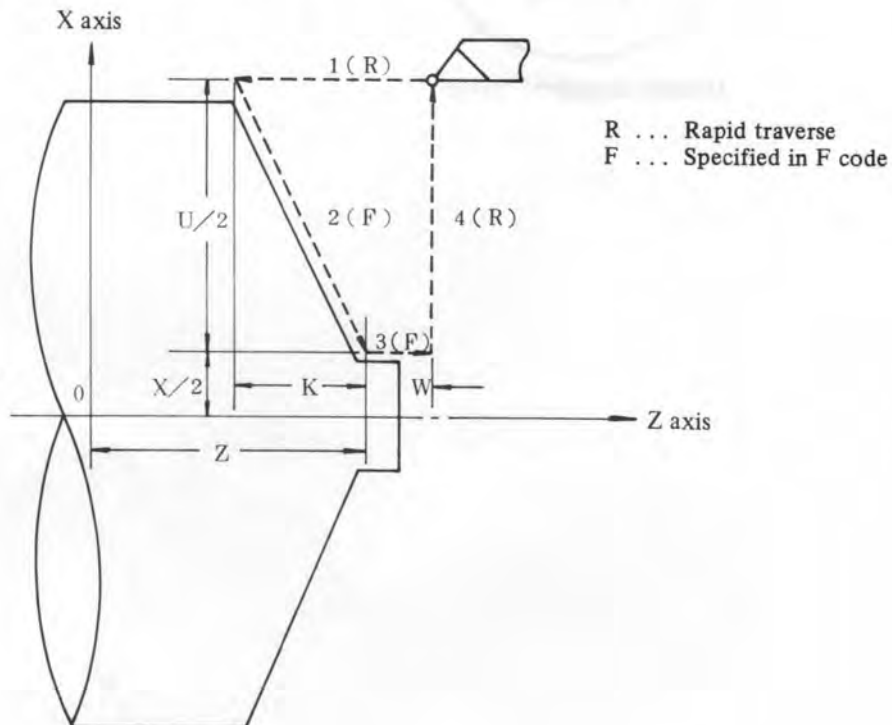


In incremental programming, the sign of numbers following addresses U and W depends on the direction of paths 1 and 2. That is, if the direction of the path 1 is in the negative direction along the Z axis, the value of W is negative.

In single block mode, operations 1, 2, 3 and 4 are performed by pressing the cycle start button once.

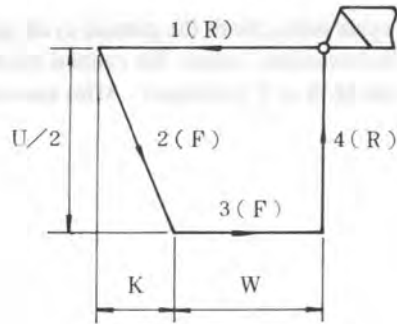
- (2) The taper face cutting cycle is specified in the following command.

G94X(U) — Z(W) — K — F — ;

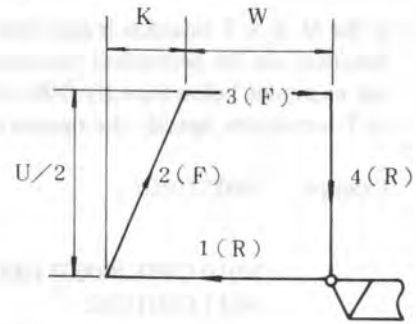


In incremental programming, the relationship between the sign of numbers following addresses U, W and K, and the tool path is as follows:

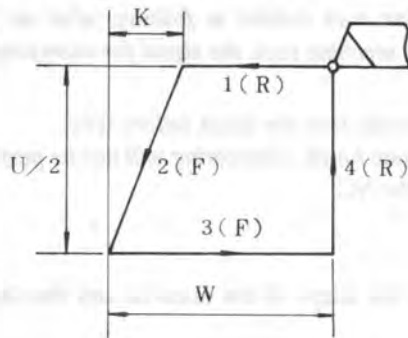
(i) $U < 0, W < 0, K < 0$



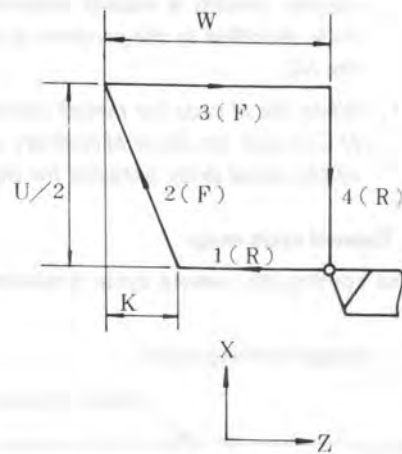
(ii) $U > 0, W < 0, K < 0$



(iii) $U < 0, W < 0, K > 0$ at $|K| \leq |W|$



(iv) $U > 0, W < 0, K > 0$ at $|K| \leq |W|$

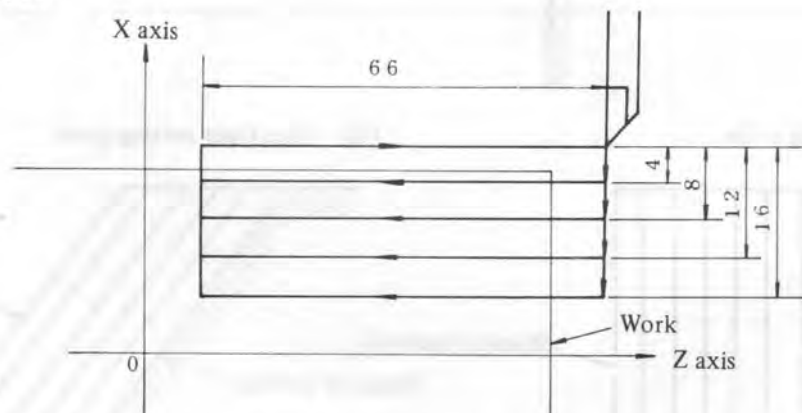


(Note 1) Since data values of $X(U)$, $Z(W)$, I and K during a canned cycle are modal, if they are not newly specified the previously specified value is effective.

Thus, when the Z -axis movement amount does not vary as in the example below, a canned cycle can be repeated by only specifying the movement commands for the X -axis.

However, this data is cleared, if a one-shot G code except for $G04$ (dwell) or a G code in 01 group except for $G90$, $G92$ and $G94$ is specified.

Example:



The cycle in the above figure is executed by the following program (Diameter programming)

```

N030 G90      U-8000 W-66000 F400;
N031          U-16000;
N032          U-24000;
N033          U-32000;

```


(Note 2) By setting parameter MOR, the following two applications can be performed.

- (1) If an EOB (;) or zero movement command is specified in the block following the block of a canned cycle, the same canned cycle is repeated.
- (2) If the M, S or T function is specified during the canned cycle mode, both the canned cycle and M, S or T function can be performed simultaneously. If this is inconvenient, cancel the canned cycle once as in the examples below (specify G00 or G01) and execute the M, S or T command. After execution of M, S or T terminates, specify the canned cycle again.

Example: N003T0101 ;

```

      :
      :
N010 G90X20000Z10000F200 ;
N011 G00T0202 ;
N012 G90X20500Z10000 ;

```

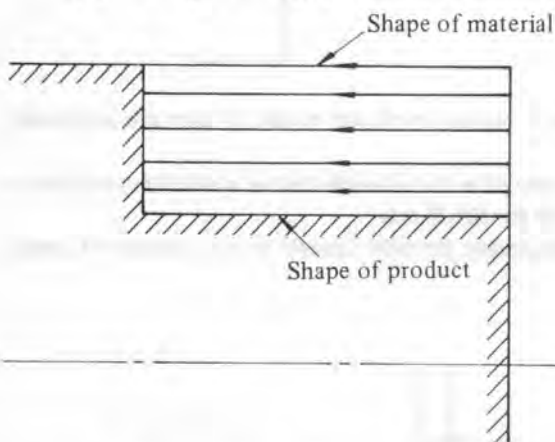
(Note 3) Thread chamfering is performed when a signal is received from the machine tool. However, the signal output process is usually designed by the machine tool builder as follows: after an appropriate M code specified in the program is transmitted to the machine tool, the signal for chamfering is output to the NC.

(Note 4) When the M code for thread chamfering is used, specify it in the block before G92. If G92 and the above M code are specified in the same block, chamfering will not be performed because of the signal delay between the machine tool and the NC.

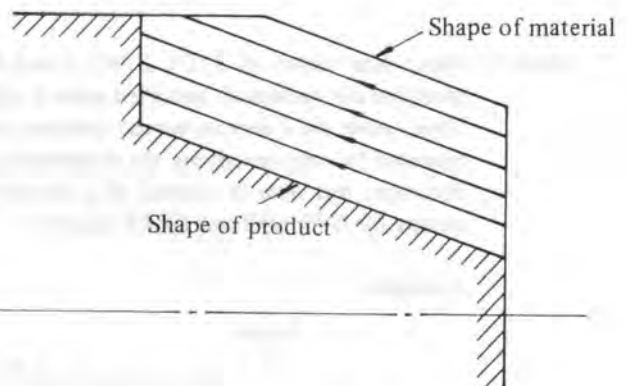
7.1.4 Canned cycle usage

An appropriate canned cycle is selected depending on the shape of the material and the shape of the product.

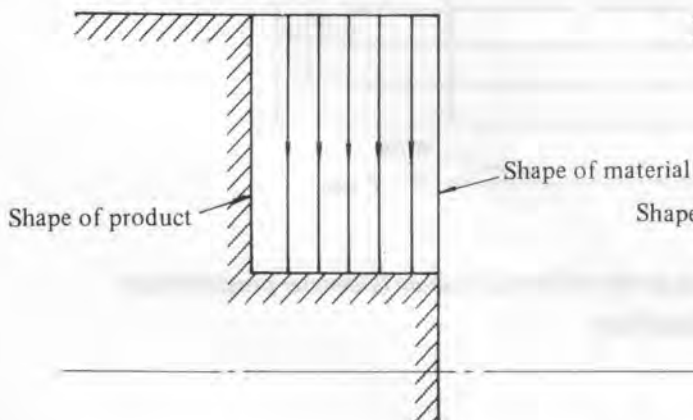
(1) Straight cutting cycle



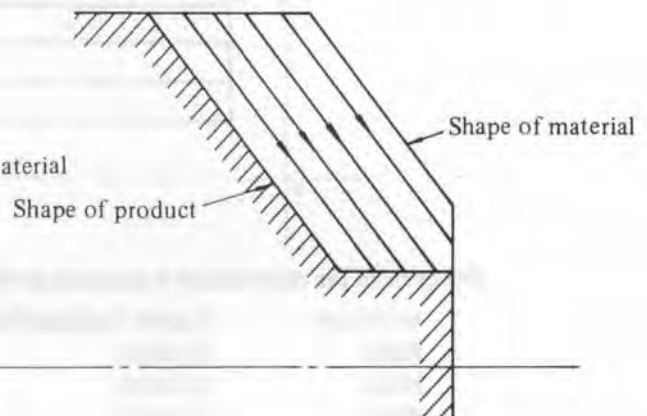
(2) Taper cutting cycle



(3) Face cutting cycle



(4) Face taper cutting cycle



7.2 Multiple Repetitive Cycles (G70 ~ G76)

This option provides canned cycles to make NC programming easy. For instance, data on the finished work-piece shape describes the tool path for rough machining. A canned cycle for thread cutting is also available.

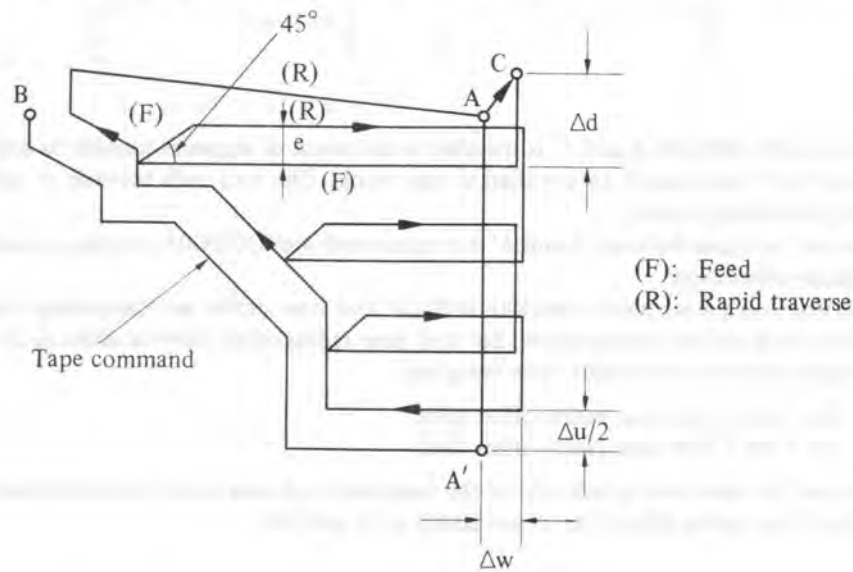
The multiple repetitive cycles (G70 ~ G73) are used only in memory operation.

7.2.1 Stock removal in turning (G71)

There are two types (Type I and II) of stock removal in turning.

(1) Type I

If a finished shape of A to A' to B is specified on a tape as in the figure below, the specified area is removed with a cutting depth of Δd , leaving a finishing allowance of $\Delta u/2$ and Δw .



The relief amount (e) is set by a parameter or SETTING 067 (MRC DT).

The first cut is made parallel to the Z axis with a cutting depth of Δd and the last cut is made parallel to the programmed path.

The tape command is specified as follows:

G71P (ns) Q (nf) U (Δu) W (Δw) D (Δd) F (f) S (s) T (t) ;

N (ns) -----

F -----

S -----

T -----

N (nf) -----

The move command from A to A' to B is specified in the blocks from sequence number ns to sequence number nf.

P: Sequence number (ns) of the first block for the program between A and B.

Q: Sequence number (nf) of the last block for the program between A and B.

U: Distance and direction of relief along the X axis (Δu) in diameter programming (In radius programming it is $\Delta u/2$)

W: Distance and direction of relief along the Z axis (Δw)

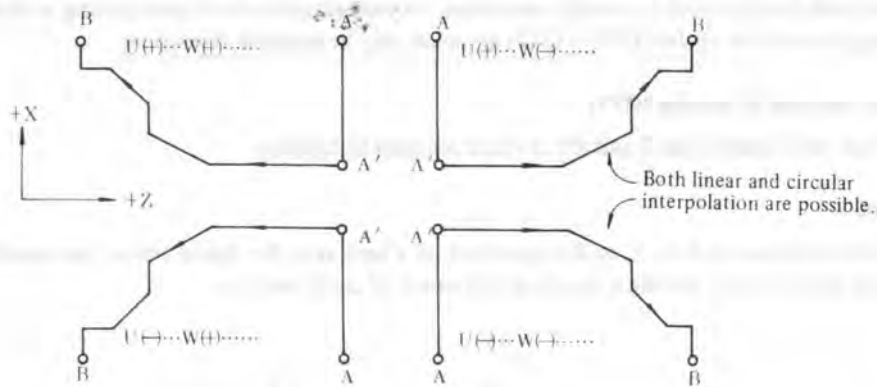
D: Depth of cut (Δd) Designation without sign.

The cutting direction depends on the direction AA'.

F, S, T: Any F, S or T functions included in blocks P to Q in the cycle are ignored, and the F, S or T function in this G71 block is effective.

F, S and T functions which are specified in the same block as G71 or specified before this block are effective. When the constant surface speed control option is selected, G96 or G97 commands are not effective between ns and nf. However, those specified in the G71 block or specified before this block are effective.

The following four cutting patterns are considered. All of these cutting cycles are performed parallel to the Z axis and the signs of Δu and Δw are as follows.



The tool path between A and A' is specified in the block of sequence number ns and a command specifying motion along the Z axis cannot be specified in this block. The tool path between A' and B must be a steadily increasing or decreasing pattern.

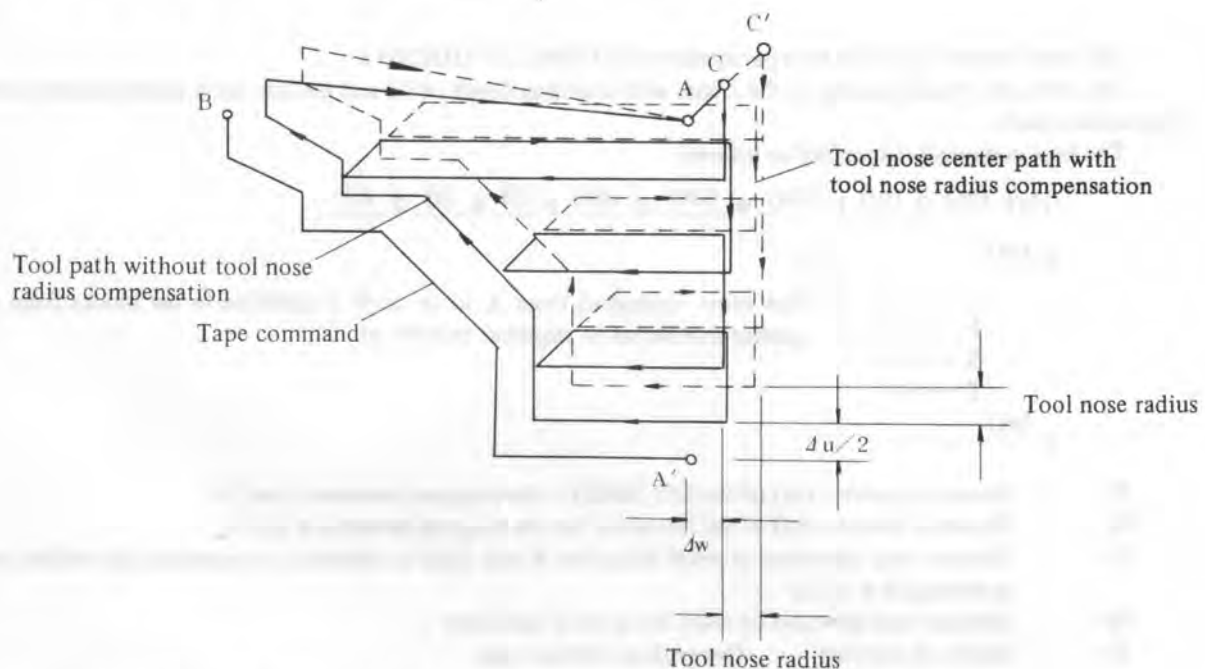
When the tool path between A and A' is programmed with G00/G01, cutting or positioning is performed in G00/G01 mode respectively.

If the machining start point coincides with the tool nose center and machining is performed by executing G71 with tool nose radius compensation, the tool nose radius offset value is added to Δu and Δw and the rough cutting is made with tool nose radius value being zero.

$$\Delta u = \Delta u + \text{Tool nose radius offset value}$$

$$\Delta w = \Delta w + \text{Tool nose radius offset value}$$

If Δu and Δw were zero previously or the imaginary tool nose center is programmed as the machining start point, the tool nose radius offset value is not added to Δu and Δw .



(Note 1) No subprogram can be called in the blocks between sequence numbers ns and nf.

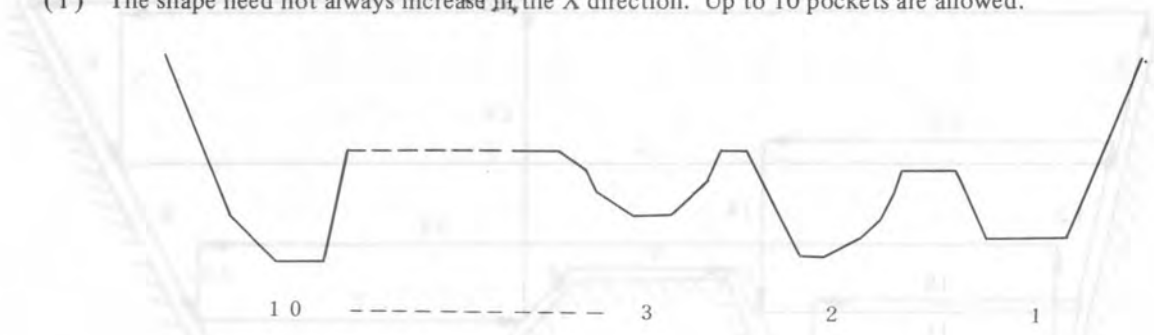
(Note 2) The last motion in this cycle is to machine along the workpiece shape leaving a finishing allowance. But this can be omitted by setting a parameter.

(Note 3) In Note 2, since cutting is terminated referring to the blocks from ns to nf for a finished shape, G01 does not always come at the end (G02 or G03 may). Therefore, set another G code in the block to be executed next.

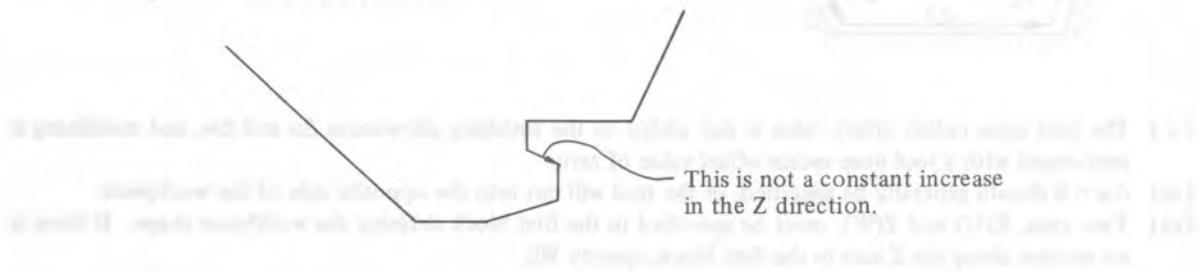
(2) Type II

Type II differs from type I in the following points.

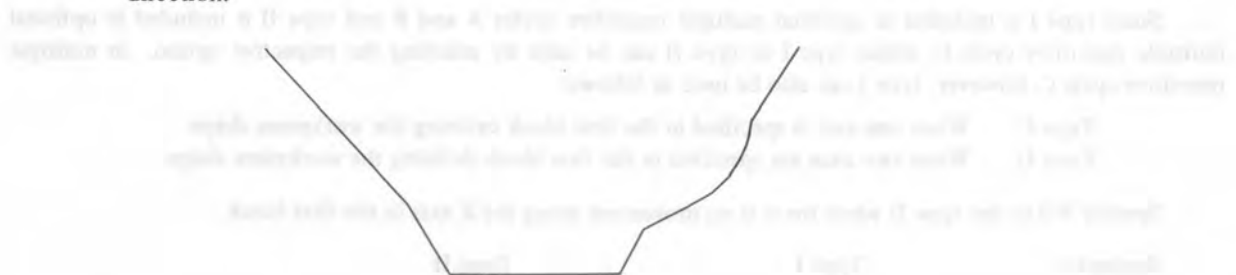
- (i) The shape need not always increase in the X direction. Up to 10 pockets are allowed.



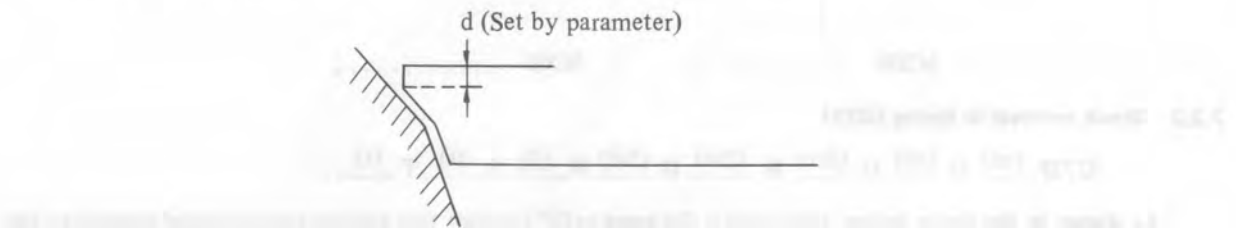
However, the shape must constantly increase in the Z direction. The following shape is not valid.



- (ii) The cut start face need not be perpendicular. Any shape is permitted if it constantly increases in the Z direction.

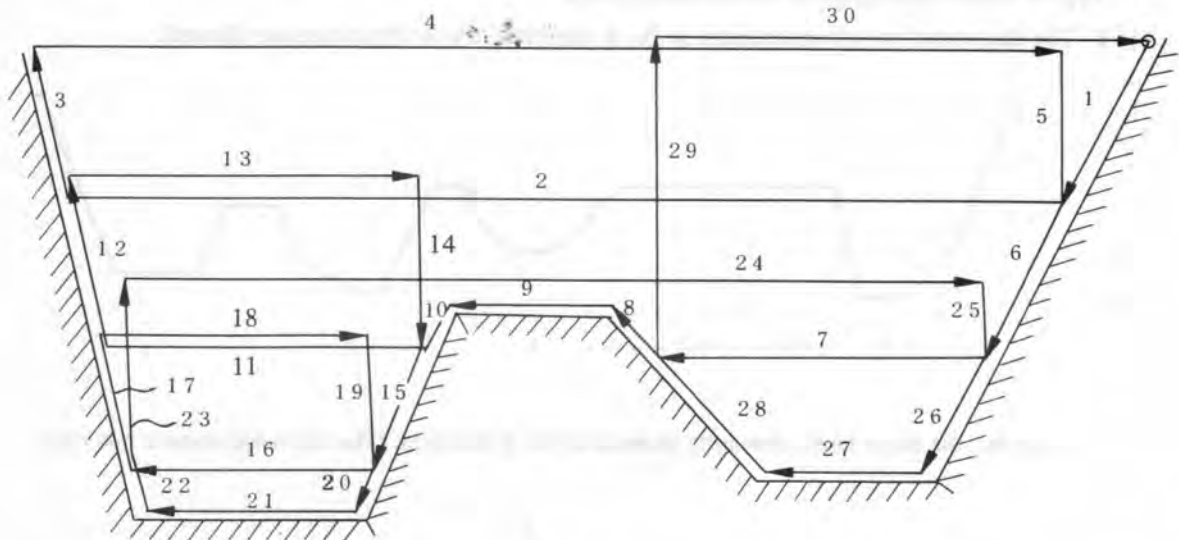


- (iii) The relief after cutting is along the shape.



The clearance d after relief should be set as a parameter in advance.

(iv) The diagram below shows an example of the cutting path when there are two pockets.



- (v) The tool nose radius offset value is not added to the finishing allowances Δu and Δw , and machining is performed with a tool nose radius offset value of zero.
- (vi) $\Delta w = 0$ should generally be specified, or the tool will cut into the opposite side of the workpiece.
- (vii) Two axes, X(U) and Z(W), must be specified in the first block defining the workpiece shape. If there is no motion along the Z axis in the first block, specify W0.

(3) Use of Type I and II

Since type I is included in optional multiple repetitive cycles A and B and type II is included in optional multiple repetitive cycle C, either type I or type II can be used by selecting the respective option. In multiple repetitive cycle C, however, type I can also be used as follows:

Type I: When one axis is specified in the first block defining the workpiece shape.

Type II: When two axes are specified in the first block defining the workpiece shape.

Specify W0 to use type II when there is no movement along the Z axis in the first block.

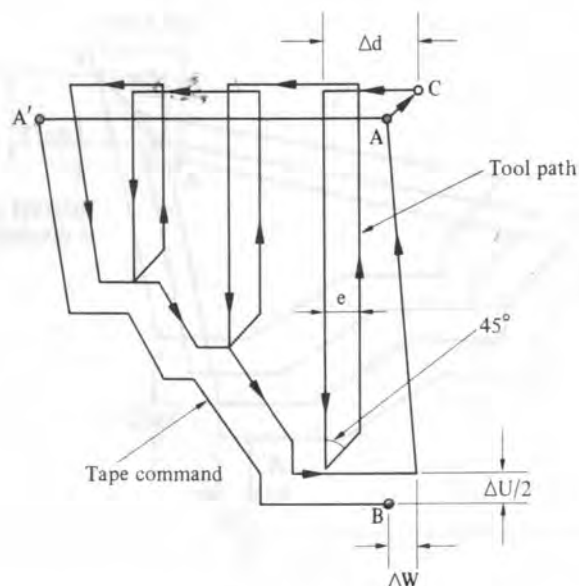
Example:

Type I	Type II
G71P100Q200	G71P100Q200
N100X(U).....;	N100X(U) Z(W)
.....	
N200	N200

7.2.2 Stock removal in facing (G72)

G72P (ns) Q (nf) U (Δu) W (Δw) D (Δd) F (f) S (s) T (t) ;

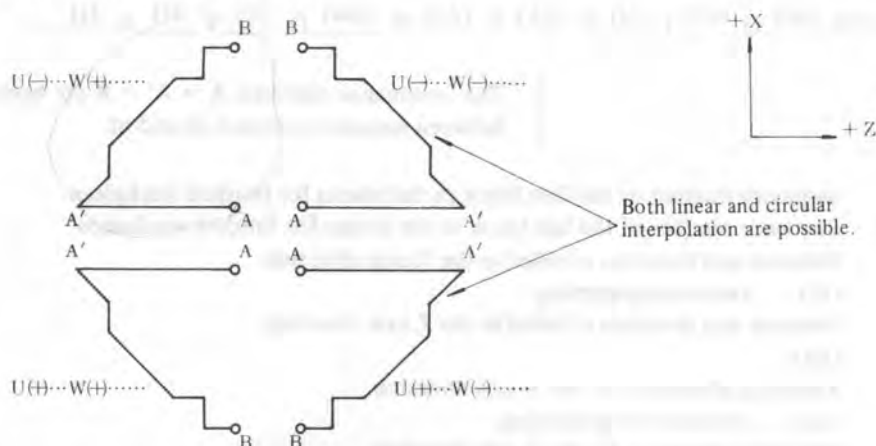
As shown in the figure below, this cycle is the same as G71 except that cutting be performed parallel to the X axis.



The relief amount (e) is set by a parameter or SETTING 067 (MRC DT)

The following four patterns can be cut by execution of G72. These patterns are cut by repetition of an operation parallel to the X axis of the tool.

The signs of ΔU and ΔW are as follows:



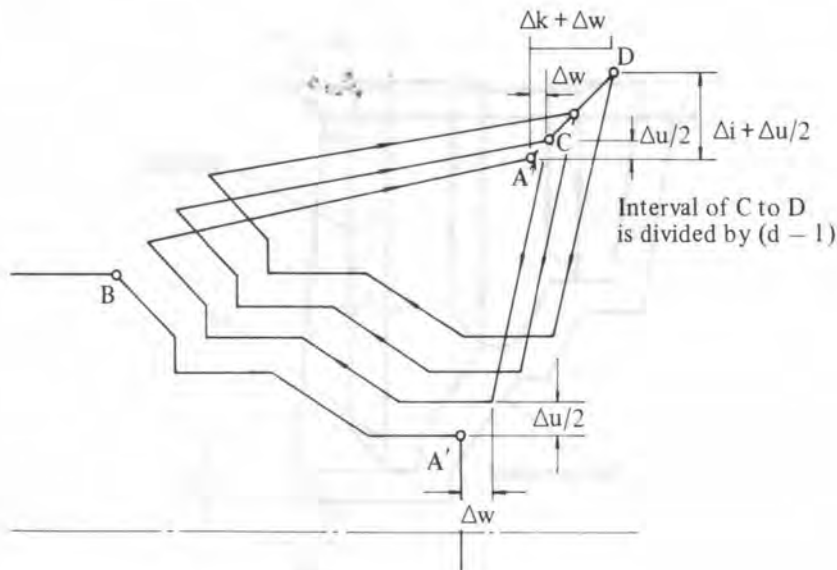
The machining mode (cutting feed or positioning) of the tool path between A and A' depends on G00/G01 in the block specifying this tool path, the same as for G71.

The tool path between A and A' is specified in the first block (sequence number ns) and a move along the Z axis cannot be specified in this block. The tool path between A' and B must be a steadily increasing or decreasing pattern.

If the machining start point is programmed at the tool nose center and machining is performed by execution of G72 with tool nose radius compensation, the tool nose radius offset value is added to Δu and Δw the same as for G71.

7.2.3 Pattern repeating (G73)

This function cuts a fixed pattern repeatedly and displaces the tool position bit by bit. It is possible using this cutting cycle to efficiently cut a workpiece whose rough shape has already been rough cut, forged or casted.



The following pattern should be specified on the tape.

$A \rightarrow A' \rightarrow B$

The tape command is specified as follows:

G73P (ns) Q (nf) I (Δi) K (Δk) U (Δu) W (Δw) D (d) F (f) S (s) ;

N (ns) ;
 ;
 N (nf) ; } The commands between $A \rightarrow A' \rightarrow B$ are specified in the blocks between sequence numbers ns and nf.

P: Sequence number of the first block in the blocks for finished workpiece.

Q: Sequence number of the last block in the blocks for finished workpiece.

I: Distance and direction of relief in the X axis direction.

(Δi) ... radius programming.

K: Distance and direction of relief in the Z axis direction.

(Δk)

U: Finishing allowance for the X axis direction.

(Δu) ... diameter programming.

W: Finishing allowance for the Z axis direction.

(Δw)

D: The division number.

(Δd) ... This value is the same as the repetitive count for rough cutting.

F, S, T: Any F, S and T functions contained in the blocks between sequence No. (ns) and (nf) are ignored; however the F, S and T functions in this G73 block are effective.

There are four possible patterns that can be cut with G73. Make sure that the proper sign is used for Δu, Δw, Δi and Δk.

When the cycle is finished, the tool returns to point A.

If the machining start point is programmed at the tool nose center and the machining is performed by executing G73 with tool nose radius compensation, the tool nose radius offset value is added to Δu and Δw the same as for G71.

7.2.4 Finishing cycle (G70)

In rough cutting with G71, G72 and G73, the following command is used for finishing.

G70P (ns) Q (nf) ;

P: Sequence number (ns) of the first block in the blocks for finished workpiece.

Q: Sequence number (nf) of the last block in the blocks for finished workpiece.

(Note 1) *F, S and T functions specified in block G71, G72 or G73 are not effective, but those specified between sequence numbers ns and nf are effective in G70 mode.*

(Note 2) *When the multiple repetitive cycles (G71, G72 and G73) are performed, they must only be specified in memory operation.*

(Note 3) *When rough cutting with G71, G72 or G73 is executed, up to 3 pairs of blocks indicated by addresses P and Q can be stored. Therefore, when executing G70, the blocks indicated by P and Q addresses can be determined immediately without searching from the start of the memory. Also, several finishing cycles can be collectively executed, after several rough cuttings (G71, G72 and G73) were executed. In this case, the cycle time for the 4th and subsequent finishing will delay execution due to the memory search.*

G71P100Q200 --- ;

N100 ---

N200

G71P300Q400 --- ;

N300 ---

N400

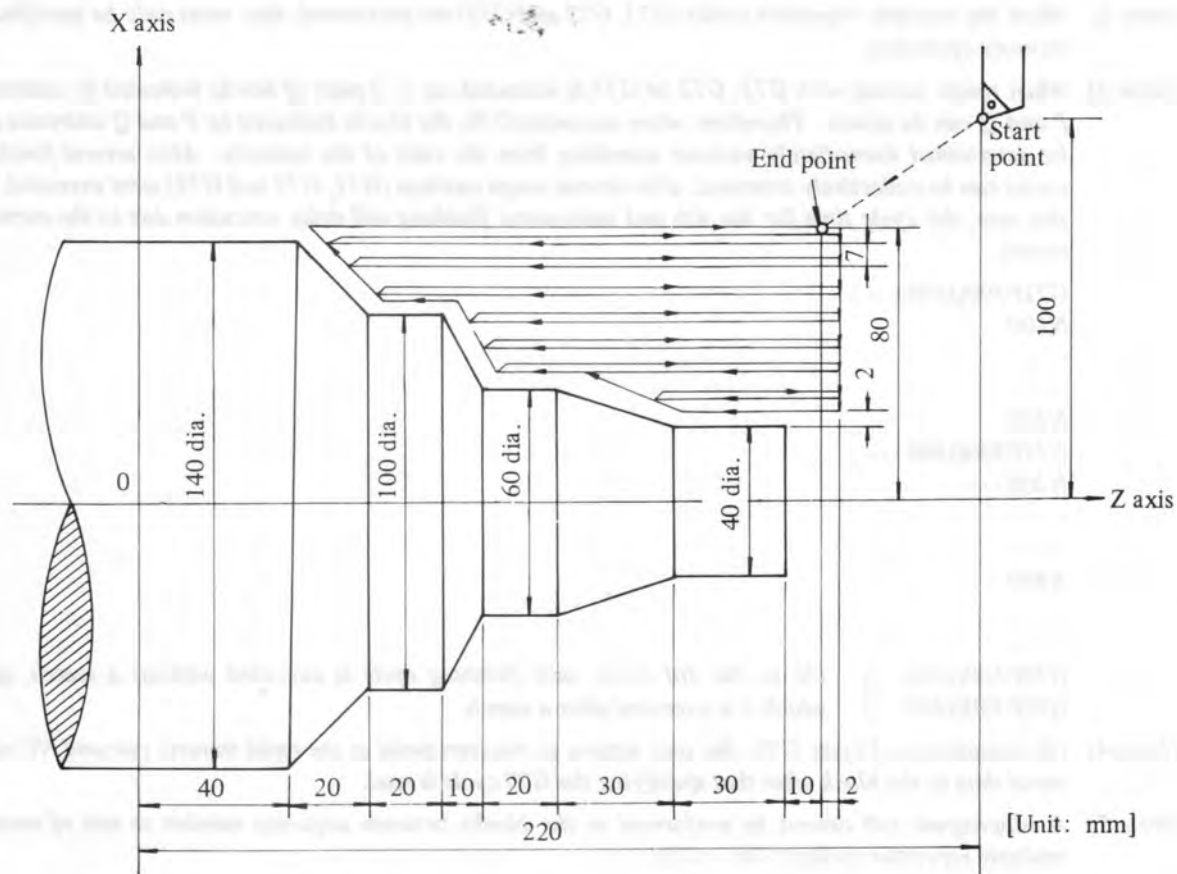
G70P100Q200; } Up to the 3rd cycle, each finishing cycle is executed without a search, after
G70P300Q400; } which it is executed after a search.

(Note 4) *On completion of cycle G70, the tool returns to the start point at the rapid traverse rate and NC command data in the block after that specifying the G70 cycle is read.*

(Note 5) *A subprogram call cannot be performed in the blocks between sequence number ns and nf used in multiple repetitive cycles (G70 – G73).*

(Note 6) *The memory addresses stored during rough machining cycles specified by G71, G72 and G73 are erased after execution of G70. All memory addresses stored are erased by a reset operation.*

(Example 7.2.1) Example of multiple repetitive cycle programming (G70, G71)

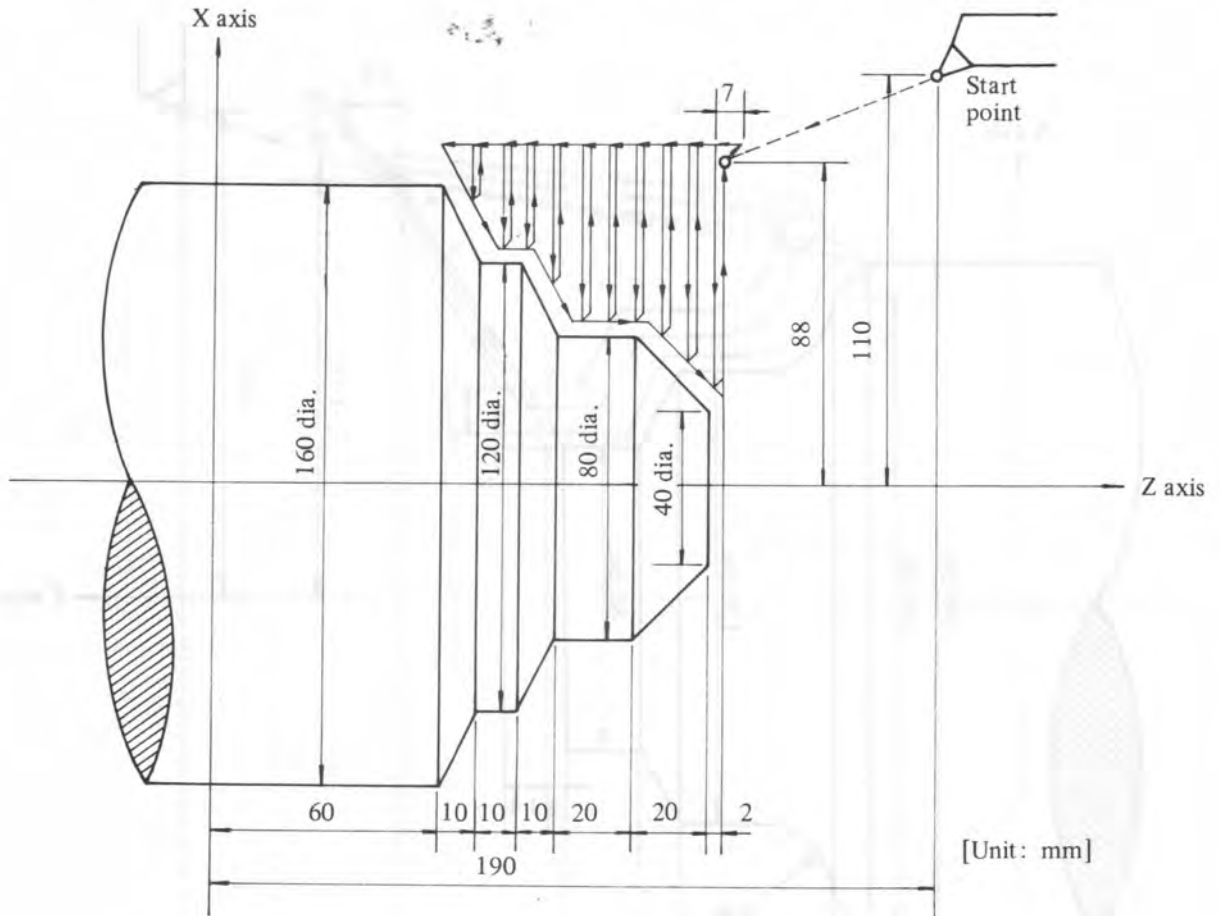


(Diameter programming and metric system input)

```

N010G50X200.0Z220.0 ;
N011G00X160.0Z180.0 ;
N012G71P013Q019U4.0 W2.0 D7000F30S55 ;
N013G00X40.0 F15S58 ;
N014G01W-40.0 ;
N015 X 60.0 W-30.0 ;
N016 W-20.0 ;
N017 X100.0 W-10.0 ;
N018 W-20.0 ;
N019 X140.0 W-20.0 ;
N020G70P013Q019 ;
    
```

(Example 7.2.2) Example of multiple repetitive cycle programming (G70, G72)

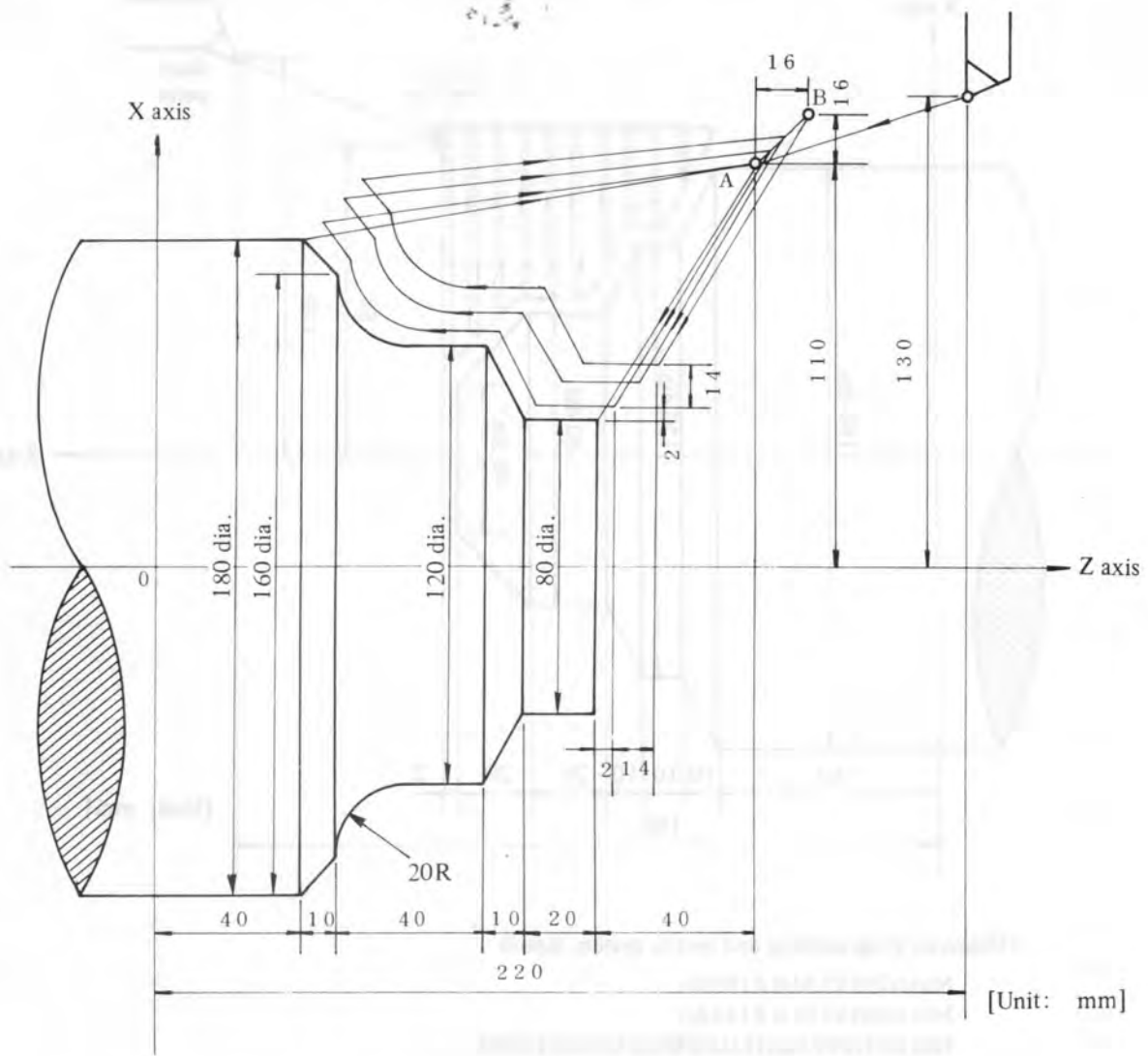


(Diameter programming and metric system input)

```

N010G50X220.0 Z190.0 ;
N011G00X176.0 Z132.0 ;
N012G72P013Q018U4.0 W2.0 D7000 F30S55 ;
N013G00Z58.0 F15S58 ;
N014G01X120.0 W12.0 ;
N015 W10.0 ;
N016 X80.0 W10.0 ;
N017 W20.0 ;
N018 X36.0 W22.0 ;
N019G70P013Q018 ;
    
```

(Example 7.2.3) Example of multiple repetitive cycle programming (G70, G73)



(Diameter programming and metric system input)

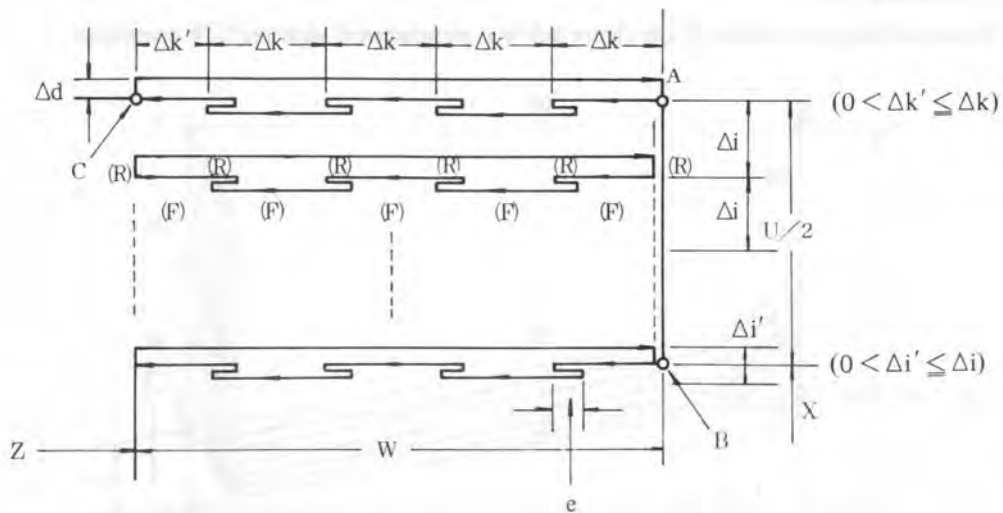
```

N010 G50X260.0 Z220.0 ;
N011 G00X220.0 Z160.0 ;
N012 G73P013Q018I14.0 K14.0 U4.0 W2.0 D3F30S0180 ;
N013 G00X80.0 W-40.0 ;
N014 G01W-20.0 F15S0600 ;
N015 X120.0 W-10.0 ;
N016 W-20.0 S0400 ;
N017 G02X160.0 W-20.0 I20.0 ;
N018 G01X180.0 W-10.0 S0280 ;
N019 G70P013Q018 ;
    
```

7.2.5 Peck drilling along the Z axis (G74)

The following command is used to execute the operation shown in the figure below. Chip breaking is possible in this cycle as shown. If X(U) and I are omitted, drilling can only be performed along the Z axis.

G74X(U)_____Z(W)_____I (Δi) K (Δk) F_____D (Δd) ;



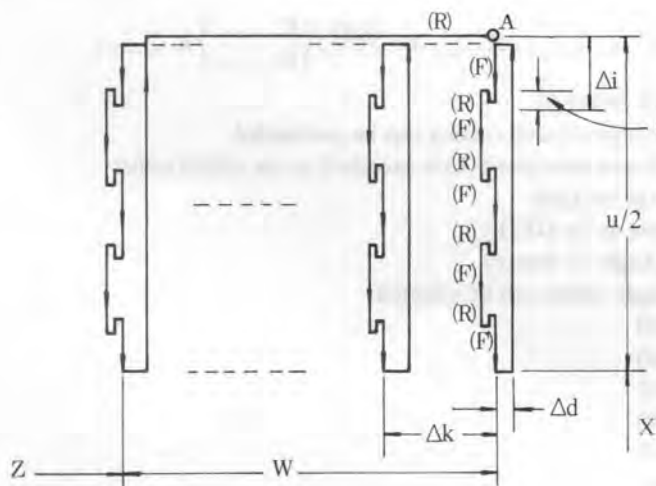
- X: X component of point B
U: Incremental amount from A to B
Z: Z component of point C
W: Incremental amount from A to C
I: Movement amount in the X direction (non-signed designation) (Δi)
K: Depth of cut in the Z direction (non-signed designation) (Δk)
D: Relief amount of the tool at the bottom of the cut (Δd)
(When there is no D, D is assumed to be zero.)
The sign of D is always +. However, if address X(U) and I are omitted, the relief direction can be determined by the sign.
F: Feedrate

(Note) Feed back amount e is set by parameter or setting operation 065 (GROVE).

7.2.6 Grooving in X axis (G75)

The following tape command is used to execute the operation shown in the figure below. This is equivalent to G74 except that X is replaced by Z. Chip breaking is possible in this cycle, and grooving and Peck drilling along the X axis (In this case, the Z, W and K axes are omitted) are possible.

G75X(U) _____ Z(W) _____ I (Δi) K (Δk) F _____ D (Δd);

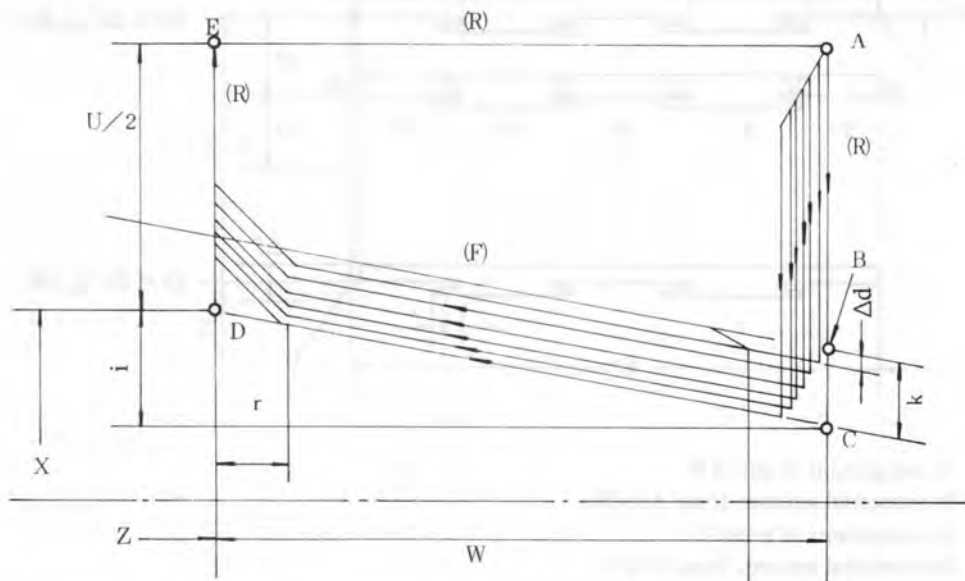


G74 and G75 are used for grooving and drilling, and allow the tool to perform relief automatically. Four symmetrical patterns are considered, respectively.

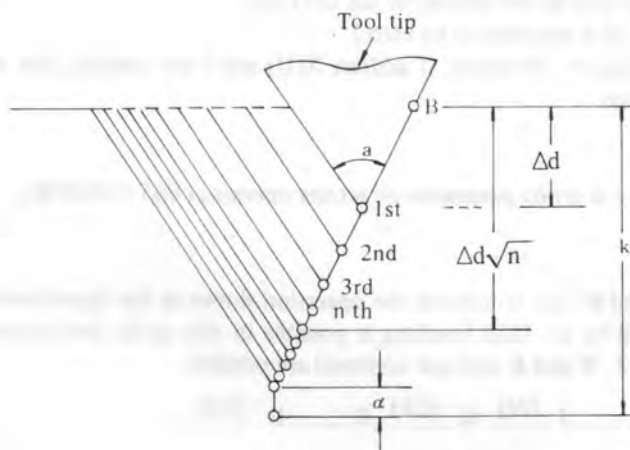
(Note) Feed back amount e is set by a parameter or setting operation 065 (GROVE).

7.2.7 Thread cutting (G76)

The thread cutting cycle shown in the figure below is programmed with the G76 command.



Detail of cutting



G76X(U) — Z(W) — I — K — D (Δd) { F — } A — ;

- I: Difference in thread radius (i)
If $I = 0$, ordinary straight thread cutting can be performed.
- K: Height of thread (X-axis direction) (k) is specified as the radius value.
- D: Cutting depth for 1st cut (Δd)
- F, E: Lead of thread (same as for G32)
- A: Angle of tool tip (Angle of thread)
..... One of six angle values can be selected.
In case of 80° : A80
In case of 60° : A60
In case of 55° : A55
In case of 30° : A30
In case of 29° : A29
In case of 0° : A0

If A is omitted, it is assumed to be zero.

r: Chamfering

When the thread lead is expressed as $\frac{L}{n}$, the value of r can be set from 0.1L to 3.1L in 0.1L increments by parameter or setting operation 064 (THDCH).

By using this cycle, one edge is cut and the load on the tool tip is reduced. If the cutting depth is Δd for the first pass and $\Delta d \sqrt{n}$ for the nth pass, the cutting amount per cycle is held constant. A minimum cutting depth can be set by setting operation 068 (THCLM) to prevent the cutting depth from becoming too small. Four patterns can be cut with this cycle by changing the sign of each address. Therefore, female screws can be machined.

In the above figure, the cutting feedrate between C and D is specified in the F or E code. Along other tool paths, the tool is fed at the rapid traverse rate.

The signs of incremental dimensions for the above figure are as follows:

U, W: Negative (determined by the direction from A to C and C to D)

I: Negative (determined by the direction from A to C)

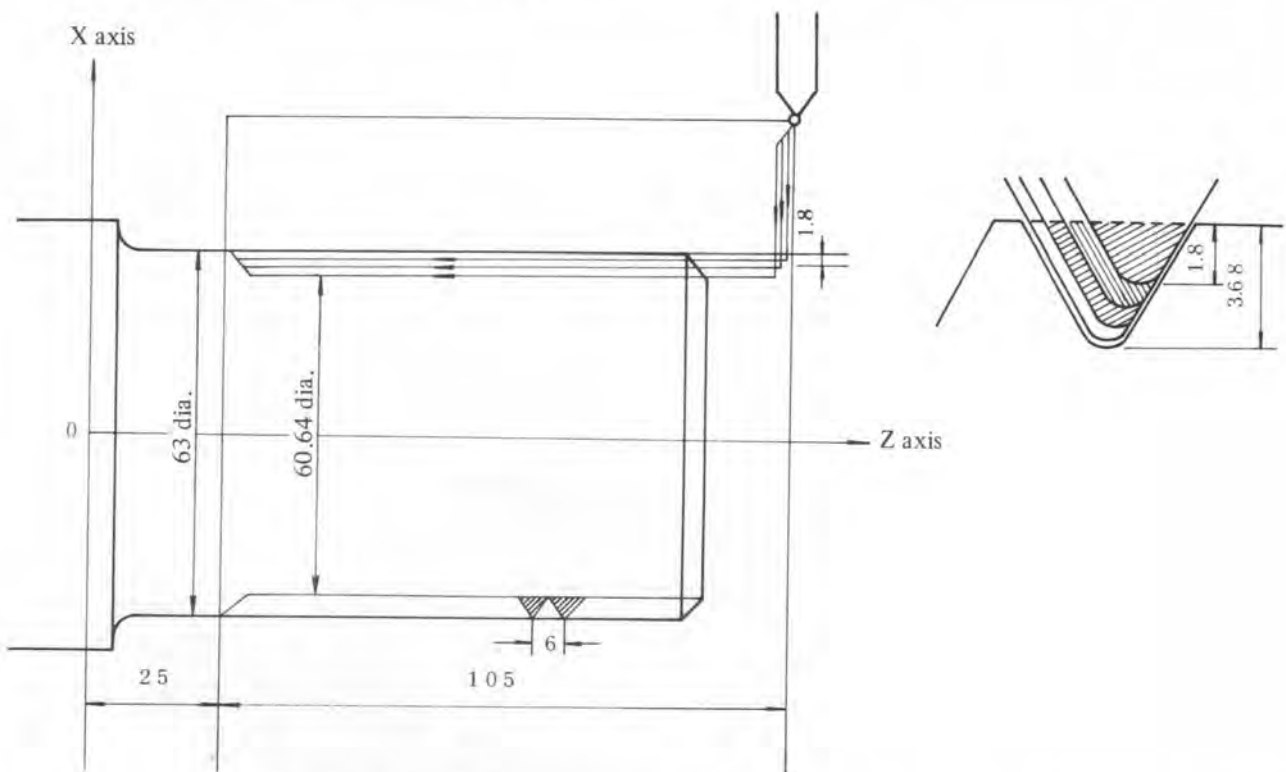
K: Positive (always)

D: Positive (always)

Finishing allowance α (radius value) is set by parameter or setting operation 066 (THDFN) in the following range:

0.000 ~ 32.767 mm or 0.000 ~ 3.2767 inches

(Example 7.2.7.1) Example of multiple repetitive cycle programming (G76)



G76X60640 Z25000 K3680 D1800 F600 A60;

(Note 1) Notes on thread cutting are the same as those on G32 thread cutting and on the G92 thread cutting cycle. When Feed Hold status is entered during thread cutting when the "Temporary Stop during Thread Cutting" option is in effect, the tool returns to the start point of the cycle.

7.2.8 Notes on multiple repetitive cycles (G70 ~ G76)

- (1) In the blocks specifying the multiple repetitive cycles, addresses P, Q, X, Z, U, W, I, K, D and A or the like should be specified correctly.
- (2) When G71, G72 or G73 is specified, 01 group G code (G00 or G01) must be specified in the block in

which the sequence number is specified with P. If it is not specified, alarm No. 65 occurs.

- (3) G70, G71, G72 and G73 cannot be specified in the TAPE mode and MDI mode. If specified, P/S alarm No. 67 occurs. G74, G75 and G76 can be specified in the TAPE, MDI and MEMORY modes.
- (4) Neither M98 nor M99 may be specified in the blocks in which G70, G71, G72 and G73 are specified and in blocks between those specified by P and Q.
- (5) The following commands cannot be specified between the blocks specified by P and Q for G70, G71, G72 and G73.
 - One shot G code except for G04.
 - 01 group G code except for G00, G01, G02 and G03.
 - 05 and 06 group G code
 - Tool function
 - M98/M99
- (6) It is possible to stop the cycle and enter a manual operation while a multiple repetitive cycle is being executed. But the cycle should be restarted from the position where the manual operation began. If restarting without returning the tool to its original position when manual operation was started, movement made during the manual operation is not added to the tool position even if the manual absolute switch is ON. The tool path is shifted by the amount of the manual movement.
- (7) When G70, G71, G72 or G73 is executed, the sequence numbers specified with P and Q must not be the same as those already been stored in the program.
- (8) The last move command in the blocks between the two blocks specified with P and Q in G70, G71, G72 or G73 must not include chamfering or corner R specifications. If specified, alarm No. 69 will occur.

8. SPINDLE SPEED FUNCTION (S FUNCTION), TOOL FUNCTION (T FUNCTION) MISCELLANEOUS FUNCTION (M FUNCTION)

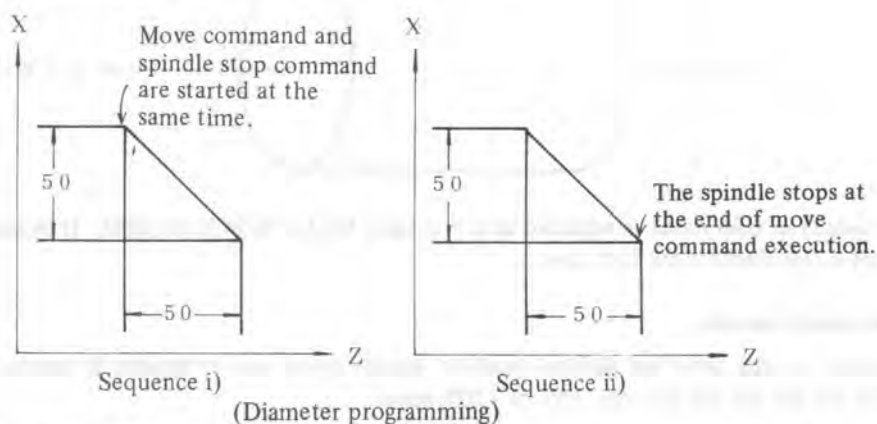
By specifying a numerical value following address S, T, M, a BCD signal and a strobe signal are transmitted to the NC machine tool. This is mainly used to control the ON/OFF function of the machine.

S codes are used for spindle control, T codes are used for tool changes, and M codes are used to control ON/OFF of the various functions of the machine tool, etc. Refer to the machine tool builder's manual for application of these addresses and codes to functions.

When a move command and S, T, M codes are specified in the same block, the commands are executed in one of the following two ways:

- (i) Simultaneous execution of the move command and S, T and M function commands.
- (ii) Executing S, T, and M function commands upon completion of move command execution.

(Example) N1 G01U-100.0 W50.0 M05; (Spindle stop)



The selection of either sequence depends on the machine tool builder's specifications. In certain cases, these sequences may be provided together in an NC machine. Refer to the manual issued by the machine tool builder for details.

8.1 Spindle Speed Function (S Function)

8.1.1 S 2-digit

The spindle speed is controlled by address S and the following 2-digit number. Refer to the manual issued by the machine tool builder for details.

(Note 1) When a 4-digit S code is specified in the 2-digit specification, the lower 2 digits are used.

8.1.2 S 4-digit

The spindle speed (rpm) is directly specified by address S and the following 4-digit number. The unit of spindle speed varies with the machine tool builder.

8.2 Constant Surface Speed Control

If surface speed (relative speed between tool and workpiece) is set after address S, the spindle speed is calculated so that the surface speed is always the specified value considering the instantaneous tool position. Voltage is fed to the spindle control section so that the spindle rotates to produce the correct surface speed.

The units of surface speed are as follows:

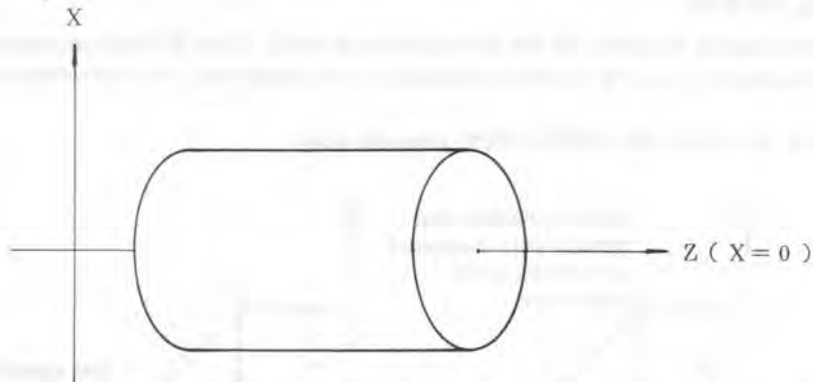
Input unit	Surface speed unit
mm	m/min.
inches	feet/min.

The surface speed unit varies with machine tool builder. The following G codes are used for surface speed control.

8.2.1 Command

G code	Meaning	Unit
G96	Constant surface speed control	m/min. feet/min.
G97	Specifies the spindle speed	rpm

When surface speed control is in effect, the work coordinate system must be set so that the axis of rotation may be set to the Z axis ($X = 0$).



S (surface speed) in G96 mode is regarded as $S = 0$ until M03 or M04 is specified. It becomes in effect when either M03 or M04 is specified for the first time.

8.2.2 Spindle speed override

An override for the specified surface speed or spindle speed can be applied by issuing a signal from the machine tool in 50, 60, 70, 80, 90, 100, 110, or 120% steps.

(Note) Parameter SOV must have been set to 1 to perform spindle speed override. When SOV is set to 0, spindle speed override is not effective.

8.2.3 Clamping of maximum spindle speed

The figure following G50S specify a maximum spindle speed for constant surface speed control in rpms.

G50S — ;

When the spindle speed in constant surface speed control reaches the value specified in the above program, the speed is clamped at this maximum value.

When optional constant surface speed control is equipped, maximum spindle speed can be set to parameter SPD MAX (No. 134). In this case, spindle speed is limited to the set maximum spindle speed in either constant surface speed control mode (G96) or not (G97).

8.2.4 Rapid traverse (G00)

In a rapid traverse block including a G00 command, the surface speed is not calculated according to the change in tool position, but on the basis of the end position of the block because there is no cutting during the rapid traverse.

(Note 1) When parameter SSC R is set to 0, the surface speed is calculated for the momentarily changing tool position in G00 mode also.

(Note 2) When the power supply is turned on, the maximum spindle speed is not set, that is, the speed is not clamped.

(Note 3) If the maximum spindle speed is set by a G50 command, clamping is only effective for G96 but it is not effective for G97. If the maximum spindle speed is set by parameter SPD MAX, clamping is effective for G96 and G97.

(Note 4) G50S0; indicates that the spindle speed is clamped at 0 rpm.

(Note 5) The value for S specified in G96 mode is also used in G97 mode and is restored when returning to G96 mode.

G96S50; (50 m/min. or 50 feet/min.)
 G97S1000; (1000 rpm)
 G96X1000; (50 m/min. or 50 feet/min.)

- (Note 6) The surface speed specified in constant surface speed control applies to the programmed path. It does not apply to the tool path after offset.
- (Note 7) Even when a machine is not operating in machine lock status, the constant surface speed is calculated according to the change in the coordinate value of the X axis in the program.
- (Note 8) Constant surface speed control is effective in thread cutting mode. Therefore, it is better to cancel constant surface speed control with G97 for face cutting or taper thread cutting so that the NC unit and servo system can adapt easily to the change in spindle speed.
- (Note 9) Although meaningless, G96 and G98 (feed per minute) can be in effect at the same time.
- (Note 10) When switching from G96 to G97 mode, the last spindle speed specified in the G96 command is used as the S value in the G97 mode if S (rpm) is not specified in the G97 block.

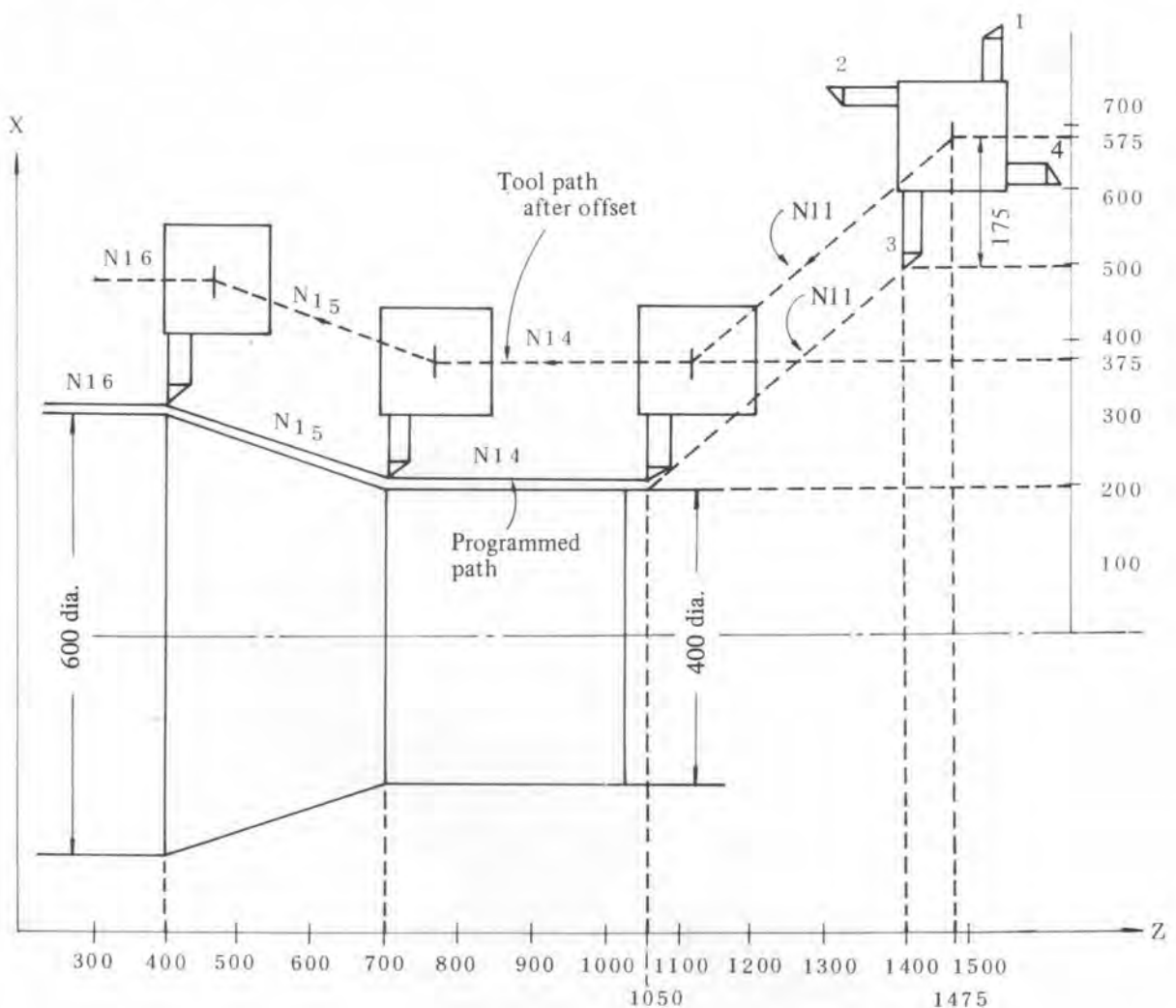
```

N111G97S800;      800 rpm
:
N222G96S100;      100 m/min.
:
N333G97;          X rpm

```

X is the spindle speed (X rpm) in the block before N333. When the G96 mode changes to the G97 mode, the spindle speed does not change. When G96 → G97, the value of S in the G96 mode is effective. When S is not specified, S = 0 m/min. (feet/min.).

8.2.5 Example



(Diameter designation)

N 8 G00 X1000.Z1400.;

N 9 T0303;

N11 X400. Z1050.;

N12 G50 S3000; (Designation of max. spindle speed)

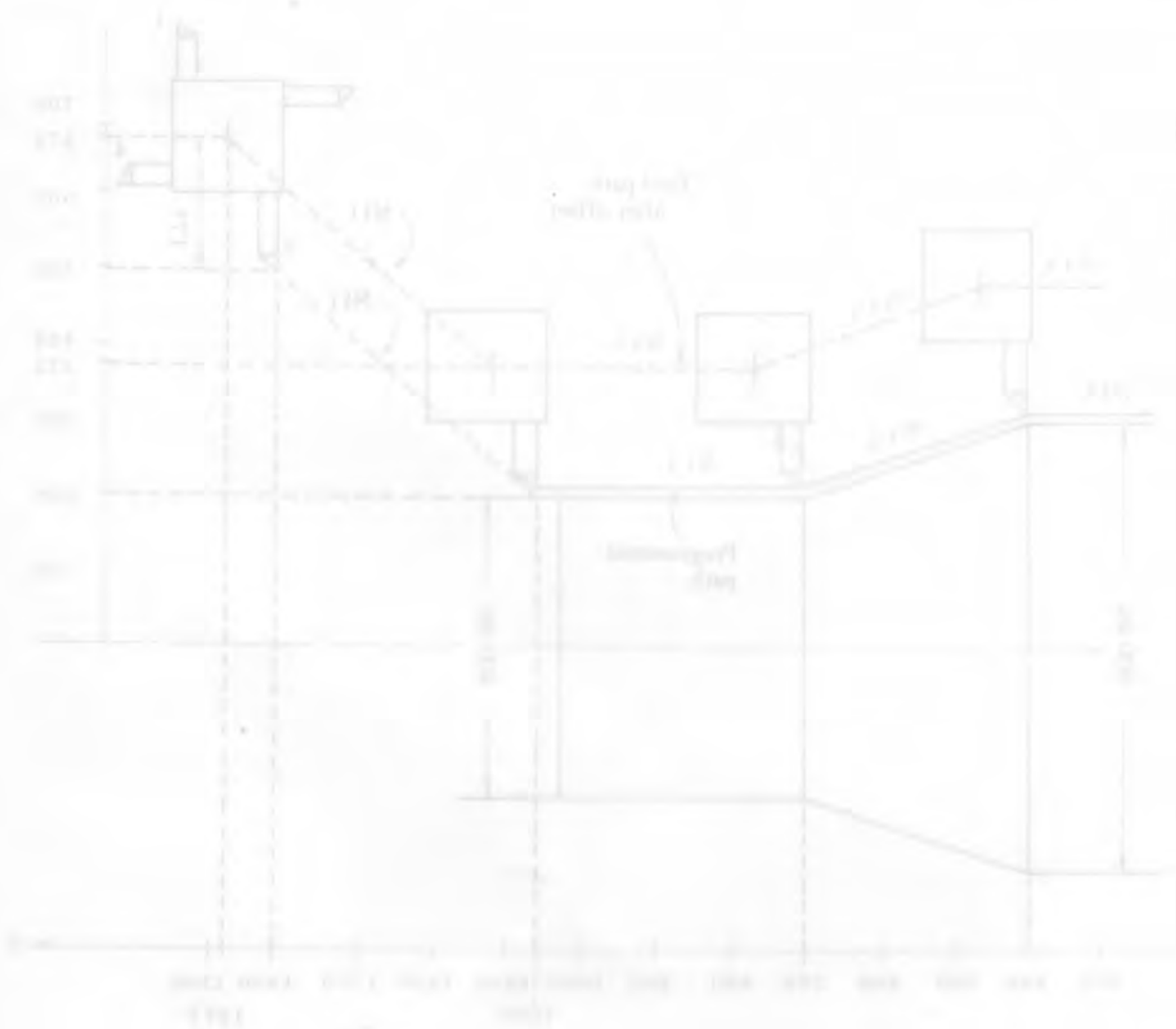
N13 G96 S 200; (Surface speed 200 m/min.)

N14 G01 Z 700.F1000;

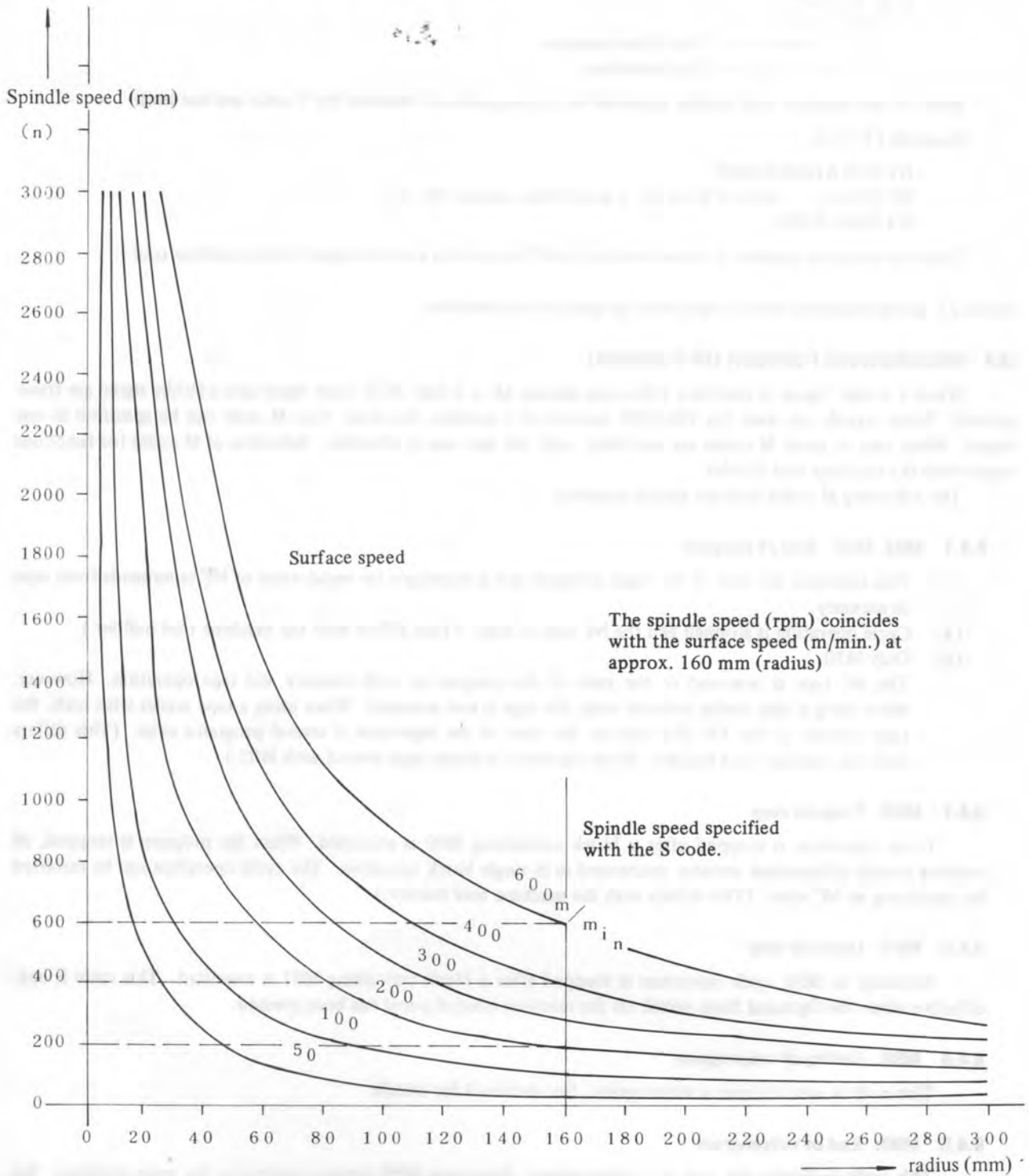
N15 X600. Z 400.;

N16 Z...;

The NC calculates the spindle speed which is proportional to the specified surface speed at the position of the programmed coordinate value on the X axis. This is not the value calculated according to the X axis coordinate after offset when offset is valid. At the end point N15 in the example above, the speed at 600 dia. (which is not the turret center but the tool nose) is 200 m/min. If the X axis coordinate value is negative, the NC uses the absolute value.



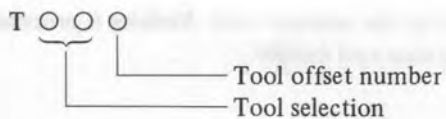
8.2.6 Relation between spindle speed and surface speed



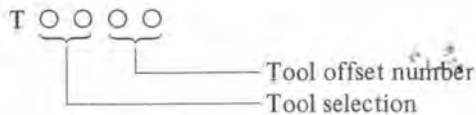
8.3 Tool Function (T Function)

The value after the T code indicates the desired tool. Part of the value is also used as the offset number indicating the compensation amount for tool offset. The following two kinds of specifications can be selected by setting parameter T2D.

(1) T2 + 1 setting



(2) T2 + 2 setting



Refer to the machine tool builder's manual for correspondence between the T-code and the tool.

Example (T2 + 2)

N1 G00 X1000 Z1400;

N2 T0313; . . . (Select Tool No. 3 and Offset amount No. 13)

N3 X400 Z1050;

This tool selection number is transmitted by a BCD code with a strobe signal to the machine tool.

(Note 1) Some machines use a 1-digit value to specify tool selection.

8.4 Miscellaneous Function (M Function)

When a 2-digit figure is specified following address M, a 2-digit BCD code signal and a strobe signal are transmitted. These signals are used for ON/OFF control of a machine function. One M code can be specified in one block. When two or more M codes are specified, only the last one is effective. Selection of M codes for functions varies with the machine tool builder.

The following M codes indicate special meaning.

8.4.1 M02, M30: End of program

- (i) This indicates the end of the main program and is necessary for registration of NC commands from tape to memory.
- (ii) Cycle operation is stopped and the NC unit is reset. (This differs with the machine tool builder.)
- (iii) Only M30
The NC tape is rewound to the start of the program in both memory and tape operation. However, when using a tape reader without reels, the tape is not rewound. When using a tape reader with reels, the tape returns to the ER (%) code at the start of the tape even if several programs exist. (This differs with the machine tool builder. Some machines indicate tape rewind with M02.)

8.4.2 M00: Program stop

Cycle operation is stopped after a block containing M00 is executed. When the program is stopped, all existing modal information remains unchanged as in single block operation. The cycle operation can be restarted by specifying an NC start. (This differs with the machine tool builder.)

8.4.3 M01: Optional stop

Similarly to M00, cycle operation is stopped after a block containing M01 is executed. This code is only effective when the Optional Stop switch on the machine control panel has been pressed.

8.4.4 M98: Calling of subprogram

This code is used to enter a subprogram. See section 9 for details.

8.4.5 M99: End of subprogram

This code indicates the end of a subprogram. Executing M99 returns control to the main program. See section 9 for details.

(Note 1) If there is a block following M00, M01, M02, or M30, it is not read into the buffer storage. Similarly, two M codes which cause the block following them to be ignored can be used by setting a parameter. Refer to the manual issued by the machine tool builder for these M codes.

(Note 2) When executing M98 or M99, a code signal and a strobe signal are not transmitted.

(Note 3) M codes except for M98 and M99 are processed by the machine tool. Nothing is processed only by the NC unit. Refer to the manual issued by the machine tool builder.

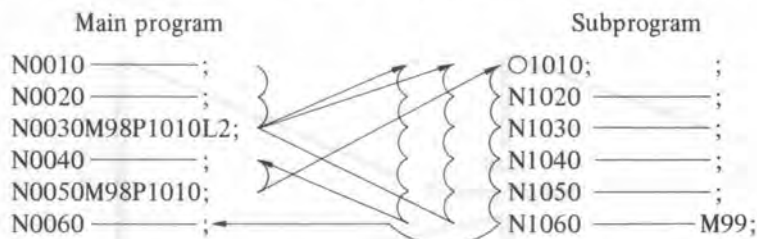
This command is read "Call the subprogram (subprogram number 1002) five times."
 The subprogram call command (M98P___ L___) and a move command can be specified in the same block.

(Example) X1000 M98P 1200;

In this example, the subprogram (subprogram number 1200) is called after completing movement in the X-axis direction.

(Example)

The execution sequence of a main program which calls a subprogram is as follows:



When the subprogram is called by another subprogram, it is executed in the same sequence as shown in the above example.

(Note 1) M98 and M99 signals are not issued to the machine tool.

(Note 2) If the subprogram number specified by address P can not be found, alarm No. 78 is generated.

(Note 3) The subprogram call command 'M98P___;' cannot be input from the MID. In this case, prepare the main program in EDIT mode as follows:

O xxxx ;
 M98P xxxx ;
 M02 ;

Then execute it in the memory mode.

(Note 4) A single block stop is not effective in blocks containing M98P___; M99; but is effective when the block contains addresses other than O, N, L and P.

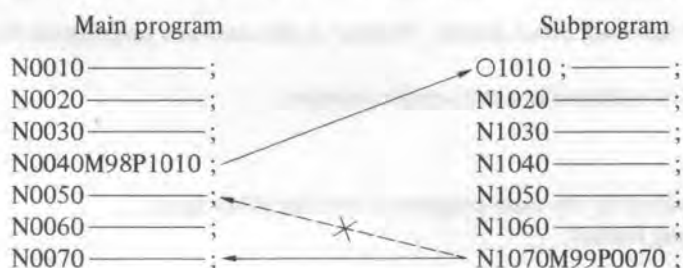
9.3 Notes on Subprogram Control

The following special specifications can be used to control subprogram execution.

9.3.1

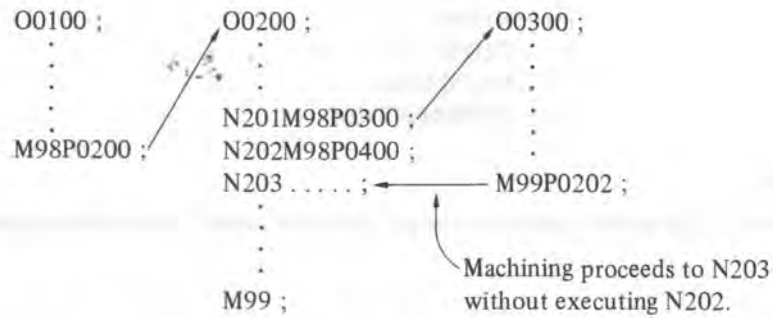
When a sequence number is specified in address P of the last block of a subprogram, control does not return to the block after the block in which the subprogram was called, but rather to the block with the sequence number specified in address P. However, this is only effective in memory operation.

The processing time for return to the specified block is considerably longer than that for normal return.



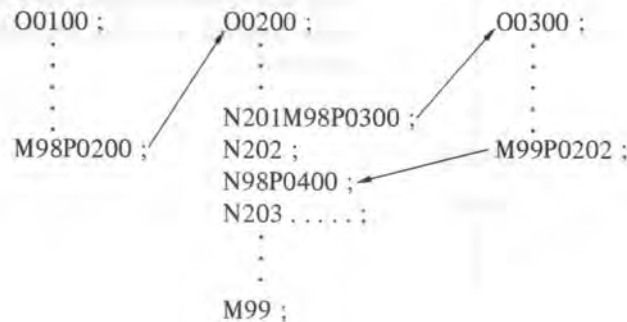
(Note) If the return destination is an independent block of M98/M99 (an address other than O, N, L, and P is not commanded) in returning from a subprogram by M99Pxxxx;, this M98/M99 block is neglected, and machining proceeds to the next block.

Example



Accordingly, designate the block of a sequence number only as the return destination by M99Pxxxx;

Example



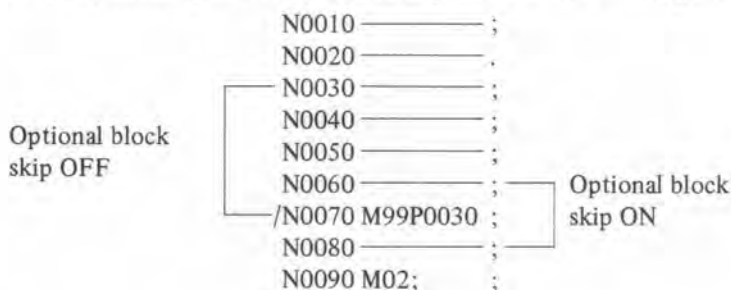
9.3.2

If the M99 command is executed in the main program, control returns to the start of the main program.

For example, a '/M99;' block is inserted in the proper position of the main program and the optional block skip is off, M99 is executed, and control returns to the start of the main program and the program is executed again.

If the optional block skip is turned on, '/M99;' is omitted and control is passed to the next block.

If '/M99Pn;' has been inserted, control does not return to the start, but returns to the block whose sequence number is 'n'. The processing time for return to sequence number n is longer than return to the start.



9.3.3

It is possible to execute a subprogram from the start by specifying a search for it from the MDI keyboard as well as the main program. (See section IV 5.14 for search)

In this case, if the M99 command is executed, control returns to the top of the subprogram and execution is repeated. If 'M99Pn;' is executed, control returns to the block whose sequence number is 'n' and execution is repeated.

In the above operation, if you want to stop execution, insert '/M02;' or the '/M30;' at the appropriate position. When the optional block skip switch is turned off, the above command is executed and the program is terminated. In the example below, execution will continue while the optional block skip is on and will stop when the optional block skip is turned off.

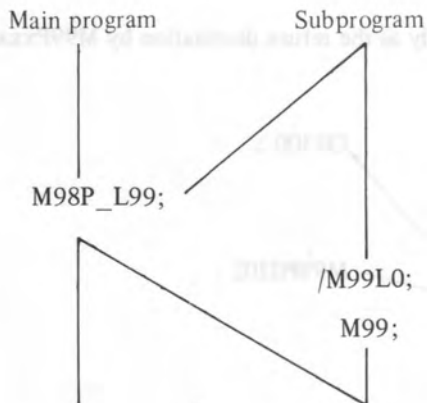

```

N1040 _____;
N1050 _____;
N1060 _____;
/N1070 M02;
N1080M99P1050;

```

9.3.4 M99L α ;

Execution of the above command changes repetitive count L in the subprogram calling block to α repetitions.



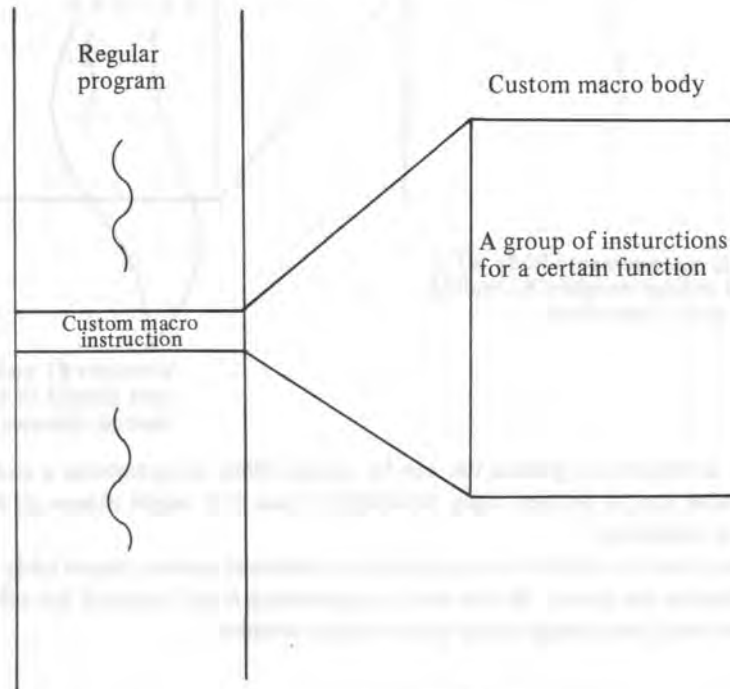
If the optional block skip is turned off in this program, the repetitive count becomes zero and control is passed to the main program when the subprogram end command (M99) is executed.

10. CUSTOM MACRO A AND B

10.1 General

Custom macros A and B are basically the same with the functional differences described in Section 10.10 (9).

A function covering a group of instructions is stored in memory the same as a subprogram. The stored function is represented by one instruction, so that only the representative instruction need be specified to execute the function. This group of registered instructions is called a "custom macro body" and the representative instruction is called a "custom macro instruction". The custom macro body may simply be called a macro. And the custom macro instruction may be called a macro call command.



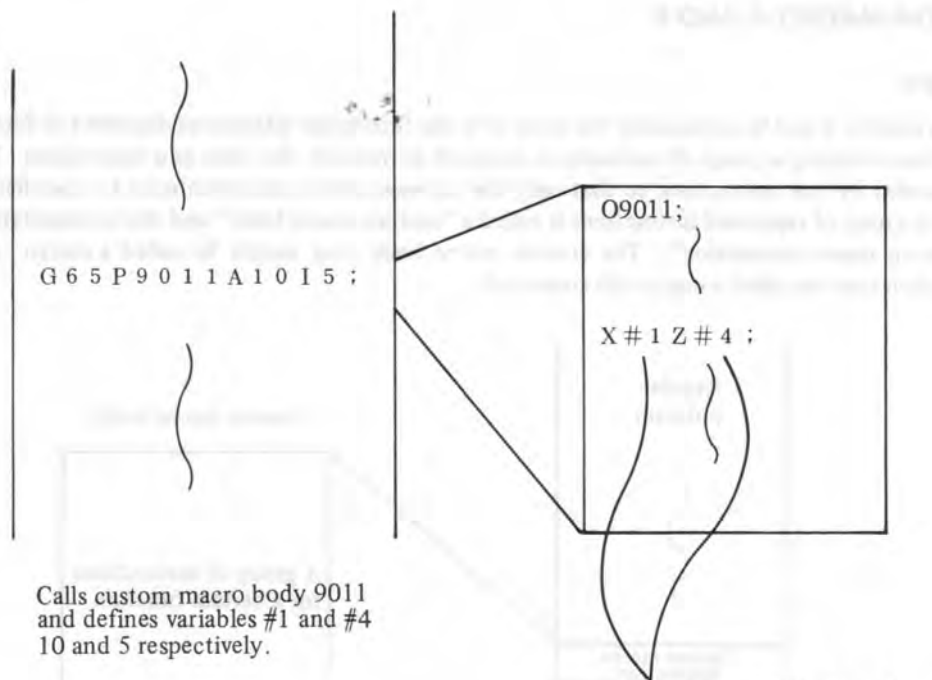
Programmers need only remember representative macro instructions without having to remember all the instructions in a custom macro body.

The three most significant points on custom macros are that variables can be used in the custom macro body, operations can be performed on variables and actual values can be assigned to the variables in custom macro instructions.

Request:

Machine tools builders are requested to attach your custom macro program tape or program list to the NC unit without fail.

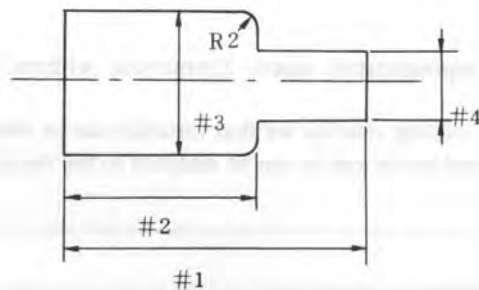
If it is necessary to replace PCB due to a failure, FANUC servicemen or end users operators in charge of maintenance should know the contents of custom macro for the purpose of repairing a trouble immediately.



Variables #1 and #4 can be used instead of the unknown traverse distance.

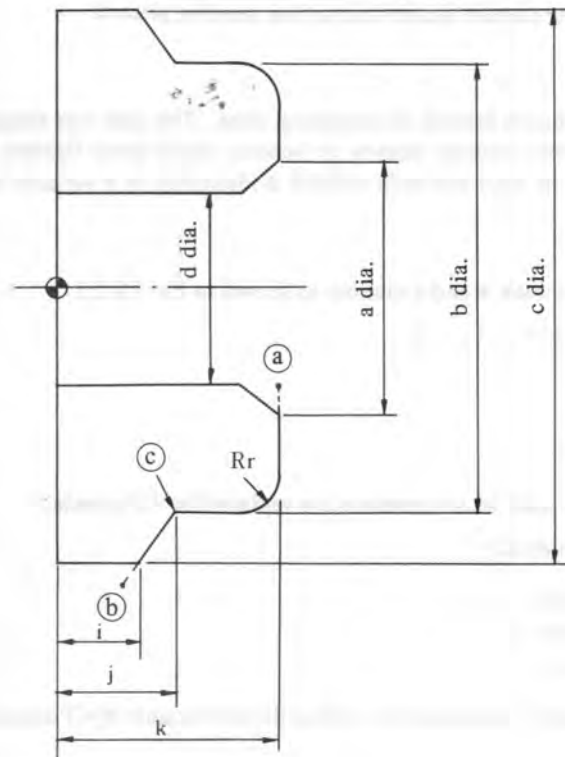
This means that a function of general use can be formed when programming a certain function as a custom macro. That is, programs can be written using variables for data that might change or be unknown. This can be further applied to group technology.

Similar workpieces can be collected in a group and a universal custom macro body can be programmed using variables that are applicable the group. In this way, programming is not required for other workpieces within the group. The programmer need only assign actual values to the variables.



Reprogramming is not required to machine the same type workpieces with different dimensions for the workpieces shown in the above diagram, if a custom macro body is programmed using variables #1 through #4. Just assign actual values to the variables.

Next we will create a custom macro to finish the following workpiece from (a) to (b).



The following program can be written.

Approach to (a) G00XdZk ;
G01XbRr ;
Zj;

Cutting from (c) to (b): Assume chamfering of up to $c + 3.0$ at (b). Incremental value w from (c) to (b) in $Z = (c - b + 3.0) * (i - j) / (c - b)$. Therefore, X $(c + 3.0) Ww$;

In practice a custom macro body uses variable #n. #n corresponds alphabetic characters as follows:

A	_____	#1
B	_____	#2
C	_____	#3
D	_____	#7
I	_____	#4
J	_____	#5
R	_____	#18
K	_____	#6
W	_____	#23

Therefore, the actual custom macro body is as follows:

A macro number is required at the beginning; M99, at the end.

```

O9801 ; The macro number is 9801
G00X#7Z#6 ;
G01X#2R#18 ;
Z#5 ;
#23=[#3-#2+3.0] * [#4-#5] / [#3-#2];
X[#3+3.0]W#23 ;
M99; End of macro

```

Now, to actually cut this workpiece from (a) to (b), values must be assigned to a, b, c, and so on. These values are specified in the custom macro instruction used to call the custom macro body.

```

G65 P9801 B120.0 C150.0 D52.0
R10.0 I20.0 J30.0 K58.0 ;

```

In the above statement, A, which is not used, is omitted. Usually, F, S, and T are specified, coordinate systems are established, and so forth before this statement is specified. To cut other workpieces of the same type with

different values, only the values in the custom macro instruction need be altered.

10.2 Variables

Variables can be used in the macro instead of numerical data. The user can assign any value (within allowed range) to them. Using variable allows custom macros to become much more flexible than the conventional sub-routines. Several variables can be used and each variable is identified by a variable number.

10.2.1 Variable expressions

A variable is composed of the code # and a number as shown in Ex. 10.2.1.

#i (i = 1, 2, 3, 4)

Example 10.2.1: #5
 #109
 #1005

Following format can also be used where numbers are replaced by <Formula>

#[<Formula>]

Example 10.2.2: #[#100]
 #[#1001-1]
 #[#6/2]

Variable #i explained hereafter can always be replaced with variable #[<Formula>].

10.2.2 Quotation variables

The numerical value following an address can be replaced with a variable.

<address> #i or <address> -#i indicates that the value of the variable or its complement is substituted for the command value of the address.

Example 10.2.3:	F#33	If #33 = 1.5, it is the same as F1.5.
	Z-#18	If #18 = 20.0, it is the same as Z -20.0.
	G#130	If #130 = 3.0, it is the same as G3.

- (1) Using a variable with addresses /, :, O and N is prohibited, i.e., :#27 or N#1 cannot be used. The value of n (n = 1 to 9) in optional block skip/n cannot be used as a variable.
- (2) A variable number cannot be replaced by a variable. When 5 in #5 is replaced with #30, it does not become ##30 but #[#30].
- (3) The value of a variable cannot exceed the maximum set for each address. For example, when #140 = 120, M#140 exceeds the maximum (an M code must be less than 99).
- (4) When a variable is used for address data, its value is rounded to the nearest.
- (5) By using <Formula> as explained later, a numerical value following the address can be also replaced with <Formula>. <address> [<Formula>] or <address> - [<formula>] indicates that the value of the <formula> or its complement is substituted for the command value of the address.
Note that a constant with no decimal point used between brackets is assumed to have a decimal point at its end.

Example 10.2.4: X[#24+#18*COS[#1]]
 Z-[#18+#26]

10.2.3 Undefined variables

The value of a variable which has not yet been defined is called <vacant>. Variable #0 is used as a variable that is always <vacant>.

An undefined variable has the following nature:

(1) Quotation

When an undefined variable is quoted, the address itself is also ignored.

When #1 = <vacant>	When #1 = 0
G90X100Z#1 ↓ G90X100	G90X100Z#1 ↓ G90X100Z0

(2) Operation

<vacant> is the same as 0 except when replaced by <vacant>.

When #1 = <vacant>	When #1 = 0
#2=#1 ↓ #2=<vacant>	#2=#1 ↓ #2=0
#2=#1*5 ↓ #2=0	#2=#1*5 ↓ #2=0
#2=#1+#1 ↓ #2=0	#2=#1+#1 ↓ #2=0

(3) Conditional expressions

<vacant> differs from 0 only for EQ and NE.

When #1 = <vacant>	When #1 = 0
#1 EQ #0 ↓ Established	#1 EQ #0 ↓ Not established
#1 NE 0 ↓ Established	#1 NE 0 ↓ Not established
#1 GE #0 ↓ Established	#1 GE #0 ↓ Established
#1 GT 0 ↓ Not established	#1 GT 0 ↓ Not established

10.2.4 Display and setting variable values

Variable values can be displayed on the CRT screen. Variable values can also be set in the MDI mode. See Chapter IV Section 5.8.2.

10.3 Types of Variables

Variables are classified into local variables, common variables and system variables, depending on the variable number. Usage and properties are different for each type.

10.3.1 Local variable #1 to #33

The local variable is a variable locally used in the macro. That is, a local variable #i used in the macro called at one point in time, is different from #i used in the macro (regardless of whether it is the same macro or different one) called at another point in time. Accordingly, when macro B is called from macro A, as in a multiplex call, a local variable used in macro A is never destroyed by being used in macro B.

A local variable is used for argument transfer. For information on the correspondence to the argument address, see section 10.7. A local variable without a transferred argument is vacant in its initial status and can be

freely used by the user.

10.3.2 Common variable #100 to #149, #500 to #509

Compared with a local variable used locally in the macro, the common variable is in common throughout the main program, each subprogram called from the main program and each macro. That is, #i used in a certain macro is the same as #i used in the other macro. Accordingly the calculated value of a common variable #i in a certain macro can be used in an other macro.

The use of common variables is not specially determined in the system. They can be freely used by the user.

Common variables #100 to 149 are cleared when the power goes off; however, variables #500 to #509 cannot be cleared by turning off the power.

10.3.3 System variables (For custom macro B)

Use of a system variable is fixed in the system.

(1) Interface signals #1000 to #1015 and #1032, #1100 to #1115 and #1132

[Input signal]

The status of the interface input signal is determined by reading the values of system variables #1000 to #1032.

System variable	Interface input signal
#1000	2 ⁰ UI 0
#1001	2 ¹ UI 1
#1002	2 ² UI 2
#1003	2 ³ UI 3
#1004	2 ⁴ UI 4
#1005	2 ⁵ UI 5
#1006	2 ⁶ UI 6
#1007	2 ⁷ UI 7
#1008	2 ⁸ UI 8
#1009	2 ⁹ UI 9
#1010	2 ¹⁰ UI 10
#1011	2 ¹¹ UI 11
#1012	2 ¹² UI 12
#1013	2 ¹³ UI 13
#1014	2 ¹⁴ UI 14
#1015	2 ¹⁵ UI 15

Value of variable	Input signal
1	Contact closed
0	Contact open

Since the variable value read is 1.0 or 0.0 regardless of the unit system, the unit system must be considered in preparing a macro.

Reading system variable #1032 is used to read all input signals at one time.

$$\#1032 = \sum_{i=0}^{15} \#[1000 + i] * 2^i$$

System variables #1000 to #1032 cannot be used as a left side term in a calculation command.

[Output signal]

Interface output signals can be issued by assigning values to system variables #1100 to #1132.

System variable	Interface output signal
#1100	2^0 U00
#1101	2^1 U01
#1102	2^2 U02
#1103	2^3 U03
#1104	2^4 U04
#1105	2^5 U05
#1106	2^6 U06
#1107	2^7 U07
#1108	2^8 U08
#1109	2^9 U09
#1110	2^{10} U010
#1111	2^{11} U011
#1112	2^{12} U012
#1113	2^{13} U013
#1114	2^{14} U014
#1115	2^{15} U015

Value of variable	Output signal
1	Contact closed
0	Contact open

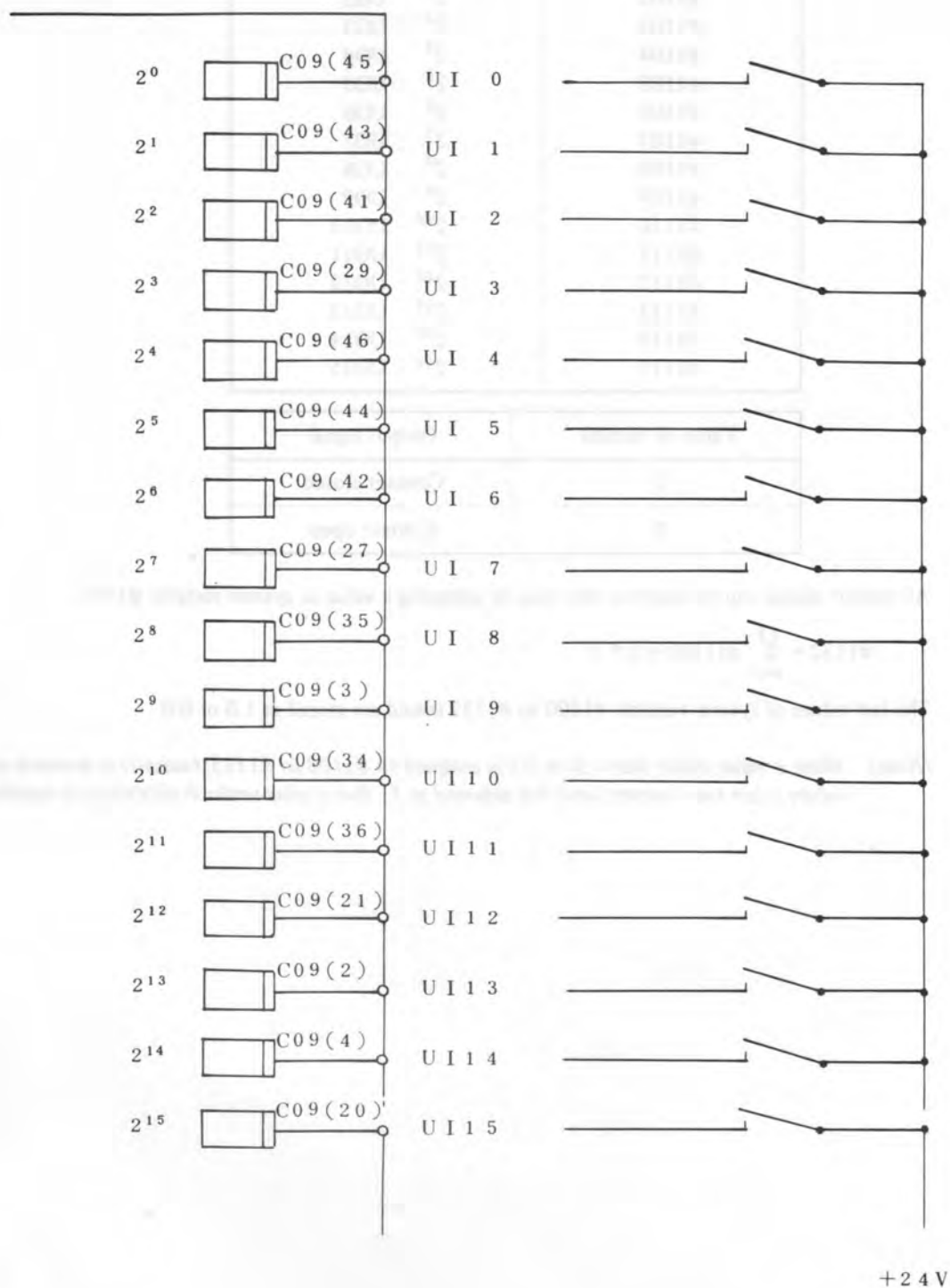
All output signals can be issued at one time by assigning a value to system variable #1132.

$$\#1132 = \sum_{i=0}^{15} \# [1100 + i] * 2^i$$

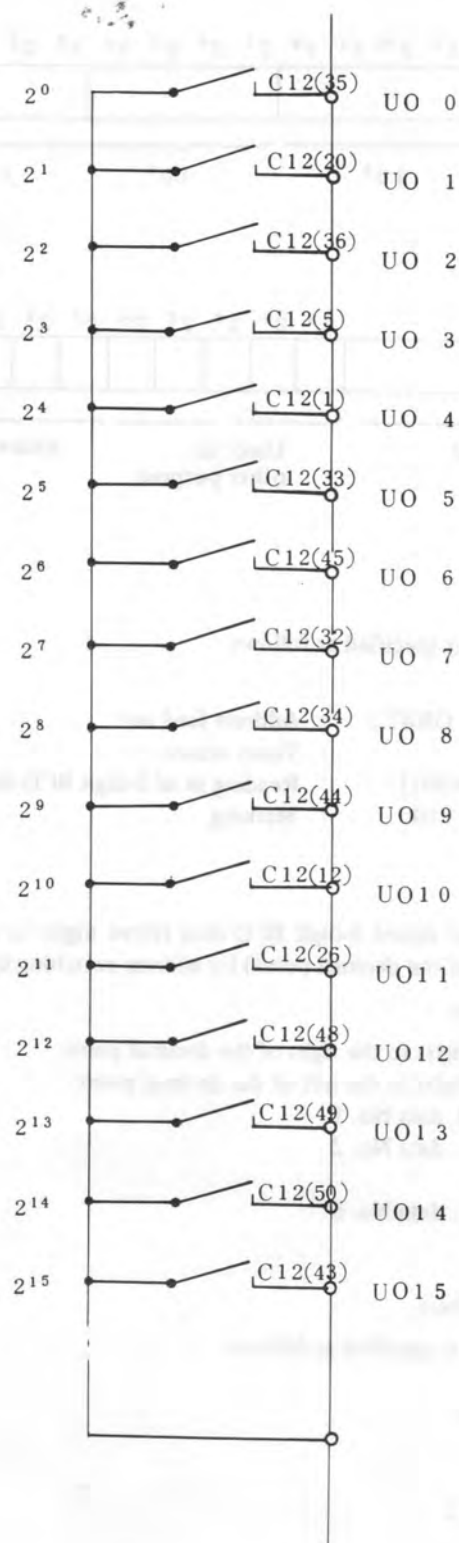
The last values of system variable #1100 to #1132 issued are stored as 1.0 or 0.0.

(Note) When a value other than 1.0 or 0.0 is assigned to #1100 to #1115, <vacant> is assumed as 0 and values other than <vacant> and 0 is assumed as 1. But a value under 0.00000001 is indefinite.

(Note 1) The following diagram shows the connection of input signals for a custom macro.

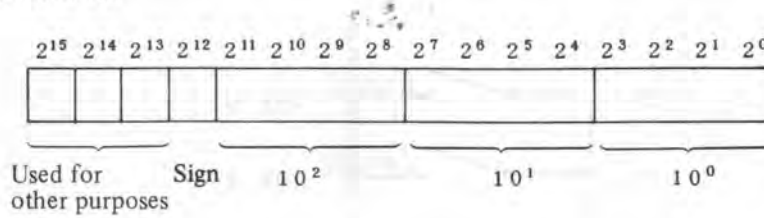


(Note 2) The following diagram shows the connection of input signals for a custom macro.

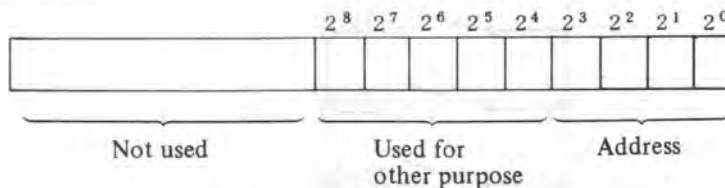


[Example 10.3.1] ① Macro used to read three digits of BCD data with a sign by address switching.

Structure of DI



Structure of DO



Macro call command

G65 P9100 D (address) ;

The custom macro body is specified as follows:

```
O 9100 ;
#1132=#1132 AND 496 OR #7 ;      : Address feed out
G65 P9101 T60 ;                  : Timer macro
#100=BIN[#1032 AND 4095] ;       : Reading in of 3-digit BCD data
LF[#1012 EQ 0] GOTO 9100 ;       : Marking
#100=-#100 ;
N9100 M99 ;
```

- ② Read eight types of signed 6-digit BCD data (three digits to the left of the decimal point + three digits to the right of the decimal point) by address switching in #101.

Machine tool composition

When DO 2⁰ = 0, three digits to the right of the decimal point
= 1, three digits to the left of the decimal point

When DO 2³ to 2¹ = 000, data No. 1
001, data No. 2
⋮
111, data No. 8

Macro call command

G65 P9101 D (data number) ;

The custom macro body is specified as follows:

```
O 9101 ;
G65 P9100 D [#7*2+1] ;
#101=#100 ;
G65 P9100 D [#7*2] ;
#101=#101+#100/1000 ;
M99 ;
```

(2) Tool geometry offset amount #2000 to #2364, tool wear offset amount #2701 to #2964

The tool offset amount and shift amount can be determined by reading the value of system variable #2000 to #2964 for the tool offset and the value of system variable #2501 and #2601 for the shift amount of the work coordinate system. The offset and shift values can also be modified by assigning a value to system variable #i.

The X-axis tool offset amount is specified as a radius value when the offset amount is designated as a radius value; as a diameter value, when specified as a diameter value.

	Tool offset number	Tool geometry offset amount	Tool wear offset amount
X	1	#2001	#2701
	?	?	?
	64	#2064	#2764
Z	1	#2101	#2801
	?	?	?
	64	#2164	#2864
R	1	#2201	#2901
	?	?	?
	64	#2264	#2964
T	1	#2301	#2301
	?	?	?
	64	#2364	#2364

Shift of work coordinate system	Shift amount specified in
X	#2501
Z	#2601

When no tool wear offset is provided, system variables for tool offset are the same as those of tool geometry offset.

#2000, #2100, #2200 and #2300 can only be read and their values are always 0.

[Example 10.3.2] #30=#2005

Tool offset amount for the X-axis value of offset number 5 is assigned to variable #30.

When the offset amount is 1.500 mm (0.1500 inch), the value of #30 becomes 1.5 (0.15).

#2210=#8

The tool nose radius compensation amount for offset number 10 is made equal to the value of variable #8.

(3) Alarm #3000

When detecting an error in the macro, an alarm can be generated. When an alarm number is specified in system variable #3000, the alarm lamp is turned on and the alarm status is entered after the preceeding block is processed.

#3000 = n (ALARM MESSAGE) ;

A number not previously used should be selected as the alarm number set in the custom macro.(n<200)

An alarm message of less than 26 characters can be specified in the section between control-out and control-in.

(4) Clock #3001, #3002

The clock time can be determined by reading the values of system variables #3001, #3002. The time can be preset by assigning a value to the system variable.

Kind	System variable	Unit time	At the time of power-on	Counting
Clock 1	#3001	1 msec.	Reset to 0	Always
Clock 2	#3002	1 hour	Same as that at the time of power-off.	While STL signal is on.

The accuracy of each clock is within 16 msec. Clock 1 is set again to zero at 65536 msec. Clock 2 continues to increase unless it is preset.

When exceeding the maximum value of 9544 hours, the time cannot be correctly determined.

[Example 10.3.3] Timer

Macro call command

G65 P9101 T (Awaiting time) msec;

This macro may be specified as follows:

```

O 9101;
#3001 = 0;           : Initial setting
WHILE [#3001 LE #20] DO1; : Wait for the prescribed length of time
END1;
M99;

```


(5) Suppression of single block stop and wait for the auxiliary function end signal

When the following values are assigned to system variable #3003, the single block stop function is suppressed and execution advances from one block to the next without awaiting the end signal (FIN) of auxiliary functions (S, T, and M). When the end signal is not awaited, the distribution end signal (DEN) is not transmitted. Be careful not to specify a subsequent auxiliary function without awaiting the end signal.

#3003	Single block stop	Auxiliary function end signal
0	Not suppressed	Awaited
1	Suppressed	Awaited
2	Not suppressed	Not awaited
3	Suppressed	Not awaited

[Example 10.3.4] Drill cycle (For incremental programming)
(Equivalent to G81 of SYSTEM 6M)

Macro call command

G65 P9081 L (Repetition time) R (R point) W (Z point);

The custom macro body is specified as follows:

O9081 ;
#3003=1 ;
G00W #18 ;
G01W #23 ;
G00W-[ROUND[#18]+ROUND[#23]] ;
#3003=0 ;
M99 ;

Single block stop is not executed. #18 corresponds to R and #23 corresponds to W.

(6) Feed hold, feedrate override, and exact stop check suppression specified in #3004

When the following values are assigned to system variable #3004 feed hold and feedrate override are suppressed for subsequent blocks and the exact stop check is not performed. Pressing the feed hold button during execution of a block for which feed hold has been suppressed:

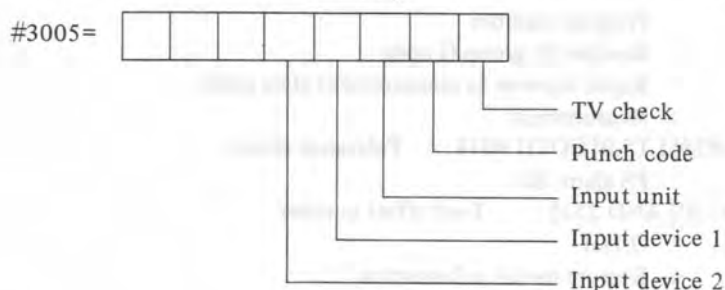
- ① Executes feed hold at the start of the first block outside the suppression range, when the button is kept pushed.
- ② Lights the feed hold lamp and does not execute feed hold at this point but at the end of the first block outside the suppression range, when the button is pushed and released.

#3004	Feed hold	Feedrate override	Exact stop check
0	○	○	○
1	×	○	○
2	○	×	○
3	×	×	○
4	○	○	×
5	×	○	×
6	○	×	×
7	×	×	×

○: Effective, X: Suppressed

(7) Variable corresponding to setting data #3005

Setting data is set by assigning a value to system variable #3005. When a value of system variable #3005 is expressed with a binary number, each digit corresponds to each setting data.



[Example]

If a command of #3005 = 5; is executed, setting data is set as follows:

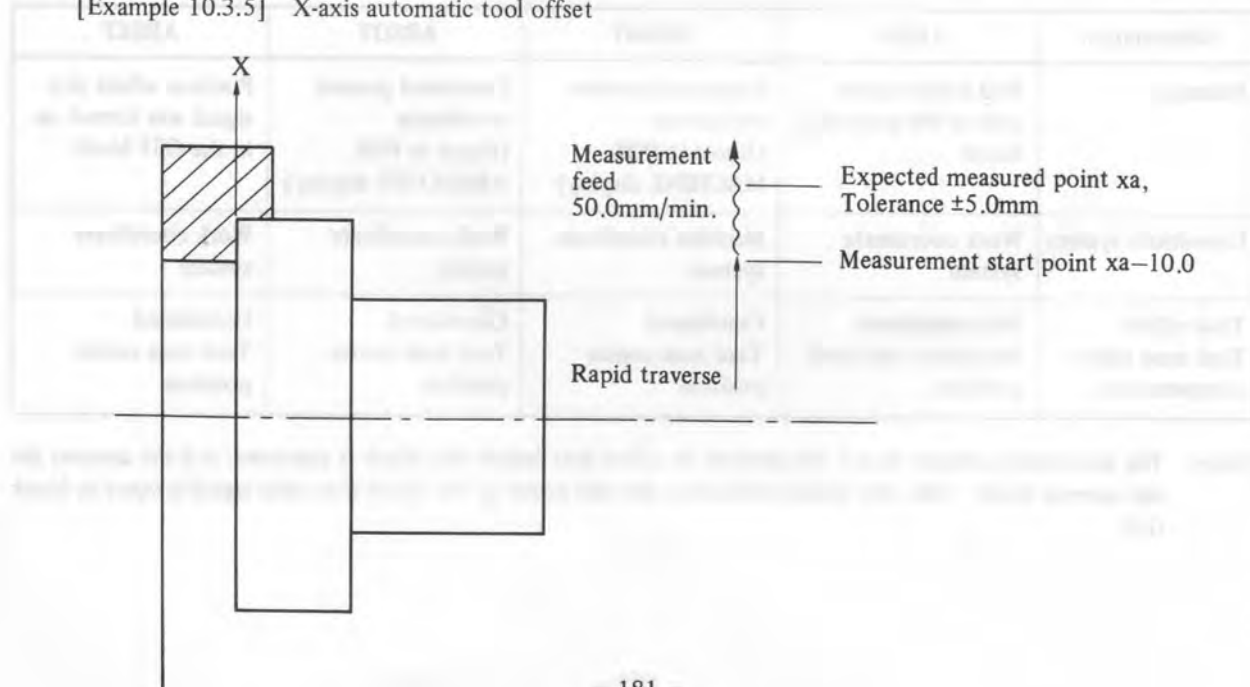
TV check = 1
Punch code = 0
Input unit = 1
Input device 1 = 0
Input device 2 = 0

(8) Modal information #4001 to #4120

Present values of modal commands can be determined by reading the values of system variables #4001 to #4120. The unit specified is effective.

System variable	Modal information
#4001	G code (group 01)
⋮	⋮
#4021	G code (group 21)
#4108	E code
#4109	F code
#4113	M code
#4114	Sequence number
#4115	Program number
#4119	S code
#4120	T code

[Example 10.3.5] X-axis automatic tool offset



Macro call command

G65 P9018 X xa ;

The custom macro body is specified as follows.

```
O 9018 ;           : Program number
#1=#4001          : Reserve 01-group G code
G00 X [#24-10.0] ; : Rapid traverse to measurement start point
G31 X [#24+5.0] ;  : Measurement
IF [ABS[#5061-#5081-#24] LT 5.0] GOTO 9018 ; : Tolerance check
#3000=80 (PS) ;    : PS alarm 80
N9018 #4=BIN [BCD [#4120] AND 255] ; : Tool offset number
G10 P#4 X [#5061-#24] ; : Offset
G #1 M99 ;         : Restore modal information
```

System variables #4001 to #4120 cannot be used as the left-hand-side term of an arithmetic command.

(9) Positional information #5001 ~ #5104

Positional information can be determined by reading system variables #5001 to #5104. This is a radius value when it is designated as a radius value; a diameter value when designated as a diameter value. The unit of positional information is mm or inches depending on the input system.

System variables #5001 to #5102 cannot be used as the left-hand-side term of an arithmetic command.

System variable	Positional information	Reading during movement
#5001 #5002	X axis block end coordinate (ABSIO) Z axis block end coordinate	Possible
#5021 #5022	X axis present coordinate (ABSMT) Z axis present coordinate	Impossible
#5041 #5042	X axis present coordinate (ABSOT) Z axis present coordinate	Impossible
#5061 #5062	X axis skip signal position (ABSKP) Z axis skip signal position	Possible
#5081 #5082	X axis tool offset amount Z axis tool offset amount	Impossible
#5101 #5102	X axis servo position deviation Z axis servo position deviation	Impossible

Abbreviation	ABSIO	ABSMT	ABSOT	ABSKP
Meaning	End point coordinate of the preceding block	Command present coordinate (Equal to POS. MACHINE display)	Command present coordinate (Equal to POS. ABSOLUTE display)	Position where skip signal was turned on in the G31 block
Coordinate system	Work coordinate system	Machine coordinate system	Work coordinate system	Work coordinate system
Tool offset Tool nose radius compensation	Not considered Imaginary tool nose position	Considered Tool nose center position	Considered Tool nose center position	Considered Tool nose center position

(Note) The tool offset amount is not the amount in effect just before the block is executed; it is the amount for the current block. The skip signal position is the end point of the block if no skip signal is input in block G31.

[Example 10.3.6]

The tool moves to a point appropriate to the machine tool (xp, zp distant from a reference point) through a programmed intermediate point and after processing a sequence of operations, returns to the original point.

Macro call command

G65 P9300 X (intermediate point) Z (intermediate point);

The custom macro body is specified as follows:

```
O 9300 ;
#1=#5001 ;
#2=#5002 ;
G00 X#24 Z#26 ;
G04 ;
U[xp-#5021] W[zp-#5022] ;
.
.      (processing)
.
X#24 Z#26 ;
X#1 Z#2 ;
M99 ;
```

10.4 Arithmetic Commands

A variety of arithmetic operations can be performed on variables. An arithmetic command must be specified the same as in general arithmetic expressions.

#i = <Formula>

<Formula>, the right-hand-side of an arithmetic command is a combination of constants, variables, functions and operators. A constant can be used instead of #i and #k. A constant without a decimal point used in <Formula> is considered to have a decimal point at the end.

10.4.1 Definition and substitution of variables

#i=#j Definition, substitution

10.4.2 Addition arithmetic

#i=#j+#k	Sum
#i=#j-#k	Subtraction
#i=#j OR #k	Logical sum (at every bit of 32 bits)
#i=#j XOR #k	Exclusive OR (at every bit of 32 bits)

10.4.3 Multiplication arithmetic (User macro B option)

#i=#j*#k	Product
#i=#j/#k	Quotient
#i=#j AND #k	Logical product (at every bit of 32 bits)

10.4.4 Functions (User macro B option)

#i=SIN [#j]	Sine (degree unit)
#i=COS [#j]	Cosine (degree unit)
#i=TAN [#j]	Tangent (degree unit)
#i=ATAN [#j]/[#k]	Arctangent (degree unit)
#i=SQRT [#j]	Square root
#i=ABS [#j]	Absolute value
#i=BIN [#j]	Conversion from BCD to BIN
#i=BCD [#j]	Conversion from BIN to BCD
#i=ROUND [#j]	Rounding off
#i=FIX [#j]	Discard fractions less than 1
#i=FUP [#j]	Add 1 for fractions less than 1

(Note) How to use function ROUND

- (1) If function ROUND is employed in an arithmetic operation command or in an IF or WHILE conditional expression, the figure in function ROUND is rounded (counting fractions of 5 and over as a unit and disregarding the rest) as ordinary data with a decimal point.

(Example)

```
#1 = ROUND [1.2345];
#1 becomes 1.0.
IF [#1 LE ROUND [#2]] GOTO 10;
ROUND [#2] is 4.0 if #2 is 3.567.
```

- (2) If function ROUND is employed in a command to an address, it is rounded to the least input increment of the address.

(Example)

```
G01 X[ROUND[#1]];
If #1 is 1.4567 and the least input increment of X is 0.001, this block becomes G01 X 1.457.
In this example, this command is the same as G01 X #1; command.
Function ROUND in an address command is mainly used in the following case.
```

(Example)

[Program to move incrementally by #1 and #2 only and then return to the starting point]

```
N1 #1 = 1.2345;
N2 #2 = 2.3456;
N3 G01 X #1 F100; : X moves 1.235.
N4 X #2; : X moves 2.346.
N5 X -[#1 + #2]; : X moves -3.58, since #1 + #2 is 3.5801.
```

Since $1.235 + 2.346 = 3.581$, the program does not return to the starting point by N5.

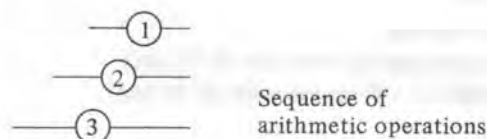
Assume $N5 X -[ROUND[#1] + ROUND[#2]]$;

It becomes equal to $N5 X -[1.235 + 2.346]$; and the program returns to the starting point.

10.4.5 Combination of arithmetic operations

The above arithmetic operations and functions can be combined. Priority in an arithmetic operation is in the order of function, multiplication arithmetic then addition arithmetic.

[Example 10.4.1] $\#i = \#j + \#k * SIN [\#l]$

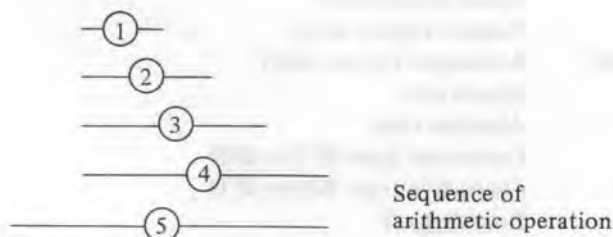


10.4.6 Modification of arithmetic sequence using []

A portion to be assigned priority in an arithmetic sequence can be enclosed in [].

[] can be nested up to five times (including [] used in functions).

[Example 10.4.2] $\#i = SIN[[[#j + \#k] * \#l + \#m] * \#n]$ (Three fold nesting)



10.4.7 Precision

Always consider the precision of a custom macro function used for preparation of programs.

(1) Data format

Numeric data handled by a custom macro is in floating decimal point format as follows:

$M*2^E$

where M: 1-bit sign + 31-bit binary data

E: 1-bit sign + 7-bit binary data

(2) Operational precision

An operation executed once generates the following error. These errors are accumulated with each repeated operation.

Operation format	Average error	Maximum error	Type of error
$a = b*c$	1.55×10^{-10}	4.66×10^{-10}	Relative error $\left \frac{\epsilon}{a} \right $
$a = b/c$	4.66×10^{-10}	1.86×10^{-9}	
$a = \sqrt{b}$	1.24×10^{-9}	3.73×10^{-9}	
$a = b + c$ $a = b - c$	2.33×10^{-10}	5.32×10^{-10}	$\min. \left(\left \frac{\epsilon}{b} \right , \left \frac{\epsilon}{c} \right \right)$
$a = \text{SIN } b$ $a = \text{COS } b$	5.0×10^{-9}	1.0×10^{-8}	Absolute error $ \epsilon $ degrees
$a = \text{ATAN } b/c$	1.8×10^{-6}	3.6×10^{-6}	

(Note) Function TAN performs SIN/COS.

10.4.8 Notes on decreased precision

(1) Addition and subtraction

Note that when absolute values are used subtractively in addition or subtraction, the relative error cannot be held under 10^{-8} . For example, suppose that the real values of #1 and #2 are as follows.

#1 = 9876543210123.456

#2 = 987654327777.777

Performing operation #2 - #1 does not produce.

#2 - #1 = 67654.321,

since the custom macro has a precision of only eight decimal digits, the values of #1 and #2 have a precision as low as approximately

#1 = 9876543200000.000

#2 = 9876543300000.000, respectively. (Strictly, the internal values differ somewhat from the above values because they are binary numbers.)

Consequently,

#2 - #1 = 100000.000

which generates a large error.

(2) Logical operation

EQ, NE, GT, LT, GE, and LE are basically the same as addition and subtraction. Therefore, be careful of errors. To determine whether or not #1 and #2 are equal in the above example, for example,

IF [#1 EQ #2]

is not always evaluated correctly. When the error is evaluated as in

IF [ABS [#1 - #2] LT 50000]

and the difference between #1 and #2 falls within the range error, both values must be considered equal.

(3) Trigonometric functions

Absolute errors occur in trigonometric functions, but since they are not under 10^{-8} be careful in integration or division after using a trigonometric functions.

10.5 Control Command

The program flow can be controlled by using the following commands.

10.5.1 Divergence (GOTO)

IF [<Conditional expression>] GOTO n

When <conditional expression> is satisfied, the next operation is executed in the block with sequence number n in the same program. Sequence number n can be replaced by a variable or <Formula>.

When the condition is not satisfied, control proceeds to the next block.

IF [<conditional expression>] can also be omitted, and when it is, control passes to block n unconditionally.

The following expressions can be used for <conditional expression>.

#j EQ #k	=
#j NE #k	≠
#j GT #k	>
#j LT #k	<
#j GE #k	≥
#j LE #k	≤

<Formula> can be used instead of #j and #k. And a variable or <Formula> can be used instead of n.

(Note) In the block with sequence number n which will be executed after a GOTO n command, the sequence number must always be at the top of the block. And passing control in the reverse direction takes longer time than passing it in the forward direction.

10.5.2 Iteration (Custom macro B option)

WHILE [<conditional expression>] DO m (m = 1, 2, 3)

END m

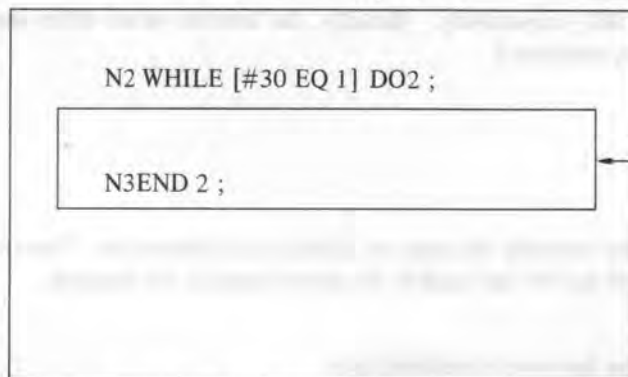
While <conditional expression> is satisfied, blocks DO m to END m are repeatedly executed. That is, the condition of DO m block is examined and when the <conditional expression> is satisfied control passes to the next block. But when it is not satisfied, the block after END m is executed.

WHILE [<conditional expression>], can also be omitted as with IF, and if omitted, blocks from DOm to ENDm are executed eternally.

WHILE [<conditional expression>] DO m and END m must be used as a pair, and identification number m identifies this pair.

Example 10.5.1

```
#120 = 1 ;  
N1 WHILE [#120 LE 10] DO 1 ;
```



← Repeated ten times

← Repeated while variable #30 = 1

```
#120 = #120 + 1 ;  
N4END 1 ;
```

(Note) Notes on REPEAT programming.

10

- ④ *DO statement can be nested up to three times.*

```

      .
      .
DO 1 ;
      .
      .
DO 2 ;
      .
      .
DO 3 ;
      .
      .
      (Yes)
END 3 ;
      .
      .
END 2 ;
      .
      .
END 1 ;
      .
      .

```

- ⑤ *DO ranges cannot be intersected.*

```

      .
      .
DO 1 ;
      .
      .
DO 2 ;
      .
      .
      (No)
END 1 ;
      .
      .
END 2 ;
      .
      .

```

- ⑥ *A branch can be made from inside to outside a DO range.*

```

      .
      .
DO 1 ;
      .
      .
GOTO 9000 ;
      .
      .
      (Yes)
END 1 ;
      .
      .
N9000 ..... ;
      .
      .

```

- ⑦ A branch cannot be made from outside to inside a DO range.

```

      .
      .
      .
GOTO 9000 ;
      .
      .
DO 1 ;      (No)
      .
      .
N9000 .....;
      .
      .
END 1 ;
      .
      .
DO 1 ;
      .
      .
N9000 .....;
      .
      .      (No)
END 1 ;
      .
      .
GOTO 9000 ;
      .
      .

```



- ⑧ Custom macro bodies or subprograms can be called from inside a DO range. DO statements can be nested up to three times more in the custom macro body or in the subprogram.

```

      .
      .
DO 1 ;
      .
      .
G65 .....;      (Yes)
      .
      .
G66 .....;      (Yes)
      .
      .
G67 ;      (Yes)
      .
      .
END 1 ;
      .
      .
      .
      .
DO 1 ;
      .
      .

```



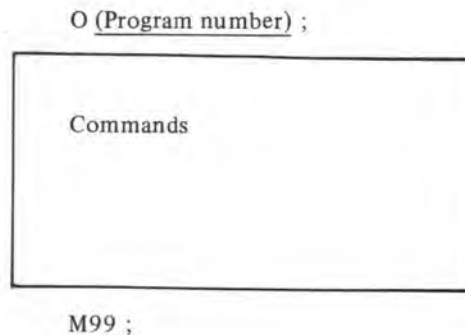
M98.....; (Yes)

END 1 ;

10.6 Creation and Registration of Custom Macro Body

10.6.1 Creation of custom macro body

The format of a custom macro body is the same as that of a subprogram as shown below.



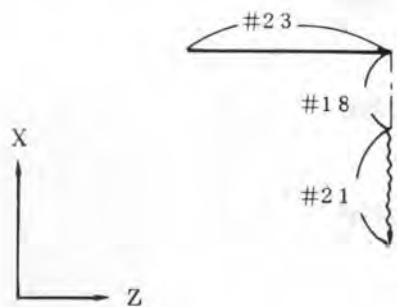
Program numbers are determined as follows:

- (1) O1 ~ O7999 for programs that can be freely registered, cancelled and edited.
- (2) O8000 ~ O8999 for programs that cannot be registered, cancelled, or edited without setting.
- (3) O9000 ~ O9019 for special call-type macros
- (4) O9020 ~ O9899 for programs that cannot be registered, cancelled or edited without setting a parameter
- (5) O9900 ~ O9999 for robot operating programs

Dummy arguments (variables used to accept data from a macro call command) are fixed, i.e. the address which indicates a parameter in a macro call command and a variable in the custom macro body correspond to each other.

[Example 10.6.1]

```
O 9081 ;
G00W#23 ;
U#18 ;
G01U #21 ;
G00U-[#18+#21] ;
M99 ;
```



10.6.2 Registration of custom macro body

A custom macro body is a sort of subprogram, and it is registered and edited in the same way as a subprogram. The memory used to register custom macro bodies is included in the storage capacity of NC.

10.7 Macro Calls

A macro can be called simply from a single block, or modally from each block in the call mode.

10.7.1 Simple calls

When the following command is executed, the custom macro body identified by P (program number) is called.

G65 P (program number) L (iteration times) <argument assignment>;

When it is necessary to transfer arguments to a custom macro body, the argument is specified by <argument assignment>. The following two types of <argument assignment> can be specified. The argument mentioned here is the actual numerical value assigned to a variable.

(Note) G65 must be specified before arguments in the G65 block. Negative sign and decimal point can be used regardless of addresses in <argument assignment>.

(1) Argument assignment I

A ____ B ____ C ____ D ____ Z ____

An argument can be assigned for all address except G, L, N, O, and P. Assignment need not be made in alphabetical order. Specification is made according to word address format. Addresses not required may be omitted.

However, when I, J, and K are used, assignment must be made in the alphabetical order.

B ____ A ____ D ____ I ____ K ____ valid

B ____ A ____ D ____ J ____ I ____ invalid

Addresses assigned in argument assignment I and the number of the variable in custom macro body correspond as follows:

Address of the argument assignment I	Variable in custom macro body
A	#1
B	#2
C	#3
D	#7
E	#8
F	#9
H	#11
I	#4
J	#5
K	#6
M	#13
Q	#17
R	#18
S	#19
T	#20
U	#21
V	#22
W	#23
X	#24
Y	#25
Z	#26

(2) Argument assignment II

A ____ B ____ C ____ I ____ J ____ K ____ I ____ J ____ K ____

In addition to the fact that arguments can be assigned in addresses A, B, and C, a maximum of ten sets of arguments can be set for addresses I, J and K. When several numbers are assigned in the same address, they must be assigned in the determined sequence.

Addresses not required can be omitted.

Addresses assigned in argument assignment II and the number of the variable used in the macro correspond as follows:

Address of argument assignment II	Variable in custom macro body
A	#1
B	#2
C	#3
I ₁	#4
J ₁	#5
K ₁	#6
I ₂	#7
J ₂	#8
K ₂	#9
I ₃	#10
J ₃	#11
K ₃	#12
I ₄	#13
J ₄	#14
K ₄	#15
I ₅	#16
J ₅	#17
K ₅	#18
I ₆	#19
J ₆	#20
K ₆	#21
I ₇	#22
J ₇	#23
K ₇	#24
I ₈	#25
J ₈	#26
K ₈	#27
I ₉	#28
J ₉	#29
K ₉	#30
I ₁₀	#31
J ₁₀	#32
K ₁₀	#33

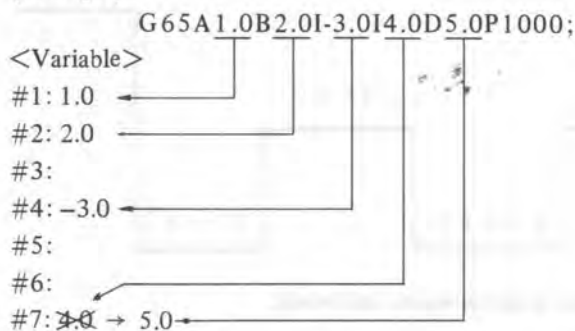
Suffixes 1 to 10 of I, J and K indicate the sequence of the assigned set.

(3) Coexistence of argument assignment I and II

No alarm is generated even if arguments of both assignment I and II are specified in the same block with a G65 command.

If an argument of type I and an argument of type II are specified to the same variable, the argument specified later is effective.

(Example)



In this example, even if arguments I4.0 and D5.0 are specified to variable #7, the latter is effective.

10.7.2 Modal calls

When the following command is executed, macro call mode is assigned. During macro call mode, the specified macro is called each time a motion command is executed.

G66 P (program number) L (iterating times) <argument assignment>;

<argument assignment> is the same as for a simple call.

When the following command is executed, the macro call mode is cancelled.

G67 ;

(Note) In the G66 block, G66 must be specified before all arguments. Negative sign and decimal point can be used regardless of addresses in <argument assignment>.

10.7.3 Multiplex calls

Similarly to a subprogram called from another subprogram, a macro can be called from another macro. The multiplicity should be less than or equal to four including simple and modal calls.

10.7.4 Multiplex modal calls

In modal calls, the specified macro is called each time a motion command is executed. When several modal macros are specified, the next macro is called each time a motion command in the first macro is executed. Macros are successively called from those assigned later.

[Example 10.7.5]

```

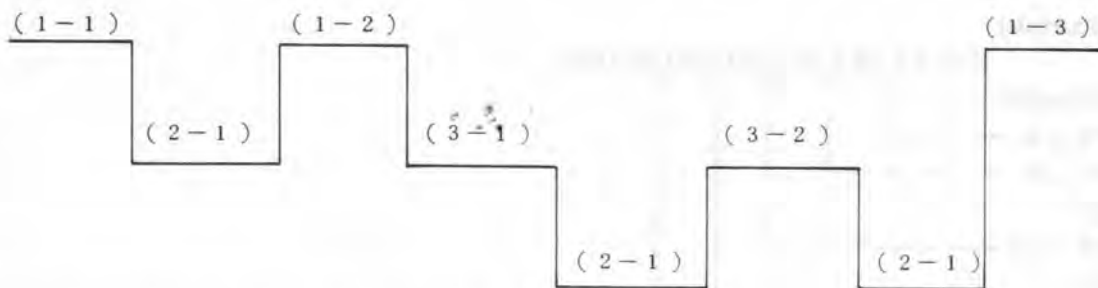
G66 P9100 ;
Z 10000 ;           (1-1)
G66 P9200 ;
Z 15000 ;           (1-2)
G67 ;               : P9200 cancelled
G67 ;               : P9100 cancelled
Z-25000 ;           (1-3)

O 9100 ;
X 5000 ;             (2-1)
M99 ;

O 9200 ;
Z 6000 ;             (3-1)
Z 7000 ;             (3-2)
M99 ;

```

Sequence of execution (A block without a motion command must be defaulted.)



(Note) A modal macro is not called after (1-3), which is not in macro call mode.

10.7.5 Macro call using G codes

A G code can be set by parameter to call a macro. That is, instead of specifying N ____ G65P△△△△<argument assignment>; the following simple command can be used.

N ____ Gxx <argument assignment>;

The correspondence between the calling G code xx and the program number △△△△ of the called macro must be set as a parameter.

A calling G code xx and the program number △△△△ of a called macro are set in a parameter.

Up to ten G01 to G255 commands can be used to call macros. G00 cannot be used. These codes cannot be specified from the MDI panel the same as with G65. These G codes cannot be specified in a macro called with a G code. These G codes cannot be specified in a subprogram called with an M code or a T code.

Set the following parameters.

0	3	2	3
---	---	---	---

G code used to call macro : 9010

0	3	2	4
---	---	---	---

G code used to call macro : 9011

⋮

⋮

0	3	3	2
---	---	---	---

G code used to call macro : 9019

10.7.6 Subprogram call with M code

An M code can be set by parameter to call a subprogram. That is, instead of N ____ G ____ X ____ Y ____ M98P △△△△ ;

the following simple command can be specified.

N ____ G ____ X ____ Y ____ Mxx ;

As for M98, the subprogram is displayed on the COMND page, but MF and M codes are not transmitted.

The correspondence between the calling M code xx and the program number △△△△ of the called subprogram must be set as a parameter.

Up to three M03 to M97 commands can be used for a macro call except for M30, M30 (parameter number 35) and M30 (parameter number 36).

This command can be specified from MDI keyboard, but no arguments can be entered. When these M codes are specified in a macro called with a G code or in a subprogram called with an M code or a T code, the subprogram is not called, but these M codes are treated as ordinary M codes.

Set the following parameters.

block stop, a G65 block does not.

- (4) G65 changes the level of local variable; M98 does not. That is, #1 specified before G65 is one thing and #1 in the calling custom macro body is another. #1 specified before M98 is the same as #1 in the calling subprogram.
- (5) Up to four G65 calls, including G66, can be made in addition, M98 calls can be made (with custom macro A or B option) up to four calls.
- (6) When inserting an operation from MDI during automatic operation, up to four M98 calls can be made in TAPE mode or in MEMORY mode, and separately up to four calls in MDI mode; up to four G65 calls can be made in all modes in common.

10.8 Relation to Other Functions

(1) MDI operation

The macro call command, arithmetic command and control command can not be specified in MDI mode.

MDI commands (except those relating to the macro) can be entered midway during execution of the macro even with a single block stop.

A macro is not called even in macro call mode (G66) if a move command is input from the MDI.

(2) Sequence number search

Sequence numbers in the custom macro body cannot be searched for.

(3) Single block

A block other than one including a macro call command, arithmetic command or control command can be executed as a single block stop even when in the macro.

A block including a macro call command (G65, G66, G67), arithmetic command and control command is not stopped even by single block operation.

However, the following settings or parameter setting executes a single block stop for commands other than macro call commands. This function is used for testing a custom macro body.

0	3	1	8
---	---	---	---

	MCS9						
--	------	--	--	--	--	--	--

0	3	1	9
---	---	---	---

	MCS8						MCS7
--	------	--	--	--	--	--	------

When MCS7 = 1, a macro statement in programs numbered O1 to 7999 and O9900 to 9999 executes a single block stop.

When MCS8 = 1, a macro statement in programs numbered O8000 to 8999 executes a single block stop.

When MCS9 = 1, a macro statement in programs numbered O9000 to 9089 executes a single block stop.

But a block in which a macro statement has executed a single block stop, is assumed to have no movement in the tool nose R compensation mode. Sometimes incorrect compensation is performed. (Strictly, the case is handled the same as for a movement amount of 0 although a move has been specified.)

(4) Optional block skip

/, appearing in the middle of <expression> (right-hand-side of an arithmetic formula, or in []), is treated as division, and not an optional block skip.

(5) Operation in EDIT mode

The following settings can be used to protect registered custom macro bodies and subprograms from accidental destruction.

0	3	1	8
---	---	---	---

PRG9							
------	--	--	--	--	--	--	--

0	3	1	9
---	---	---	---

PRG8							
------	--	--	--	--	--	--	--

(Setting operation is effective)

Setting PRG8 and PRG9 to 1 specifies that custom macro bodies or subprograms with program numbers 8000 to 8999 and 9000 to 9899 cannot be registered, cancelled, and edited, respectively.

However, this does not affect the deletion of all programs by turning on the power or the punching of a program.

(6) PRGRM page display in other than EDIT mode

Generally, when a custom macro body or a subprogram is called, the called program is displayed. The following setting is used to keep displaying the previous program.

0	3	1	8
---	---	---	---

		MPD9					
--	--	------	--	--	--	--	--

0	3	1	9
---	---	---	---

		MPD8					
--	--	------	--	--	--	--	--

Setting MPD8 and MPD9 to 1 specifies that custom macro bodies or subprograms with program numbers 8000 to 8999 and 9000 to 9899 cannot be displayed on the PROGRAM page in other than EDIT mode, respectively.

(7) Reset

When a cleared status is produced by a reset, all local variables and common variables #100 to #149 become <vacant>.

Performing a reset in other than MDI mode clears the called status and the DO status of custom macro body or of a subprogram, returning control to the main program. Performing a reset in MDI mode clears only the program called in MDI mode.

(8) Macro and NC statements

The following blocks are called macro statements.

- ① Operation command (block including =)
- ② Control command (block including GOTO, DO or END)
- ③ Macro call command (block including G65, G66, G67 or G codes calling macro)

Blocks other than macro statements are sometimes referred to as NC statements.

(9) MDI operation during automatic operation

If a macro is called while inserting an operation from MDI during automatic operation the degrees of multi-macro calling and of multiple DO statements are counted as continuing from the automatic operation, the former and the latter being up to four degrees and up to three degrees in total respectively. Up to four subprogram calls can be made in MDI mode.

(10) Multiple repetitive cycle

Macro statements (control command, operation command, macro call command) cannot be used in a multiple repetitive cycle (G71, G72, G73).

(11) Display in PROGRAM RESTART page

The M and T codes used for subprogram calling are not displayed like M98.

10.9 Special Codes and Words Used in Custom Macro

The following code can be used in the program of the custom macro in addition to codes used in conventional programs.

(1) ISO

Meaning	8	7	6	5	4	3	2	1	Character
[	○	○		○	○	°		○	[
]	○	○		○	○	°	○		]
#	○		○			°		○	#
*	○		○		○	°		○	*
=	○		○	○	○	°	○		=
O	○	○			○	°	○	○	O
+			○		○	°		○	+

(2) EIA

Meaning	8	7	6	5	4	3	2	1	Character
[				○	○	○			
]		○			○	○			
#				Parameter					
*					○	○			
=			○	○	○		○	○	&
+		○	○	○					,
									+

O, the same code O as in the program number must be used. The hole pattern for # in EIA code must be set as a parameter. However, the character with no punched hole cannot be used. Note that alphabetic characters can be used, but when used as #, they are not used in their proper sense.

Parameter number											
0	3	1	7								

↑
Sprocket hole

Special words used in user macro A are:

OR, XOR, IF, GOTO, EQ, NE, GT, LT, GE, and LE.

Special words used in user macro B, in addition to those above, are:

AND, SIN, COS, TAN, ATAN, SQRT, ABS, BIN, BCD, ROUND, FIX, FUP, WHILE, DO, and END.

10.10 Limitations

(1) Usable variables

#0, #1 to #33, #100 to #149, #500 to #509, system variables.

(2) Valid variable values

Maximum value $\pm 10^{47}$

Minimum value $\pm 10^{-29}$

(3) Constant value valid in <Expression>

Maximum value ± 99999999

Minimum value ± 0.0000001

Decimal point can be used.

(4) Arithmetic precision

Eight-digit decimal number.

(5) Macro nesting

Maximum of four.

(6) Iteration identification number

1 ~ 3

(7) Nesting of []

Maximum of five.

(8) Subprogram nesting

Maximum of four.

(9) Functions available from custom macro A

Custom macro B can perform everything described above, but custom macro A can perform only the following:

- (i) Can use variables other than system variables.
- (ii) Can operate using the following variables: +, -, OR, XOR
- (iii) Can use IF [<conditional expression>] GOTO n.
- (iv) Can use simple and modal calls.

10.11 P/S Alarm Explanation

The following alarms supplement the alarms listed in Appendix 7.

- ① Alarm number 004
An appropriate address was not found at the proper position.

Example
X1*1
X1, a word, should be followed by an address but was followed by *, generating alarm 004.
- ② Alarm number 114
A format other than <Formula> is invalid. This alarm occurs under one of the following conditions:
 - (a) The character following an address is not a numeral, ., -, #, [, and +.

Example: XF1000 ;
XSIN[10] ;
 - (b) The form is not: IF (or WHILE) [<formula> ΔΔ <formula>]

Example: [IF #1 EQ #2 GOTO 10 ;
WHILE [#1 SIN #2] DO 1 ;

10.12 External Output Commands

The following macro commands can be executed in addition to the standard custom macro commands. (These commands are called external output commands.)

- (a) B P R N T
- (b) D P R N T
- (c) P O P E N
- (d) P C L O S

These commands are provided to output variable values and characters through the RS232C interface. Specify these commands according to the following procedure.

(1) Open command: POPEN

The connection processing to external I/O units is made before executing a series of data output commands.

(2) Data output command: BPRNT or DPRNT

Necessary data output commands are executed.

(3) Close command: PCLOS

This command is specified after all data output commands have been completed to disconnect external I/O units.

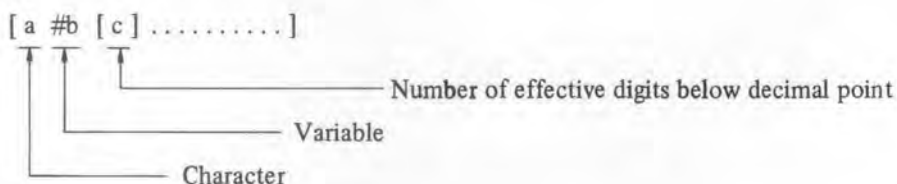
10.12.1 Open command POPEN

POPEN:

This command is provided to connect to external I/O units, and it is specified before sending a series of data output commands. The DC2 control code is output from NC.

10.12.2 Data output commands BPRNT, DPRNT

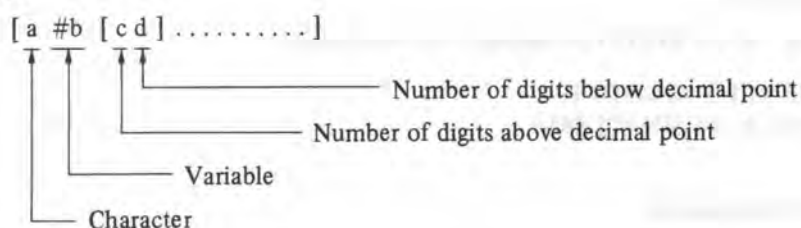
(1) BPRNT



Characters are output and variable values are binary-output when BPRNT is given.

- (a) The specified characters are output by ISO codes.
The following characters are commandable.
* Alphabetic characters (A ~ Z)
* Numeric characters
* Special characters (* , / , + , -)
Asterisk (*) is output by the space code.
- (b) Since all variables are being stored with a decimal point, the number of effective digits below decimal point must be specified by parenthesizing it just after specifying variables.
Variable values are treated as 2-word (32-bit) data by taking the number of digits below decimal point into account, and they are output as binary data, starting with high-order bytes.
- (c) EOB code is output by ISO code after command data output.
- (d) No "vacant" variable is outputtable (otherwise P/S alarm No. 114 occurs).

(2) DPRNT



When DPRNT is specified, characters and numeric characters every variable value digit are output by ISO codes.

- (a) Same as in (a), (c), (d) of BPRNT command
- (b) For outputting variable values, specify the variable number following #, and then, specify the number of digits above decimal point and the number of digits below decimal point by parenthesizing these values.
Variable values are output by the specified number of digits every digit starting with higher significant digit by ISO codes. The decimal point is also output by ISO code.
A variable value is regarded as being composed of max. 8-digit numeric value. If the higher significant digits are 0, no code is output when parameter No. 315 PRT=1 and the space code is output when PRT=0.
Whenever the number of digits below decimal point is other than 0, the numeric value below decimal point is output. When the number of digits below decimal point is 0, the decimal point is not output. For the plus (+) code in case of positive sign, the space code is output when parameter No. 315 PRT=0, but no code is output when PRT=1.

[Example 1]

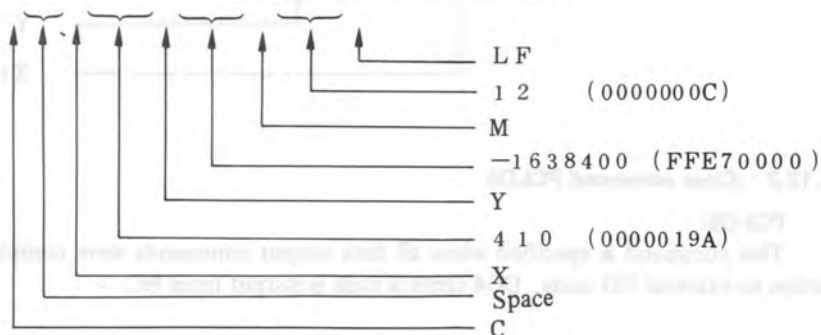
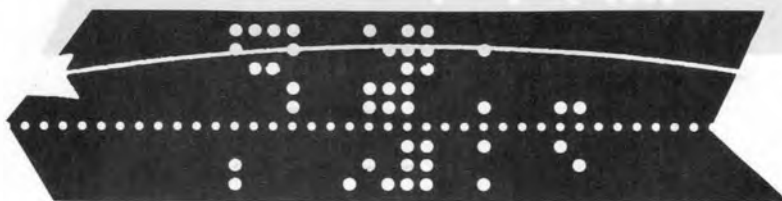
BPRNT [C * * X # 100 [3] Y # 1 0 1 [3] M # 1 0 [0]]

Variable value

100 = 0.40956

101 = -1638.4

10 = 12.34



[Example 2]

DPRNT [X # 2 [5 3] Y # 5 [5 3] T # 3 0 [2 0]]

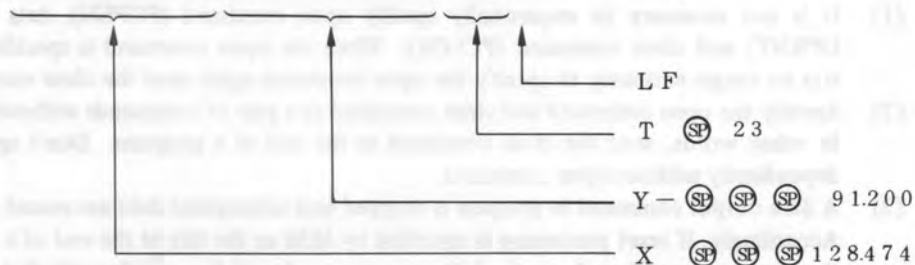
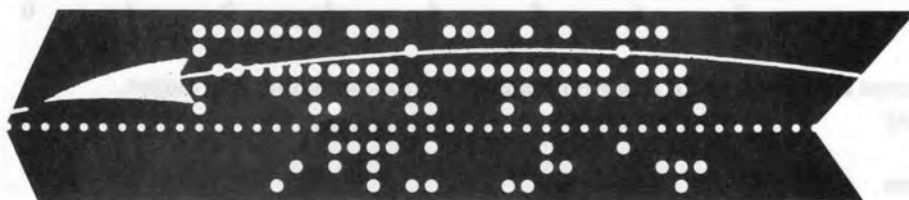
Variable value

2 = 128.47398

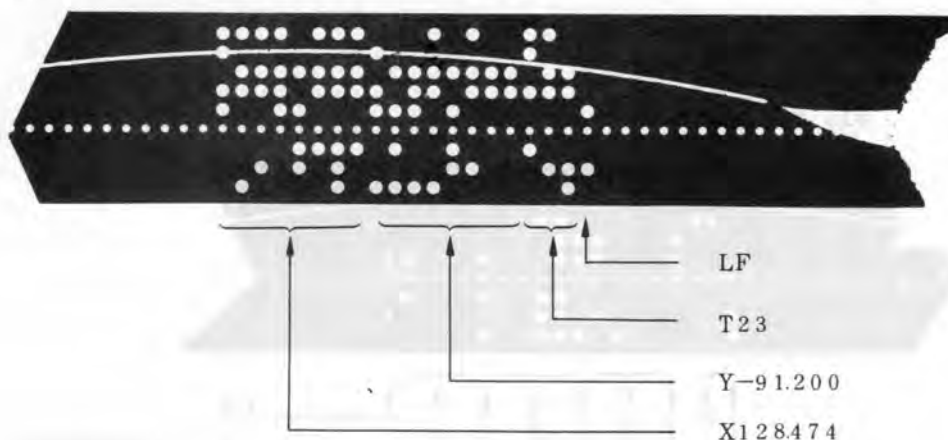
5 = -91.2

30 = 123.456

① Parameter PRT = 0



② Parameter PRT = 1



10.12.3 Close command PCLOS

PCLOS:

This command is specified when all data output commands were completed to release the processing connection to external I/O units. DC4 control code is output from NC.

10.12.4 Setting required for using this function

- (1) Set parameter No. 341 so that the output unit RS232C is employed for punch-out. However, the output to FANUC cassette is not allowable in this case.
- (2) Set various data of RS232C (baud rate, etc.) to one of parameter No. 310 to 313 according to the number of the output units preset to the above parameter No. 341.
- (3) Set output codes to ISO codes.
- (4) Set parameter No. 315 to determine whether leading zeroes are spaced or not when data are output by DPRNT command.

0	3	1	5
---	---	---	---

PRT							
7	6	5	4	3	2	1	0

Leading zeroes are treated by PRT DPRNT command as follows during data output.

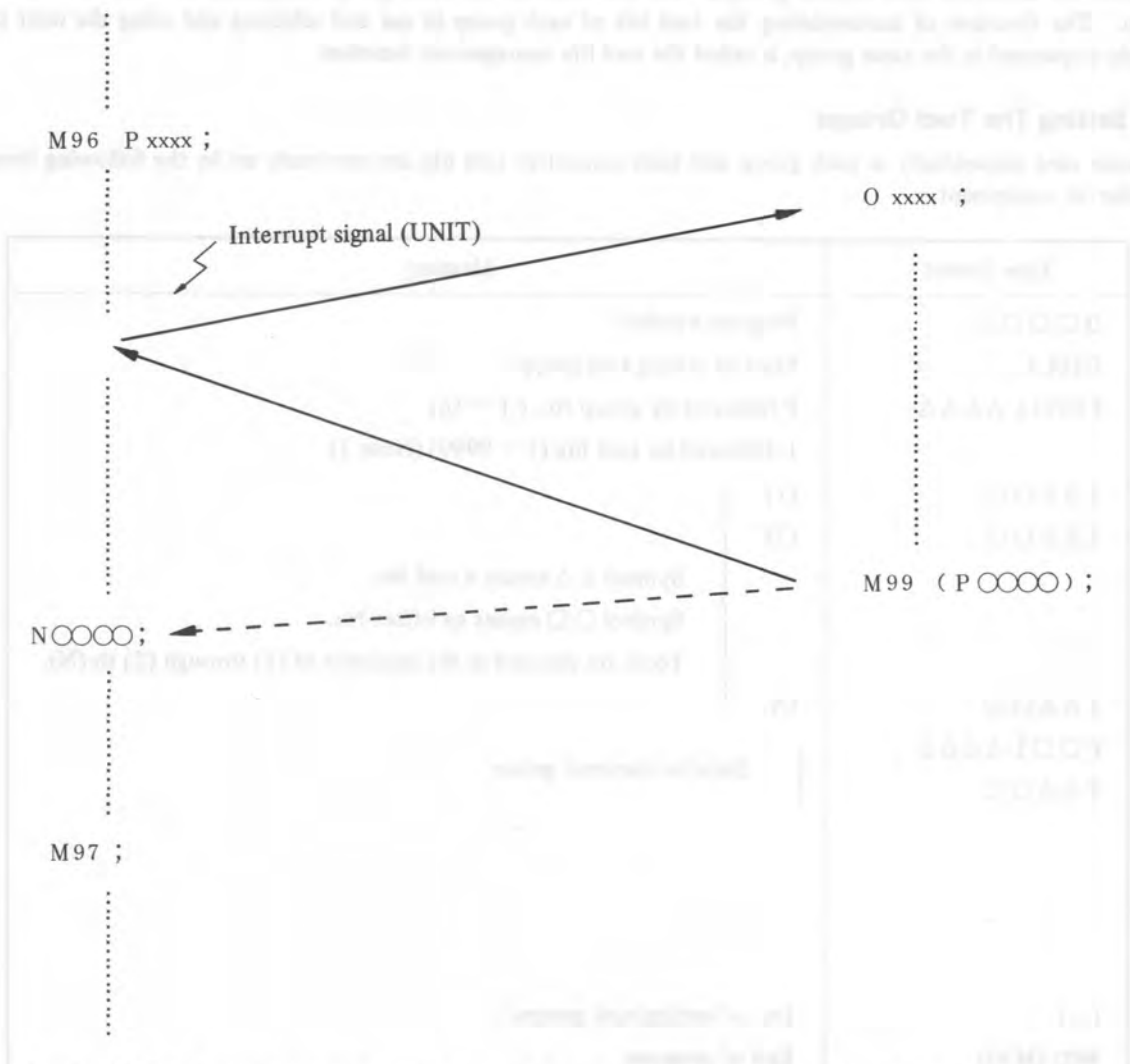
0 : Spaced 1 : No output

10.12.5 Cautions

- (1) It is not necessary to sequentially specify open command (POPEN), data output commands (BPRNT, DPRNT) and close command (PCLOS). When the open command is specified at the start of a program, it is no longer necessary to specify the open command again until the close command is specified next.
- (2) Specify the open command and close command as a pair of commands without fail. In other words, send the close command at the end of a program. Don't specify the close command independently without open command.
- (3) A data output command in progress is stopped and subsequent data are erased by reset processing. Accordingly, if reset processing is specified by M30 or the like at the end of a data output program, specify the close command at the end of the program, and wait for a while until all data are output before starting M30 processing or other reset processing.
- (4) This function requires custom macro B and I/O interface options.

10.13 Custom Macro Interrupt Function (Custom Macro B)

If an interrupt signal is input to NC halfway during the execution of a block between M96Pxxxx; and M97; blocks, processing is branched to Pxxxx program,



Specify M99; for returning from custom macro interrupt to the original program. The sequence number in the program to be returned can also be specified by address P.

- (Note 1) For details of custom macro function, see appendix 12.
- (Note 2) For using this function, refer to the operator's manual published by the machine tool builder, without fail.

11. TOOL LIFE MANAGEMENT

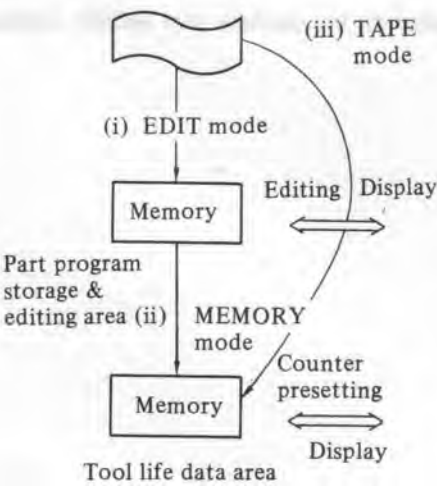
11.1 General

Tools are classified into various groups, with the tool life for each group (frequency or time of use) being specified. The function of accumulating the tool life of each group in use and selecting and using the next tool previously sequenced in the same group, is called the tool life management function.

11.2 Setting The Tool Groups

Tools used sequentially in each group and their respective tool life are previously set by the following format tape in the NC equipment.

Tape format	Meaning
O □ □ □ □	Program number
G10L3 ;	Start of setting tool group
P □ □ L △ △ △ △	P followed by group No. (1 ~ 16) L followed by tool life (1 ~ 9999) (Note 3)
T △ △ □ □ ;	(1) ⎵
T △ △ □ □ ;	(2) ⎵
⋮	⎵
⋮	⎵
⋮	⎵
T △ △ □ □ ;	(N) ⎵
P □ □ L △ △ △ △ ;	⎵
T △ △ □ □	⎵
⋮	⎵
⋮	⎵
⋮	⎵
G11 ;	End of setting tool groups
M02 (M30) ;	End of program



The setting operation is as follows.

- (i) Insert the above tape into the tape reader and press the **READ** button in the EDIT mode, as with an ordinary NC tape.
The program will be registered in the part program memory, and made ready for display and editing.
- (ii) Perform a cycle start operation in the MEMORY mode to run the program. The data will be stored in the tool life data area of the memory, and at the same time, the already existing tool life data of all groups will be canceled and the life counters also cleared.
Data stored once is not erased by turning the power off.
- (iii) Executing a cycle start operation in the TAPE mode instead of the operation of (i), stores the program contents directly from the tape onto the tool life data area. In this case, however, display and editing cannot be done as in (i).

The following is a concrete example of the tape format.

O0001 ;	
G10L3 ;	
P01L0150 ;	} Data of group 1
T1101 ;	
T0203 ;	
T0504 ;	
P02L0100 ;	} Data of group 2
T1205 ;	
T1307 ;	
T0915 ;	
T0712 ;	
P03L0050 ;	} Data of group 3
T0808 ;	
T1514 ;	
G11 ;	
M02 ;	

- (Note 1) Group Nos. specified by P need not be consecutive. All registerable groups need not be set, either.
- (Note 2) The same tool No. can appear at any frequency and at any place in the set data.
- (Note 3) Whether tool life is to be indicated by frequency or by time (minutes), is set by the parameter (309-LCTM).
- (Note 4) The number of groups and the number of tools per group that can be registered are 16 and 16, respectively. Accordingly, maximum 256 tools can be registered.
- (Note 5) If two or more offset numbers are used for the same tool in one process of machining, prepare the following format tape.

Tape format	Meaning
P04L0500 ; T0101 ; T0105 ; T0203 ; T0206 ; T0202 ; T0304 ; T0309 ; P05L1200 ; T0405 ; 	 } (1) Tools of group 4 are used 500 times (or minutes) in the order of (1) to (2) to (3), respectively. } (2) If this group is specified 3 times in one process of machining, offset numbers are selected in the following sequence. } (3) 01 to 05 to 05 for tool 01. 03 to 06 to 02 for tool 02. 04 to 09 to 09 for tool 03.

11.3 Specifying Tool Groups in a Machining Program

In a machining program, tool groups are specified by using the T codes as follows.

Tape format	Meaning
<pre> . . T△△99 ; —— Group No. . . T△△88 </pre>	Terminates the tool specified up to this time and begins to use the tools of group △△ . Number 99 is used to discriminate from usual designation. Cancels the tools of group △△ . Number 88 is used to discriminate from usual designation.

The following is a concrete example of the tape format.

Tape format	Meaning
<pre> ----- T0199 ; ----- T0188 ; ----- T0508 ; ----- T0500 ; ----- T0299 ; ----- T0199 ; ----- ----- ----- </pre>	Terminates the preceding tool and begins to use the tools of group 01. Cancels offsets of the tools of group 01. Terminates the tools of group 01 and selects tool No. 05 and offset No. 08. Cancels offsets of tool No. 05. Terminates the tool of No. 05 and begins to use tools of group 02. Terminates the tools of group 02 and begins to use the tools of group 01. If the tool is specified with two or more offset numbers, the second offset number is selected. If not, the same offset number as previous one is selected.

11.4 Execution of Tool Life Management

11.4.1 Counting tool life

(1) When tool life is specified by frequency

The counter for groups of tools that were used is increased by one, every process from the time that a cycle start operation is performed until the time that M02 or M30 is commanded and the NC is reset. Even if the command for the same group is given a number of times in one process, the counter increase stays at one. The maximum life value is 9999.

(2) When tool life is specified by time (minutes)

In this case, T△△99 is specified, then another T code which specifies the use of another tool is specified in a machining program. This time, during which tools are actually used in the cutting mode, is counted at certain intervals (four seconds). The time for single block stop, feed hold, rapid traverse, dwell, etc., is disregarded. The maximum settable life value is 4300 minutes.

In either case, life is counted for each group, and the contents of the counter are not erased by turning off the power.

(Note) When having executed M02 or M30 with life specified by the frequency of use, input the External Reset (ERS) or Reset & Rewind (RRW) signal to the NC.

11.4.2 Tool change signal and tool change reset signal

Tools are selected one after another in a predetermined sequence upon the end of each life. When the last tool has come to its life end in a tool group, the Tool Change signal is output. Changing the tools at life end, which are displayed on the CRT screen, then specifying the group No. concerned and inputting the Tool Change Reset signal or operating the MDI panel (see 11.5.3), clears all data of the group, such as the life counter, *, @, etc. (see 11.5.2). Effecting tool change reset for all groups at life end releases the Tool Change signal automatically. Specifying this group after resumption of machining begins selection again with the first tool.

(Note) With life specified by frequency, the Tool Change signal is output when M02 or M30 resets the NC after life end has been reached. With life specified by time, the Tool Change signal is output even during machining when the life end is reached. However, machining continues until the end of the program.

11.4.3 Tool skip signal

A tool before its life end can forcibly be changed in one of the following ways.

- Specifying the group No. concerned and inputting the Tool Skip signal selects the next tool in the group with the next T code command.
- Specifying no group No. and inputting the Tool Skip signal assumes that the tool being selected has been specified. The other is the same as in (i).

Whether (i) or (ii) is to be taken is set by parameter. In either way, life is counted from 0. When, however, the Tool Skip signal is input to the last tool, the Tool Change signal is output.

11.4.4 New tool select signal

When a new tool is about to be selected in a group, the New Tool Select signal is output simultaneously with the output of the T code of the tool. When a new tool is selected, this signal can be utilized for automatic measurement of compensation amounts of the tool, etc.

(Note) Input neither Tool Change Reset signal nor the Tool Skip signal when either STL or SPL lamp is on or when both STL and SPL lamps are on.

11.5 Display and Input of Tool Data

11.5.1 Display and modification of tool group data

Tool group data in the part program storage & editing area can be displayed and modified as with ordinary program editing. Be sure to execute the modified program, as was discussed in Section 11.2. Otherwise, it will not be stored in the tool life data area.

11.5.2 Display of tool life data during program execution

Pressing the **DGNOS** button twice in any mode displays the first page of the tool life data in the following format on the CRT screen.

Group No.		Life		
TOOL LIFE DATA		01 :	00001 N0012	
		SELECTED GRP	01	No. of group being commanded
GRP	01:	LIFE	0120	COUNT 0025
				Life counter
		*0124	*1428	*0417 *0906
		*1201	@0325	1030 1631
		2322	0702	1913 2116
		0609	2512	1305 2732
				Tool Nos. of group 1
				*: tool at life end
				@: tool being used
				#: tool skipped
GRP	02:	LIFE	0050	COUNT 0012
		*1503	*0514	*2619 *1727
		*0210	*1104	*0823 *2407
		#1811	2215	2226 2920
		2008	2818	3021 3129
				Tool Nos. of group 2
GRP TO BE CHANGED:				
		003 006 012 013 014	→	
N		LSK INC		Group Nos. of tools to be changed

PAGE

To change a life counter, select the MDI mode then:

The counter of the group at the current cursor position is preset to . Other data on this group remains unchanged.

	3	0	9	TLSK	GRST		LCTM				
--	---	---	---	------	------	--	------	--	--	--	--

Alarm No.	
140	The group No. exceeds the maximum value (16).
141	The tool group specified in the machining program is not set.
142	The number of tools in the group exceeds 16.
143	No T code is stored in the block of the program to set tool groups.
145	Group No. $\Delta \Delta$ specified by T $\Delta \Delta$ 88 does not coincide with the group No. of the tool currently being used in the machining program.
146	P and L were not specified at the beginning of the program to set the tool group.
147	The number of the tool groups exceeds the maximum value (16).

12. PROGRAM TAPE COMPOSITION

The program written on a process sheet is punched on paper tape. Various devices for punching paper tape are on the market. Any of them will do so long as the paper tape is punched as shown in Section 12.1.

The SYSTEM P-G is sold as a paper tape punch device.

(Note) The SYSTEM P-G also edits programs using a graphic display. It can also be used as an automatic programming system.



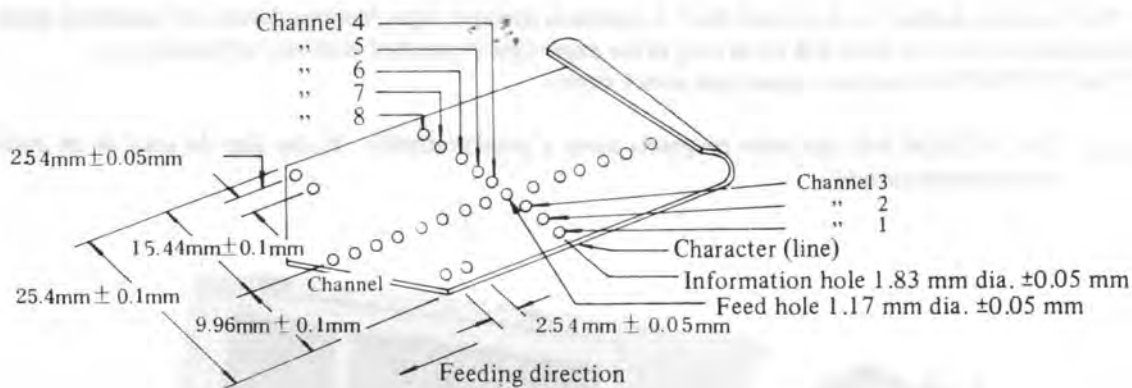
12.1 Paper Tape Used for NC Control

A program containing NC machine tool commands is ordinarily punched on paper tape and read into the NC system through a Tape Reader. An 8-unit paper tape 1 inch in width as normally used for NC control, and dimensions of the tape, such as width, thickness, hole-position, hole-diameter and so on, are determined by various standards including EIA, ISO or JIS. The following table shows the specification of paper tape that can be used in this control.

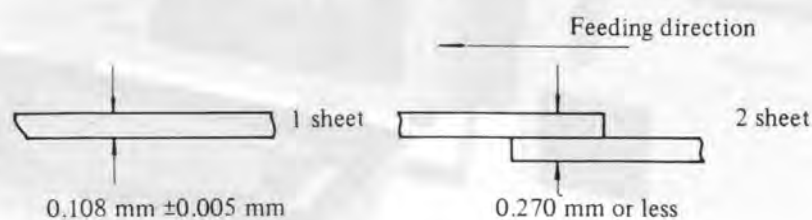
Table 12.1 Specification for Paper Tape Used by NC

1	Kind of tape		8-channel paper tape (Mylar tape or so on cannot be used)	Same as left
2	Percentage of light transmission $= \left(\frac{\text{Transmission light}}{\text{Incident light}} \times 100\% \right)$		40% or less	0.1% or less
3	Colour of tape		Any colour can be used such as black, grey, blue, pink, white, etc., if percentage of light transmission is 40% or less.	Black only
4	Standard	Associated with material of paper tape	JIS C 6243, EIA RS-227-A or ISO 1729 (Provided that percentage of light transmission conforms to Item (2).)	Same as left
		Associated with dimensions and locations of holes in paper tape	JIS C 6246, EIA RS-227-A or ISO 1154	Same as left

Example of EIA standard RS-227 is shown below:



(a) Dimensions of the punched tape



(b) Thickness of tape

Standard dimensions of punched tape (EIA, RS-227)

12.2 Tape Code

A punched line on paper tape represents a numeral, character or sign, and the line is called a character. A character is composed of 8-hole positions which are called the code. The following two tape codes can be used in the SYSTEM 6T: However, ISO code is an option.

- (i) EIA code EIA RS-244-A
- (ii) ISO code ISO 840

(1) EIA/ISO determination

The code EIA or ISO, used on a punched tape is determined automatically. When reading a paper tape in Label Skip mode (Note 1), if the first End of Block code is read as CR in EIA code, subsequent codes are read written in EIA code; while, if the first End of Block code is read as LF in ISO code, subsequent codes are read as ISO code. EIA and ISO codes cannot be mixed on a paper tape.

(Note 1) After power-on or reset, all codes read before the first EOB code are ignored. This status is called Label Skip mode.

(2) Parity check (TH check)

An alarm (TH alarm) will be generated, if the following conditions are not satisfied by the number of code-holes read in a significant information section (a block following the first CR or LF, excluding the comment section):

- (i) EIA code An odd numbered code-holes. (All punches are excluded.)
- (ii) ISO code An even numbered code-holes.

When the TH alarm is sensed, reading of the tape is stopped immediately (Note 2) and alarm No. 001 is displayed on the CRT screen. This keeps the NC machine tool from machining erroneously due to incorrect punches or a tape-read error.

(Note 2) The tape reader does not stop immediately when a TH alarm is generated. Because of the read-in buffer, the tape reader stops after reading one block or when the read-in buffer is filled. The position and the character where the TH alarm occurred are displayed in DIAGNOSE 710 and 711.

A parity check is performed in the vertical direction of a block, and an alarm (TV alarm) is generated if the block contains an odd number of characters (from the character following an EOB to the next EOB).

When the TV alarm is sensed, reading of the tape is stopped immediately, and alarm No. 002 is displayed on the CRT screen. This keeps the NC machine tool from machining erroneously due to incorrect punches or a tape-read error.

This check function can be specified as Valid/Invalid by the SETTING operation. When using this function, punch an ignored code such as the space code so that the number of characters is even.

12.3 Codes Used in Programming

Codes shown in the table below can be used for programming in EIA/ISO code. Codes not included in this table will be ignored when read from a paper tape, but invalid codes will be counted in the parity check which can result in a TH alarm.

Table 12.3 Codes used for programming

ISO code									EIA code									Meaning		
Character	8	7	6	5	4	3	2	1	Character	8	7	6	5	4	3	2	1			
0			○	○					0			○						Numeral 0		
1	○		○	○				○	1								○	Numeral 1		
2	○		○	○			○		2							○		Numeral 2		
3			○	○				○	3				○				○	Numeral 3		
4	○		○	○			○		4							○		Numeral 4		
5			○	○			○		5				○			○		Numeral 5		
6			○	○			○	○	6				○			○	○	Numeral 6		
7	○		○	○			○	○	7							○	○	Numeral 7		
8	○		○	○	○				8						○			Numeral 8		
9			○	○	○			○	9					○	○			Numeral 9		
A		○							a		○	○					○	Address A		
B		○						○	b		○	○				○	?	Address B		
C	○	○						○	c		○	○	○				○	?	Address C	
D		○						○	d		○	○				○		Address D		
E	○	○						○	e		○	○	○			○		Address E		
F	○	○						○	f		○	○				○	○	Address F		
G		○						○	g		○	○				○	○	Address G		
H		○							h		○	○		○				?	Address H	
I	○	○						○	i		○	○	○	○				○	Address I	
J	○	○						○	j		○		○					○	?	Address J
K		○						○	k		○		○				○		Address K	
L	○	○						○	l		○						○	○	Address L	
M		○						○	m		○		○				○		Address M	
N		○						○	n		○						○		Address N	
O	○	○						○	o		○						○	○	Address O	
P		○		○					p		○		○				○	○	Address P	
Q	○	○		○				○	q		○		○	○					Address Q	
R	○	○		○				○	r		○			○				○	Address R	
S		○		○				○	s			○	○				○		Address S	
T	○	○		○				○	t			○					○	○	Address T	
U		○		○				○	u			○	○				○		Address U	
V		○		○				○	v			○					○	?	Address V	
W	○	○		○				○	w			○					○	○	Address W	
X	○	○		○					x			○	○				○	○	Address X	
Y		○		○				○	y			○	○	○					?	Address Y
Z		○		○				○	z			○	○	○				○	Address Z	
DEL	○	○	○	○	○	○	○	○	Del		○	○	○	○	○	○	○	*	Delete (cancel an error punch).	
NUL									Blank									*	Not punched. Can not be used in significant section in EIA code.	
BS	○								BS			○		○			○	*	Back space	
HT								○	Tab			○	○	○			○	*	Tabulator	
LF or NL								○	CR or EOB	○									End of block	
CR	○							○										*	Carriage return	
SP	○		○						SP				○					*	Space	
%	○		○					○	ER					○			○	○	Absolute rewind stop	
(			○		○				(2-4-5)				○	○			○		Control out (a comment is started)	
)	○		○		○			○	(2-4-7)					○			○		Control in (the end of a comment)	
+			○		○			○	+			○	○	○				*	Positive sign	
-			○		○			○	-			○							Negative sign	
:			○	○	○			○											Colon	
/	○		○		○			○	/				○	○				○	Optional block skip	
.			○		○			○	.			○	○				○	○	Period (A decimal point)	
#	○		○					○											Sharpe	
\$			○					○											Dollar sign	
&	○		○					○	&						○		○	○	Ampersand	
'			○					○											Apostrophe	
*	○		○		○			○											Asterisk	
,	○		○		○			○	,				○	○	○			○	Comma	
;	○		○		○			○	;										Semicolon	
<			○		○			○											Left angle bracket	
=	○		○		○			○	=										Equal	
>	○		○		○			○	>										Right angle bracket	
?			○		○			○	?										Question mark	
@	○	○							@										Commercial at mark	
”			○					○	”										Quotation	
[	○	○			○			○	[										Left brace	
]	○	○			○			○	]										Right brace	

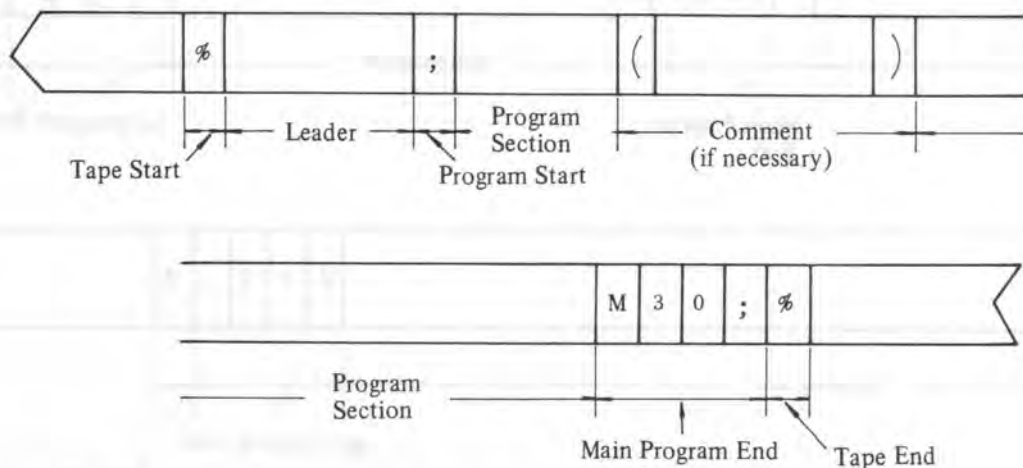
- (Note 1) The codes with an asterisk are read into tape memory only when specified in the comment section. They are ignored in significant information sections.
- (Note 2) The codes with a question mark are read in tape memory only when specified in a comment. They generate an alarm if used in significant information sections.
- (Note 3) When a user macro option is used, the following codes can also be used in significant information sections.
 [,], #, *, =, +, B, C, H, J, V, and Y in ISO
 [,], &, +, a code set by parameter, B, C, H, J, V, and Y in EIA
- (Note 4) Codes not included in this table and with correct parity are always ignored.
- (Note 5) A code that does not have the correct parity generates the TH alarm but ignored in a comment section.
- (Note 6) A character with all eight holes punched does not generate TH alarm even in EIA code.

12.4 Program Tape

A program punched on a paper tape is composed of the following sections:

- (i) Tape Start
- (ii) Leader
- (iii) Program Start
- (iv) Program Section
- (v) Comment
- (vi) Program End
- (vii) Tape End

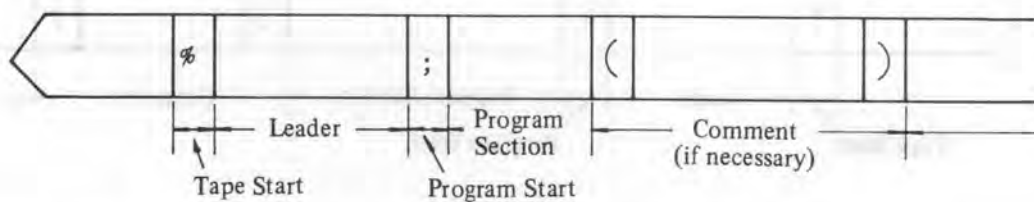
(1) A main program on a tape

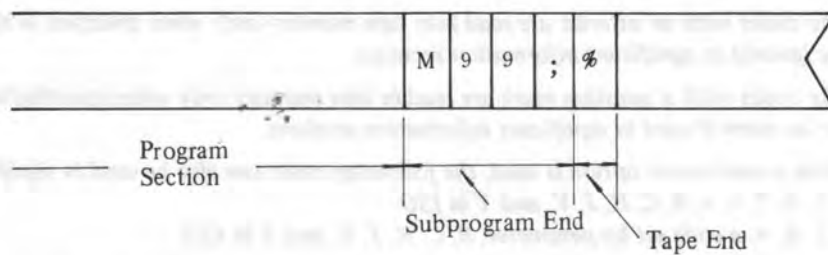


(Note 1) M02 can be used instead of M30 at the end of the main program.

(Note 2) When in ISO code, LF is used for ;, while in EIA code, CR is used.

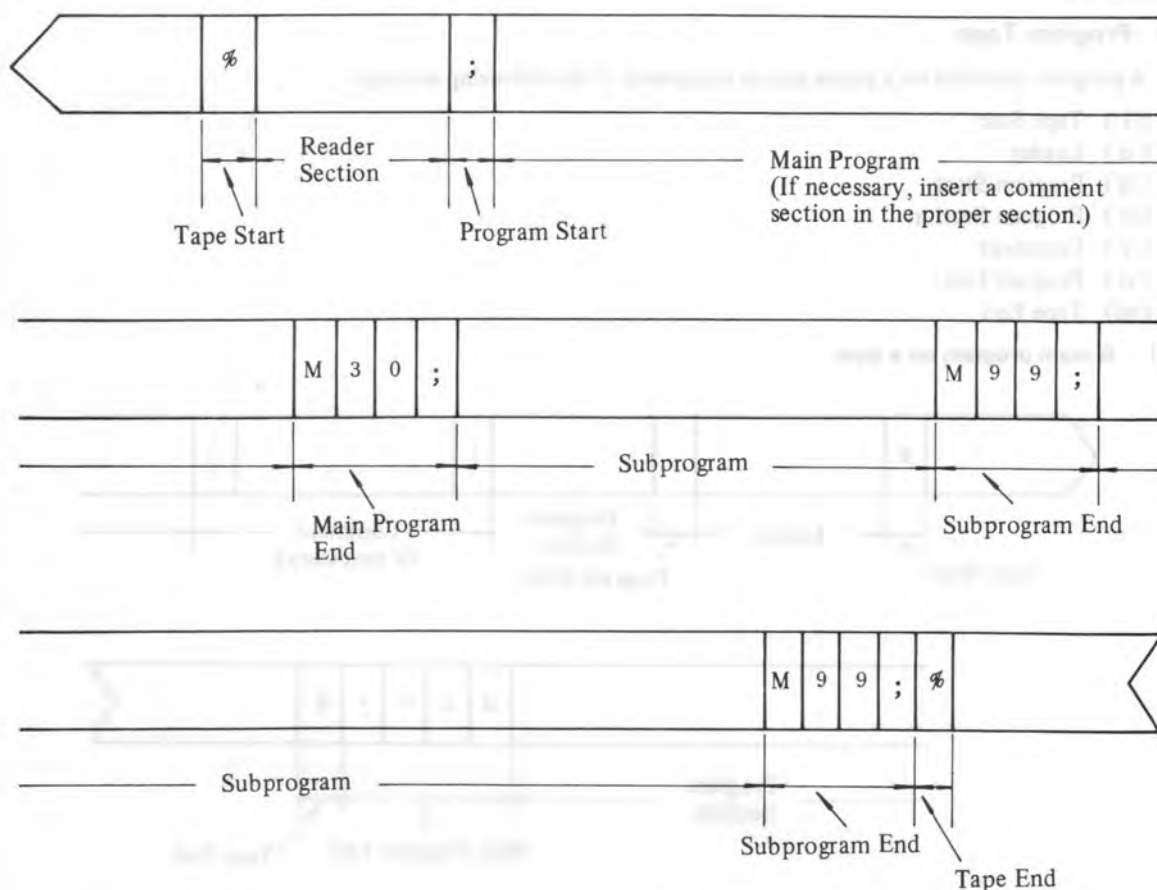
(2) One subprogram on a tape





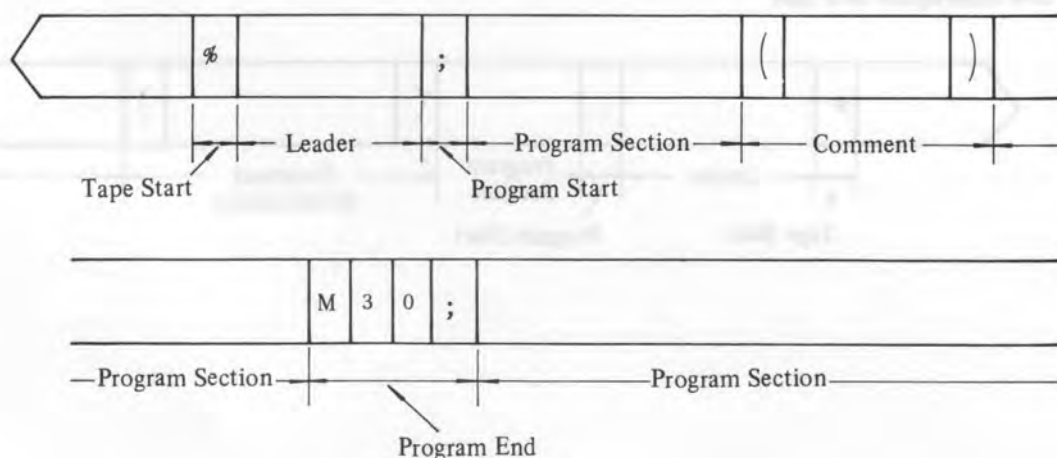
A subprogram must be used after being entered into the tape memory area.

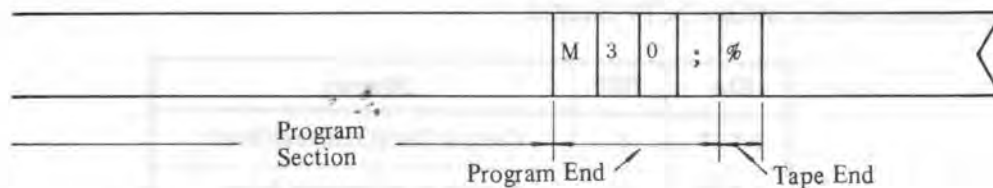
(3) One main program and a number of subprograms on a tape



A program including subprograms on a tape must be used after being entered into the tape memory area.

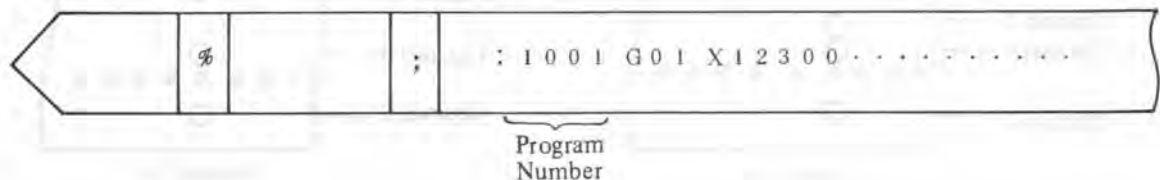
(4) A number of programs on a tape





A program number must be put at the first block of each main program section or subprogram section. A program number is a four-digit number following address O. (When in ISO, : can be also used instead of O)

Program numbers are not necessarily given. See Section 2.6 for details.



12.4.1 Tape start

The Tape Start must be punched at the top of the paper tape by using the following code. This section is necessary to stop rewinding the tape when used with a Tape Reader with Reels. This section may be omitted when the tape is used with a Tape Reader without Reels.

EIA	ISO	Meaning
ER	%	Tape Start (Rewind Stop)

12.4.2 Leader section

Information punched before the first LF (for ISO) or CR (for EIA) on a paper tape is called the Leader Section. Normally, the NC unit is set to the Label Skip condition when turning the power on or resetting and this section of paper tape is loaded on a Tape Reader. The Tape Reader reads but skips this section (by the Label Skip function). Generally, a label for a tape is punched in this section. Since no parity check is performed on the skipped leader section, any combination of codes other than LF and CR may be punched in this section.

The label skip function is used to skip all information encountered from the time the power is turned on or a resetting is performed until reading the first EOB code. (There is no parameter used to delete label skip function.)

12.4.3 Program start

Punching the following code immediately after the end of the Leader (at the top of the first block of a program) indicates the start of a program. This code is necessary to release the Label skip function.

EIA	ISO	Meaning
CR	LF	Program Start

12.4.4 Program section

The part of the paper tape punched between the Program Start and Program End code (except the Comment section explained in Section 12.4.5) is called the Program Section. Information specifying tool travel and machine ON/OFF functions is punched in this section. This section is called the significant section and is different from the non-significant section which includes the Leader and Comment Section. Details on the Program Section are given in Section 12.5.

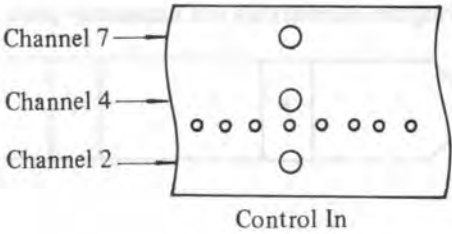
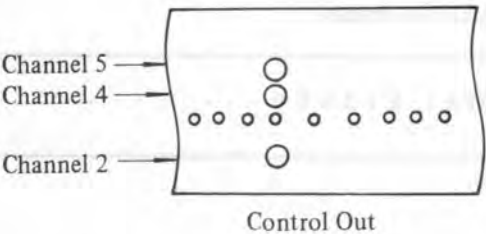
12.4.5 Comment section

Information punched in the section between Control Out and Control In codes is read but skipped and is regarded as a comment. The TH check is not performed on this section, therefore, the leader or comments can be punched by using any combination of punch-holes desired.

The comment section will also be TV checked.

EIA	ISO	Meaning
2-4-5*	(	Control Out (Comment Start)
2-4-7*	)	Control In (Comment End)

*: Hole combinations in EIA code.



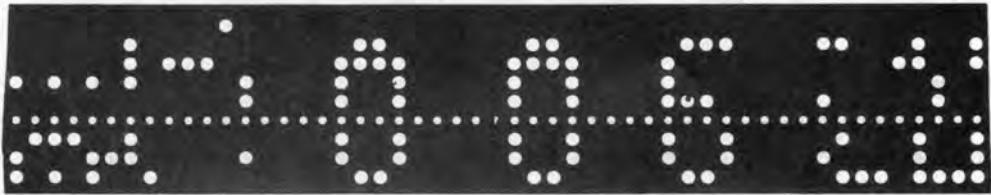
(Example) In ISO code

```
..... LF
N1000G00X ..... LF
(MEASURE WORK ..... ) LF
N1001G01X ..... LF
..... LF
```

When a command tape is read into memory, for memory-controlled operation, the comment section is read into memory. However codes not listed in the Table 12.3 are ignored and are not read into memory. When data in the memory is punched out, information in the Comment Section is also punched out. The Comment Section is displayed on the screen of the CRT. However codes ignored when read-in are neither punched nor displayed on the CRT screen.

During a memory-controlled operation in memory command mode, the comment section in memory is ignored and operation is executed.

(Example) Used as heading



F1000CR(

)N110

Significant section

Name of the tape, etc.

Significant section

(Note 1) The rewind stop code (% or ER) cannot be used in the comment section. Reading this code resets the NC.

(Note 2) When a comment section is in the middle of a program section and it is long, movement is interrupted for a long time. Therefore, insert comment sections in a place where movement can be interrupted or a place specifying no movement.

(Note 3) The length of the comment section is unlimited.

(Note 4) When the control out code does not come first and only the control in code is read in, it is ignored.

(Note 5) The TV check for the comment section can be made disabled by specifying parameter TVC.

12.4.6 Program end

The end of a program is indicated by punching the following code at the end of the Program Section.

EIA	ISO	Meaning
M02 CR	M02 LF	Program End
M30 CR	M30 LF	Program End and Rewind
M99 CR	M99 LF	Subprogram End

When the End of Program is sensed during the execution a program, the NC unit terminates the execution of the program and enters a cleared status. In case of M30CR or M30LF, the tape is rewound (in Tape Command mode) or the program is returned to its beginning (in Memory Command mode). When the code is sensed in a subprogram, control is returned to the program which called the subprogram. (Whether or not the tape is rewound by execution of M30 depends on the machine tool.)

12.4.7 Tape end

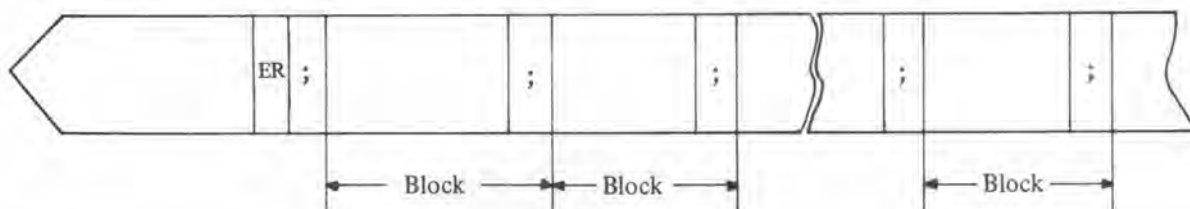
The end of a tape is indicated by punching the code below immediately following the Program End of the last program on a tape.

EIA	ISO	Meaning
ER	%	Tape End

(Note) If M02 and M30 are not at the end of the program section, and ER (EIA) or % (ISO) is about to be executed, the NC enters a reset status.

12.5 Block Composition

A program section is composed of a number of blocks. A block is partitioned into a number of words, and the end of a block is identified by an End of Block Code (LF in ISO code and CR in EIA code).



(Note 1) Maximum number of characters allowed in a block.

During actual cycle operation, a program is read into the tape input buffer, block by block. The size of the input buffer is limited to 64 characters, so if the length of a block exceeds this limit, 64 characters are inserted into the input buffer. After processing the 64 characters, the others are read into the buffer. (Of course this is done while the previous block is being executed.) Ignored codes are also read into the input buffer once.

(Note 2) TV check (vertical parity check)

A parity check is performed in the vertical direction of a block, and alarm No. 002 is generated if the block is found to contain an odd number of characters (between a code immediately following an EOB and the next EOB). The TV check is not performed for a part of tape which is read but skipped by the Label Skip function, but the Command Section between "(" and ")" is checked for the total number of characters in each block. The TV check function can be specified Valid/Invalid by using the MDI unit. (See Section 5.7 in Chapter IV.)

(Note 3) In program examples in this manual, ; is used instead of the End of Block code (LF for ISO and CR for EIA), but the proper EOB code LF (ISO) or CR (EIA) must be used in actual programming.

The end of a program is indicated by punching the following code at the end of the Program Section.

EIA	ISO	Meaning
M02 CR	M02 LF	Program End
M30 CR	M30 LF	Program End and Rewind
M99 CR	M99 LF	Subprogram End

When the End of Program is sensed during the execution of a program, the NC unit terminates the execution of the program and enters a cleared state. In case of M30 CR or M30 LF, the tape is rewound (in Tape Command mode) or the program is returned to its beginning (in Memory Command mode). When the code is sensed in a subprogram, control is returned to the program which called the subprogram. (Whether or not the tape is rewound by execution of M30 depends on the machine tool.)

12.4.7 Tape end

The end of a tape is indicated by punching the code below immediately following the Program End of the last program on a tape.

EIA	ISO	Meaning
EH	#	Tape End

(Note) If M02 and M30 are not at the end of the program section, and EH (EIA) or # (ISO) is about to be executed, the NC enters a wait state.

12.5 Block Composition

A program section is composed of a number of blocks. A block is partitioned into a number of words, and the end of a block is identified by an End of Block Code (LF in ISO code and CR in EIA code).



(Note 1) Maximum number of characters allowed in a block.
During normal operation, a program is read into the tape input buffer block by block. The size of the input buffer is limited to 84 characters, so if the length of a block exceeds this limit, 84 characters are inserted into the input buffer. After processing the 84 characters, the others are read into the buffer. (Of course this is done while the previous block is being executed.) Ignored codes are also read into the input buffer once.

(Note 2) TV check (vertical parity check).
A parity check is performed in the vertical direction of a block, and when No. 003 is generated if the block is found to contain an odd number of characters (between a code immediately following an EOB and the next EOB). The TV check is not performed for a part of tape which is read but skipped by the Label Skip function, but the Command Section between "V" and "I" is checked for the total number of characters in each block. The TV check function can be specified (valid/invalid) by using the MDI unit. (See Section 2.3 in Chapter IV.)

(Note 3) In program examples in this manual, LF is used instead of the End of Block code (LF in ISO and CR in EIA), but the proper EOB code (LF (ISO) or CR (EIA)) must be used in actual programming.

IV. OPERATION

IV. OPERATION

1. TURNING THE POWER ON AND OFF

1.1 Turning the Power On

- (1) Make sure that the front and back doors of the equipment and the door of the tape reader are closed.
- (2) Turn on the power according to the manual issued by the machine tool builder.
- (3) Make sure that something is displayed on the CRT. Something is displayed in several seconds after the power is turned on.
- (4) Make sure that the fan motor is on.

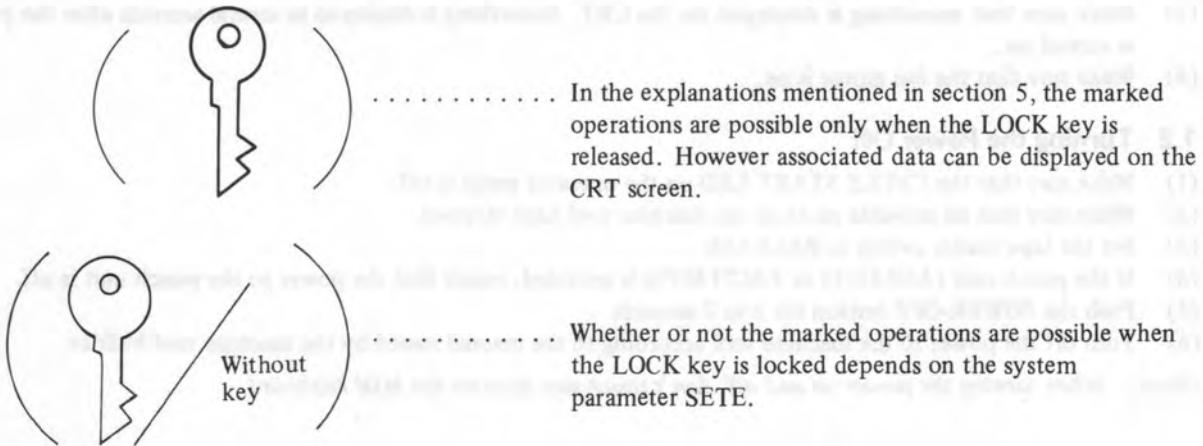
1.2 Turning the Power Off

- (1) Make sure that the CYCLE START LED on the operator panel is off.
- (2) Make sure that all movable parts on the machine tool have stopped.
- (3) Set the tape reader switch to RELEASE.
- (4) If the punch unit (ASR33/43 or FACIT4070) is provided, insure that the power to the punch unit is off.
- (5) Push the POWER-OFF button for 1 to 2 seconds.
- (6) Turn off the power to the machine tool according to the manual issued by the machine tool builder.

(Note) When turning the power on and off, don't touch any keys on the MDI keyboard.

2. LOCK KEY

A LOCK key can be provided on the operator's panel on the machine tool.
If the LOCK key is not released, some operations cannot be performed.

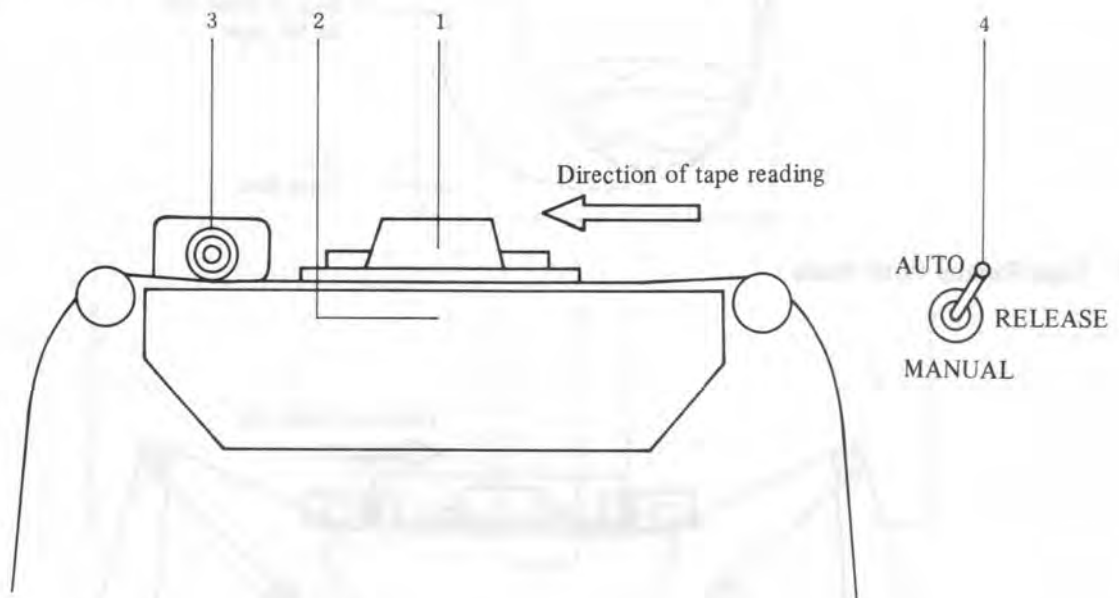


3. TAPE READER

The characteristics of the Tape Reader are as follows:

		Tape Reader without Reels	Tape Reader with Reels
Read Speed		250 characters/sec (50 Hz) 300 characters/sec (60 Hz)	
Rewind Speed			500 characters/sec (50 Hz) 600 characters/sec (60 Hz)
Reel Capacity			Diameter 187 mm 150 m (tape length)
Tape box capacity	Free-standing type	Approximately 30 m max.	
	Built-in type 1	Approximately 10 m max.	
	Built-in type 2	Approximately 10 m max.	

3.1 Tape Reader Without Reels



1. Light Sources

An LED (light emitting diode) is mounted for each channel and for the feed hole (9 diodes in total). A built-in Stop Shoe functions to decelerate the tape.

The light source is attracted to the optical reader by a magnet so that the tape will be held in the correct position. This unit can be opened upward, by turning the tape reader control switch to the RELEASE position (this turns off the magnet).

2. Optical Reader

Reads data punched on the tape, through a glass window. Dust or scratches on the glass window can result in reading errors. Keep this window clean.

3. Capstan Roller

Controls the feeding of tape as specified by the Control Unit.

4. Tape Reader Control Switch

A 3-position switch used to control the Tape Reader.

RELEASE The tape is allowed to be free, or used to open the light source. When loading or unloading the tape, this position is selected.

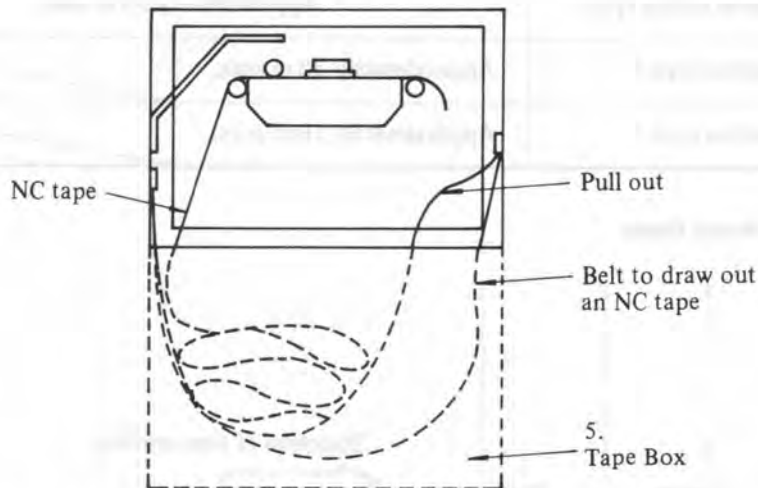
AUTO The tape is set to fixed position by the Stop Shoe. The feed and stop of the tape is controlled by commands from the Control Unit.

During a tape-controlled operation or data input from tape, the Light Source must be closed and this position must be selected.

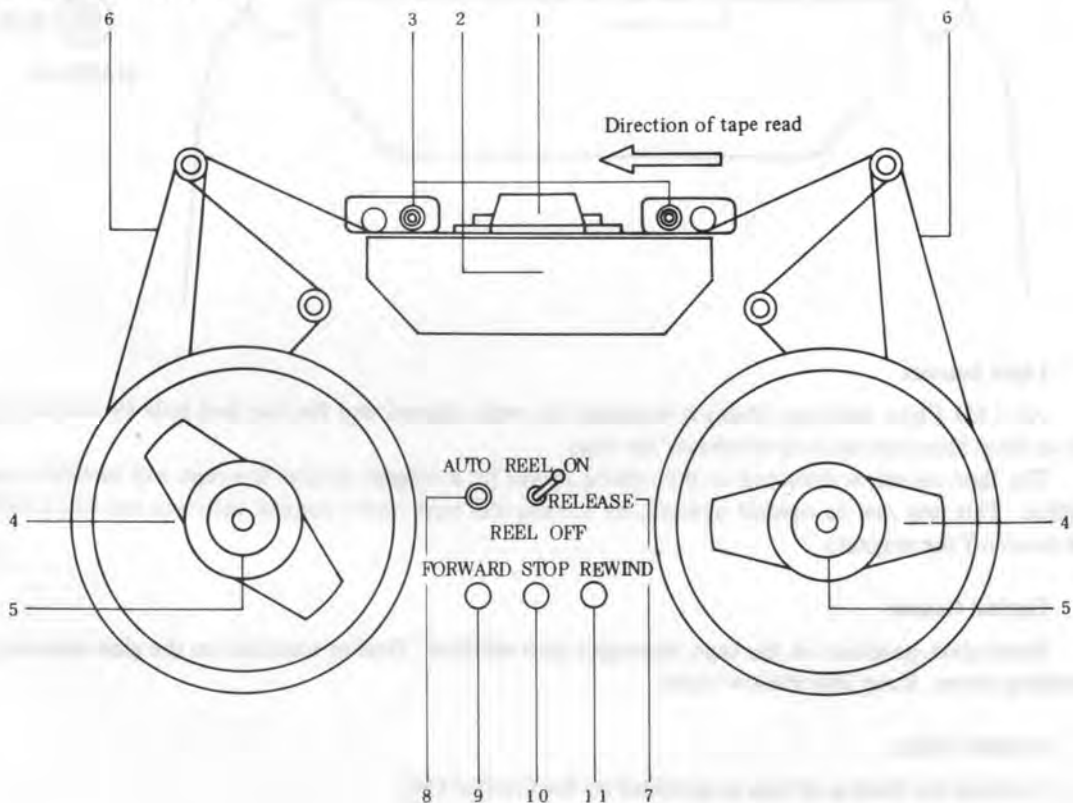
MANUAL The tape can be fed in the forward reading direction. If another position is selected, the tape feed is stopped.

5. Tape Box

A Tape Box is located below the Tape Reader. A belt used to draw out a paper tape is located inside the box. The paper tape can easily be pulled out using this belt.



3.2 Tape Reader With Reels



1. Light Source

An LED (light emitting diode) is mounted for each channel and for the feed hole (9 diodes in total). A built-in Stop Shoe functions to decelerate the tape.

The light source is attracted to the optical reader by a magnet so that tape will be held in the correct position adequately pressurized. The light source can be opened upward by turning the tape reader control switch to the RELEASE position (Turn off the magnet).

2. Optical Reader Unit

Reads data punched on the tape, through a glass window. Dust or scratches on the glass window can result in reading errors. Keep this window clean.

3. Capstan Rollers

Control the feeding of the tape as specified by the Control Unit. One capstan is for Forward Feed; the other, for Reverse Feed.

4. Reel

Used to feed out and take up the NC tape. A 150 m NC tape can be used.

5. Reel boss

A boss used to accept the reel. Provided with pawls to secure the reel. Driven by a reel motor, the boss turns the reel as specified by the Control Unit.

6. Tension Arm

Absorbs the internal delay of the reel motor response at the start/stop of tape motion. This arm also adjusts the feeding of the tape from the feed reel.

7. Tape Reader Control Switch

A 3-position switch used to change the operating mode of the Tape Reader.

RELEASE The tape is allowed to be free. The Light Source unit can be opened. This position must be selected when loading/unloading an NC tape.

REEL ON The tape is secured by the Stop Shoe, and the reel motor starts running. This position is selected to operate the Tape Reader with a reeled tape.

REEL OFF The tape is secured by the Stop Shoe, but the reel motor is not started. This position is selected to operate the Tape Reader without using reels.

When the Tape Reader is operated, or when inputting data from an NC tape, the Light Source must be closed and the switch must be set to the REEL ON or REEL OFF position.

8. AUTO Lamp

Lights while the Tape Reader is automatically operated from the Control Unit. While this lamp is lit, the manual control buttons on the Tape Reader (FORWARD, STOP and REWIND) cannot be used.

9. FORWARD Button

A push button used to feed tape in the reading direction (forward). When the AUTO lamp is OFF and the switch is at REEL ON or REEL OFF, the tape can be fed forward by pushing this button until it is stopped by one of the following conditions:

- (i) 10. STOP button is pushed.
- (ii) ER in EIA code or % in ISO code is read.
- (iii) RESET button on the MDI & DPL panel is pushed.

10. STOP Button

A button used to stop tape feed performed by pressing the FORWARD or REWIND button.

11. REWIND Button

A button used to feed the NC tape in the reverse direction. The NC tape is fed in the reverse direction when this button is pushed while the AUTO lamp is OFF and the switch is set to REEL ON or REEL OFF. Once

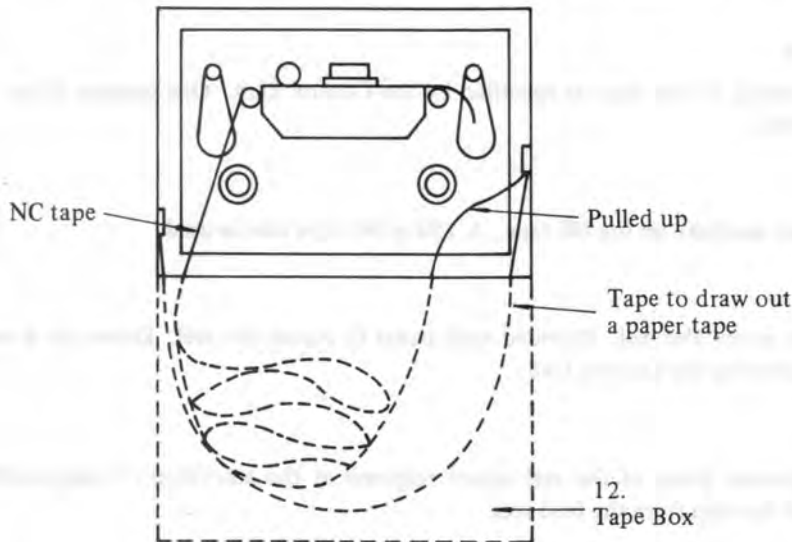
fed in the reverse direction, the tape is rewound until one of the following condition is sensed:

- (i) STOP button is pushed.
- (ii) ER in EIA code or % in ISO code is read.
- (iii) RESET button on MDI & DPL panel is pushed.

The NC tape cannot stop at the specified character but will overrun.

12. Tape Box

A Tape Box is located below the Tape Reader. A belt to draw out a paper tape in the box facilitates the removal of an NC tape.

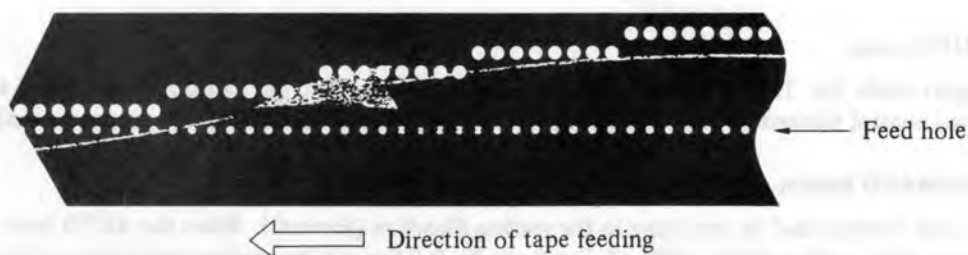


3.3 Tape Reader Handling

(1) Tape loading

(a) Tape Reader without Reels

- ① Open the Tape Reader Door.
- ② Set the Tape Reader switch to the RELEASE position.
- ③ Lift the Light Source Unit, and insert an NC tape between the gap. The tape must be positioned as shown in the figure, when viewed looking downward.



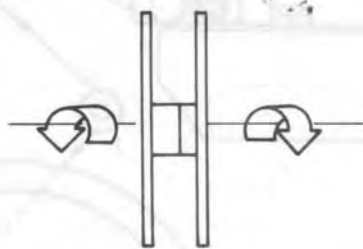
- ④ Pull the tape until the top of the tape goes past the Capstan roller.
- ⑤ Check that the NC tape is correctly positioned by the Tape Guide.
- ⑥ Lower the Light Source.
- ⑦ Turn the switch to the AUTO position.
- ⑧ Suspend the top and rear-end of the tape in the Tape Box.
- ⑨ Close the Tape Reader Door.

(b) Tape Reader with Reels (when reels are not used)

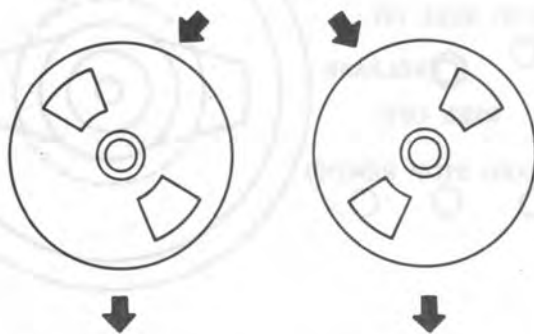
Identical to the steps in (a) except that the switch is set to the REEL OFF position, in step ⑦.

(c) Tape Reader with Reels (when reels are used)

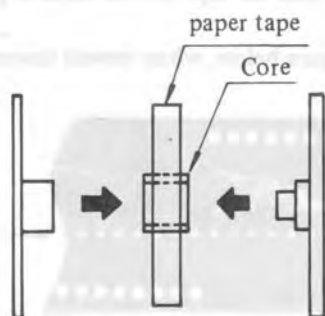
① Correctly reel the NC tape.



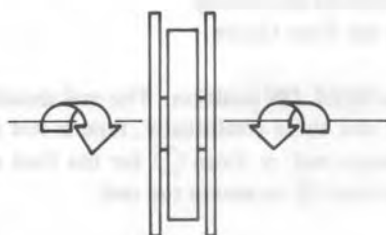
(i) The reel can be dismantled by twisting the reel hub in the direction of the arrow.



(ii) Separated reel.

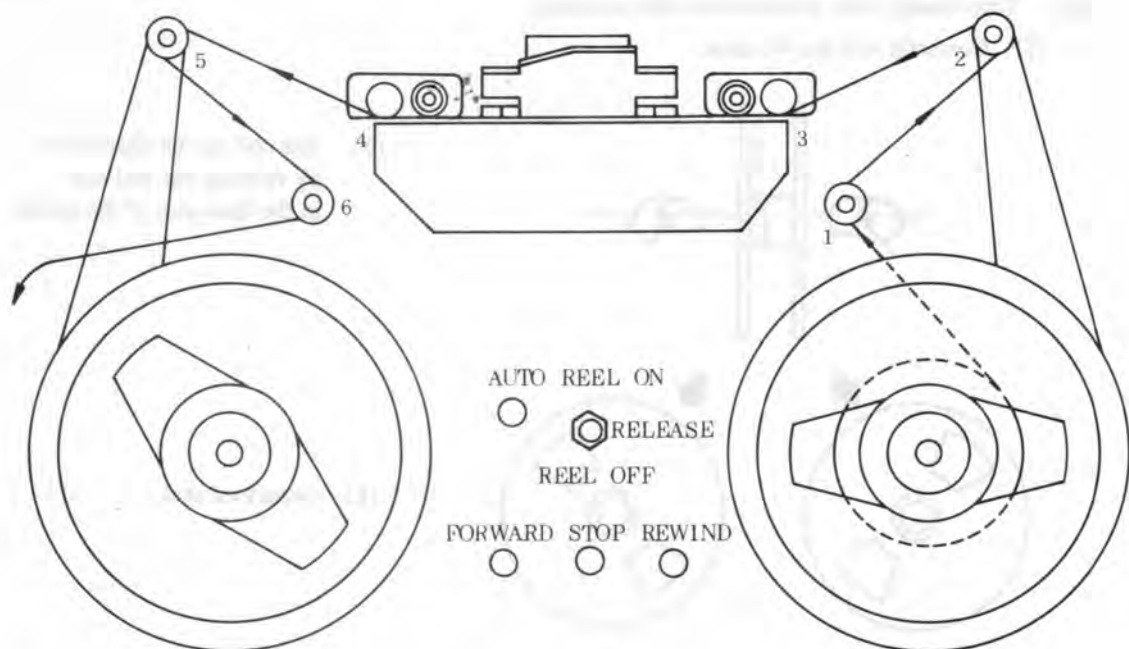


(iii) Load the NC tape.

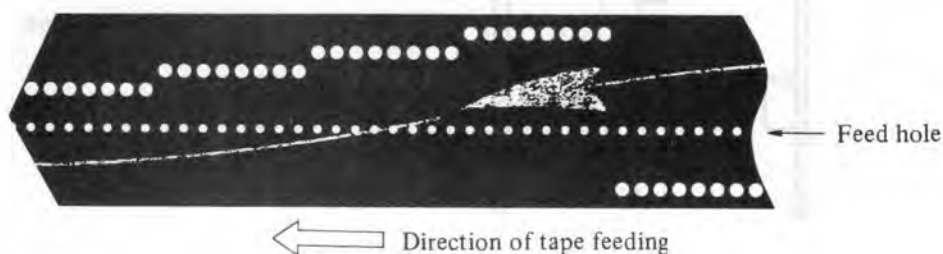


(iv) Twist the reel hub in the direction of the arrows, until the hubs are secured.

- ② Open the Tape Reader door.
- ③ Set the control switch to the RELEASE position.
- ④ Mount the reel with the NC tape on the boss located on the right side, and an empty reel on the left boss. Push the screw pawl inward, mount the reel, line pawl with one of reel slots, pull the pawl outward to secure the reel. The reel with the NC tape must be loaded to allow counterclockwise feeding of the NC tape (see the figure below).
- ⑤ Lift the Light Source.
- ⑥ Thread the NC tape on rollers and tension arms.



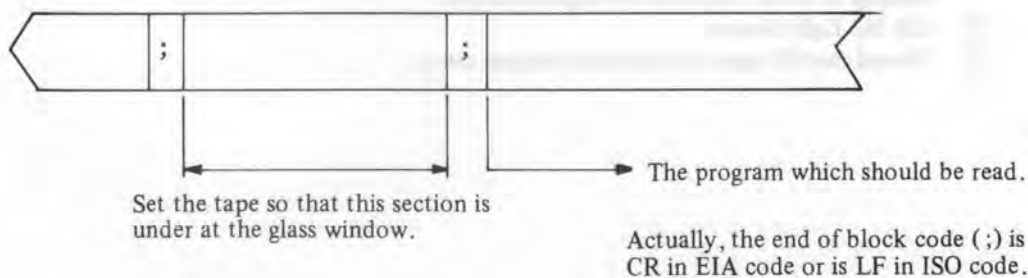
- ⑦ Roll the top of the NC tape on the take-up reel. Roll the tape several turns to protect idle running of the reel.
- ⑧ Check that the tape is received as shown in the figure below, when viewed downward.



- If this condition is not obtained, the reel is mounted incorrectly.
- ⑨ Check that the tape is correctly positioned by the Tape Guide.
 - ⑩ Lower the Light Source.
 - ⑪ Change the Tape Reader control switch to the REEL ON position. The reel should be turned slightly, and tensioned by the Tension Arms. If the reel turns continually, tape is not received by the reel correctly. Repeat steps from ⑦ for the take-up reel, or from ① for the feed reel. If the reel boss runs idly while the reel stays still, repeat steps from ④ to secure the reel.
 - ⑫ Close the door.

(Note 1) Precautions in tape loading

When the NC tape is loaded, the Label Skip function should be activated to read but skip data until the first End of Block code (CR in EIA code or LF in ISO code) is read. When loading an NC tape, the location within the tape, from which data reading should be started must be properly selected and the NC tape should be set as shown in the figure below.



(Note 2) *Disconnection and connection of tape reader connection cable*

Don't disconnect or connect NC ↔ tape reader connection cable (signal cable) without turning off the NC and tape reader power supply when using a separate tape reader, otherwise the PCB of the tape reader and master PCB of NC controller may be broken.

Turn off the NC and tape reader power supply before disconnecting or connecting the connection cable, accordingly.

(2) Unloading the NC tape

(a) Tape Reader without Reels

- ① Open the tape reader door.
- ② Turn the switch to the RELEASE position.
- ③ Lift the Light Source and remove the tape.
- ④ Lower the Light Source.
- ⑤ Close the door.

(b) Tape Reader with Reels (when reels are not used)
Identical to the above.

(c) Tape Reader with Reels (when reels are used)

- ① Open the Tape reader door.
- ② Push the REWIND button to rewind the tape. When the ER (EIA) or % (ISO) code is read, the rewind stops. Then, push the REWIND button again to rewind the tape completely. Push the STOP button to stop the rewinding.
- ③ Change the Tape Reader switch to the RELEASE position.
- ④ Push the boss pawl inward, and pull the reel toward you.
- ⑤ Close the door.

4. OPERATION OF THE OPERATOR'S PANEL

4.1 Operator's Panel

The operator's panel varies in functioning and switch arrangement with different machine tools.

Operation of a typical operator's panel is explained in Fig. 4.1.1. However, for details, refer to the manual issued by the machine tool builder.

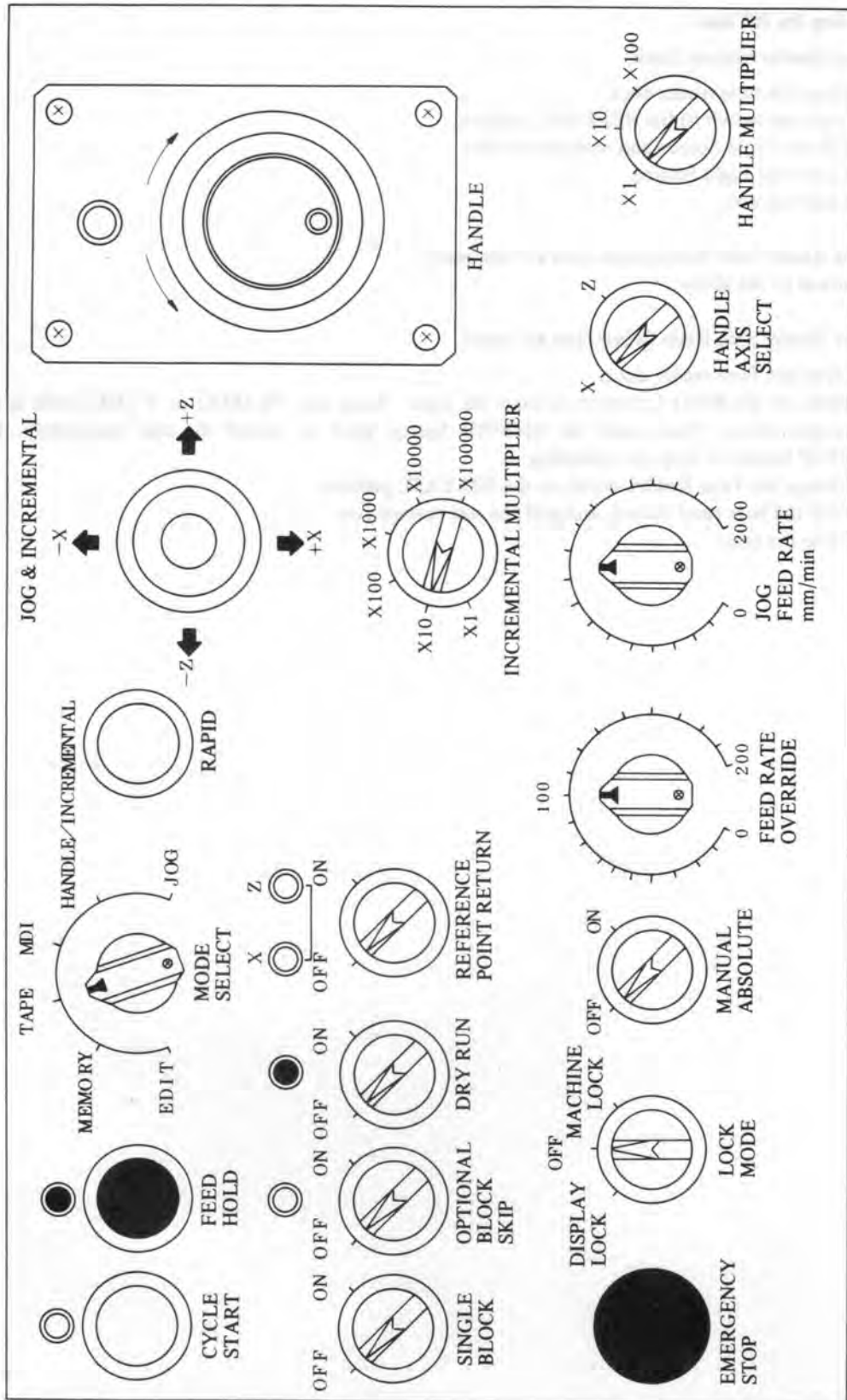


Fig. 4.4.1 Example of operator panel

4.2 Emergency Stop

This button is pushed to stop in an emergency. If this button is pushed the feed is stopped immediately, and the button remains latched in the stop position.



The method of release varies with the machine tool builder.

Usually, this button is released by turning it clockwise.

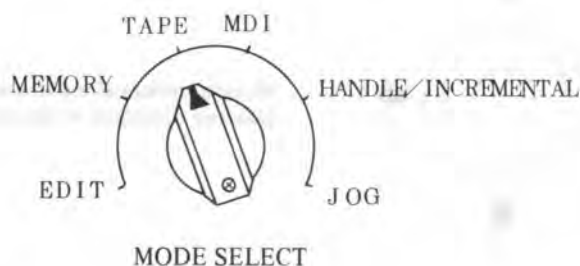
(Note 1) If this button is pushed, the current to the motors is interrupted.

(Note 2) The control unit is reset.

(Note 3) Remove fault causes before the button is released.

(Note 4) After the button is released, reference point return by manual operation or G28 is required.

4.3 Mode Select



This switch is used to specify an operational mode.

Mode	Function
EDIT	The following program edit can be executed. (i) Registration of program in memory (ii) Modification, addition and deletion of program (iii) Punch-out of program in memory
MEMORY	(i) The program stored in memory can be executed. (ii) A sequence number search can be executed for a program in memory.
TAPE	(i) Operation can be executed by reading a program via the tape reader. (ii) Sequence number search can be executed for commands on an NC tape.
MDI	(i) Operation via the MDI & DPL panel can be executed.
HANDLE/ INCREMENTAL	(i) HANDLE feed or INCREMENTAL feed can be executed.
JOG	(i) JOG feed can be executed.

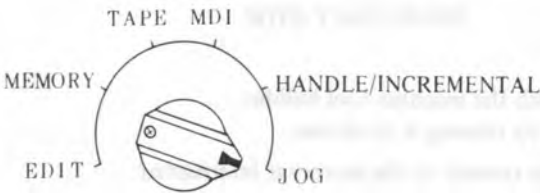
4.4 Operation Relative to Manual Operations

The following manual operations can be performed by using the switches, buttons, manual pulse generator, etc. on the operator's panel.

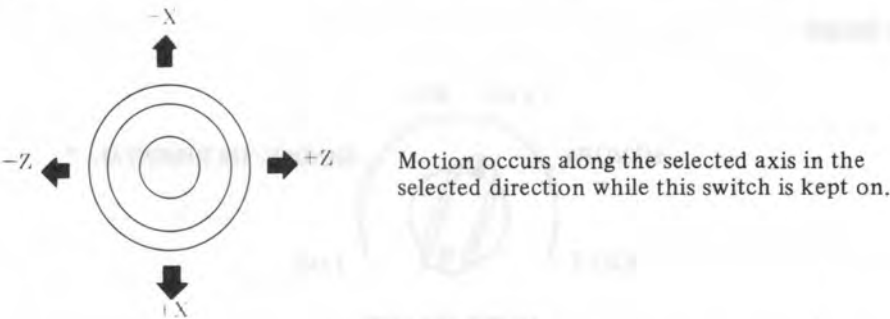
4.4.1 JOG feed

It is possible to move the tool continuously until a limit is reached.

- (1) Set the MODE SELECT switch to the JOG position.

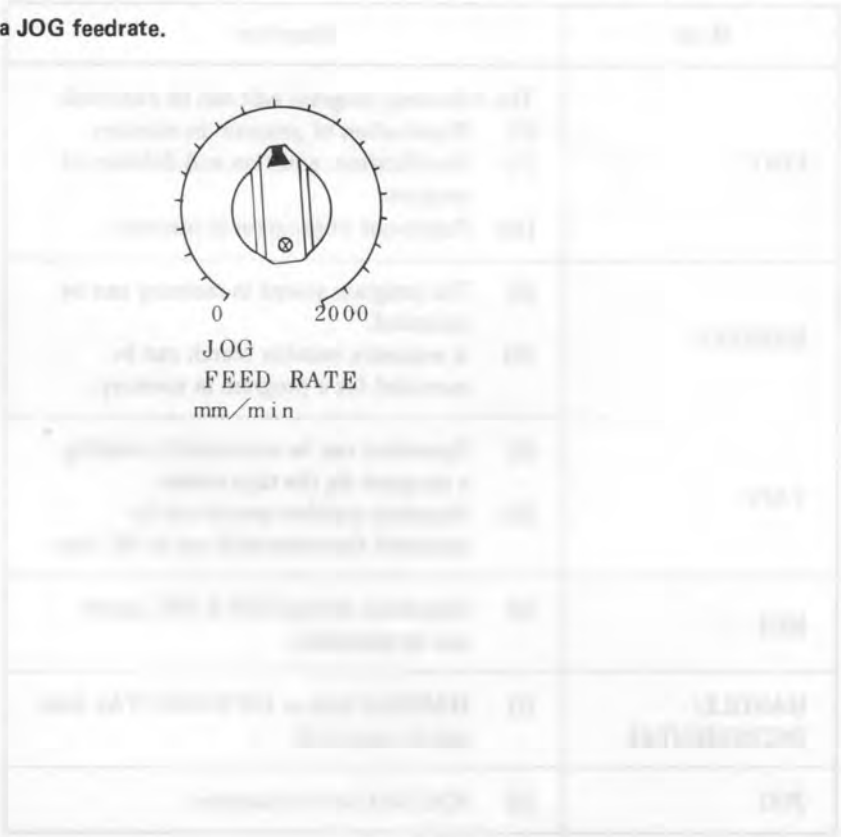


- (2) Select an axis.



(Note 1) Even if the MODE SELECT switch is set to the JOG position under an axis being selected, the tool does not move immediately after the power is turned on. In this case, an axis must be selected again.

- (3) Select a JOG feedrate.



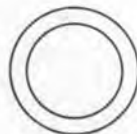
Position on rotary switch	JOG feedrate			
	Metric feed screw		Inch feed screw	
	mm/min	inches/min	inches/min	mm/min
0	0	0	0	0
1	1.0	0.04	0.02	0.508
2	1.4	0.055	0.028	0.711
3	2.0	0.079	0.04	1.02
4	2.7	0.106	0.054	1.37
5	3.7	0.146	0.074	1.88
6	5.2	0.205	0.104	2.64
7	7.2	0.283	0.144	3.66
8	10	0.394	0.2	5.08
9	14	0.551	0.28	7.11
10	20	0.787	0.40	10.2
11	27	1.06	0.54	13.7
12	37	1.46	0.74	18.8
13	52	2.05	1.04	26.4
14	72	2.83	1.44	36.6
15	100	3.94	2.00	50.8
16	140	5.51	2.80	71.1
17	200	7.87	4.00	102
18	270	10.6	5.40	137
19	370	14.6	7.40	188
20	520	20.5	10.40	264
21	720	28.3	14.40	366
22	1000	39.4	20.00	508
23	1400	55.1	28.00	711
24	2000	78.7	40.00	1016

(Note 1) The value varies with the machine tool builder.

(Note 2) A feedrate error (about $\pm 3\%$) will exist for the feedrates in the above table.

(4) Rapid traverse

Press the RAPID TRAVERSE button to move along an axis at the rapid traverse rate.



RAPID
(RAPID TRAVERSE button)

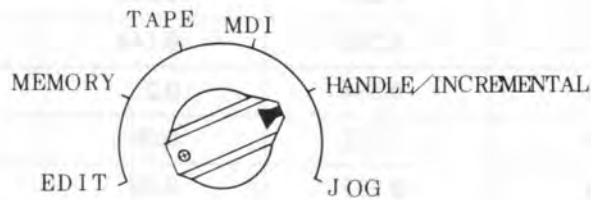
Motion along the selected axis proceeds at the rapid traverse rate while this button is pushed.

- (Note 1) The time constant, method of automatic acceleration/deceleration, and the traverse rate for manual rapid traverse are the same as those in the programmed command G00.
- (Note 2) When the stored stroke limit function is used, the feedrate does not become the rapid traverse rate but remains the jog feedrate even when the RAPID TRAVERSE button is pressed as long as the reference point return is not performed after turning the power on or an emergency stop on the axes with the reference point return function. This is to prevent movement along axes from reaching the stroke end at the rapid traverse rate since the stored stroke limit does not become effective until the manual reference point return is made.

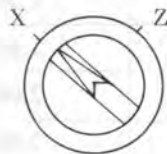
4.4.2 HANDLE feed

The feedrate can be adjusted precisely by using the manual pulse generator.

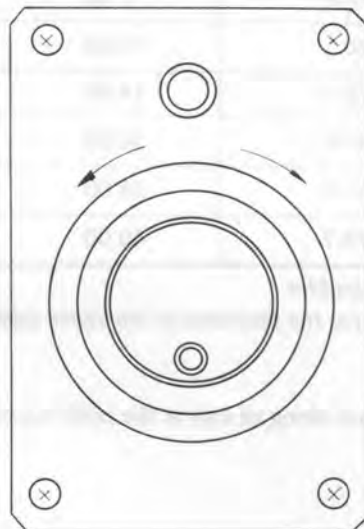
- (1) Set the MODE SELECT switch to the HANDLE position.



- (2) Select an axis.



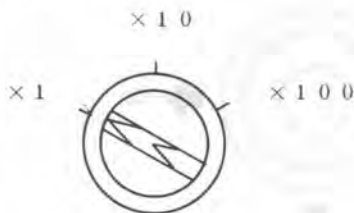
- (3) Rotate the handle of the manual pulse generator.



clockwise rotation +direction
counterclockwise rotation - direction
(The direction varies with the machine tool builder.)

(4) Movement amount

In some cases, the operator's panel is provided with the following selector switch. X10 multiplies the movement amount by 10; X100 by 100.



HANDLE MULTIPLIER

Input system	Movement amount per degree		
	x 1	x 10	x 100
Metric input	0.001 mm	0.01 mm	0.1 mm
Inch input	0.0001 inch	0.001 inch	0.01 inch

(Note 1) If the handle is rotated in excess of 5 turns/sec, there is a difference between the handle rotation amount and the machine movement distance.

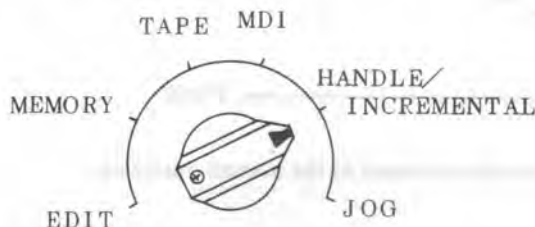
(Note 2) When the optional manual pulse generator is equipped, the incremental feed function is not used.

(Note 3) Rotating the handle too fast when X100 is selected moves the tool or table at a rate as fast as the rapid traverse rate. A sudden stop gives the machine tool a shock. The Automatic Acceleration/Deceleration option provided applies automatic acceleration/deceleration to the manual handle feed also, reducing mechanical shock.

(Note 4) For optional manual handle interruption, see section 4.7.

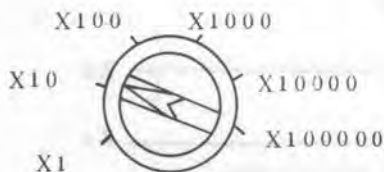
4.4.3 INCREMENTAL feed

(1) Set the **MODE SELECT** switch to the **INCREMENTAL** position.



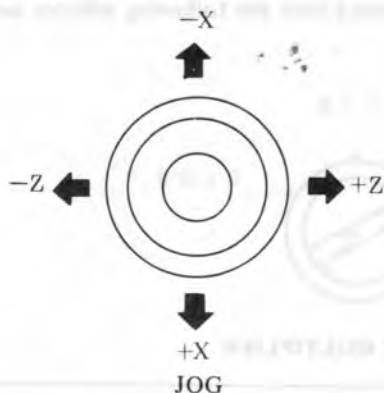
The INCREMENTAL feed is used in a machine without a manual pulse generator.

(2) Select the movement amount.



Input system	x 1	x 10	x 100	x 1000	x 10000	x 100000
Metric input	0.001 mm	0.01 mm	0.1 mm	1 mm	10 mm	100 mm
Inch input	0.0001 inch	0.001 inch	0.01 inch	0.1 inch	1 inch	10 inches

(3) Select an axis along which to move



The tool moves along the selected axis by the amount specified in the above table each time the axis select switch is turned on.

(Note 1) The incremental feedrate is the same as the JOG feedrate.

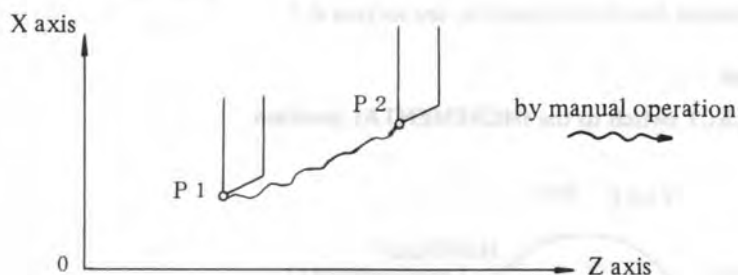
(Note 2) The rapid traverse button can be used. If it is, the rapid traverse override can be used.

(Note 3) Depending on the machine tool, sometimes none of the values X1 to X100000 can be used.

4.4.4 Manual absolute

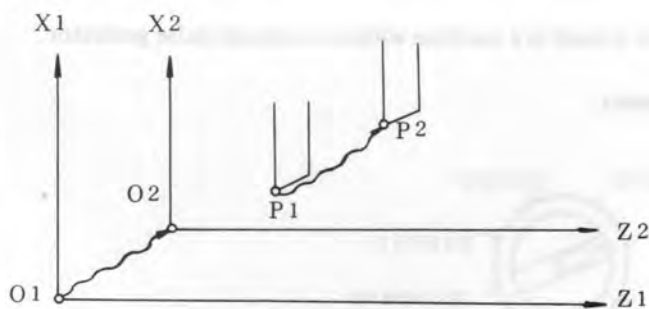
This switch is used to select whether or not the movement amount in a manual operation is added to a coordinate value.

(1) Manual absolute switch ON



The coordinate value varies by the value attained in the manual operation.

(2) Manual absolute switch OFF

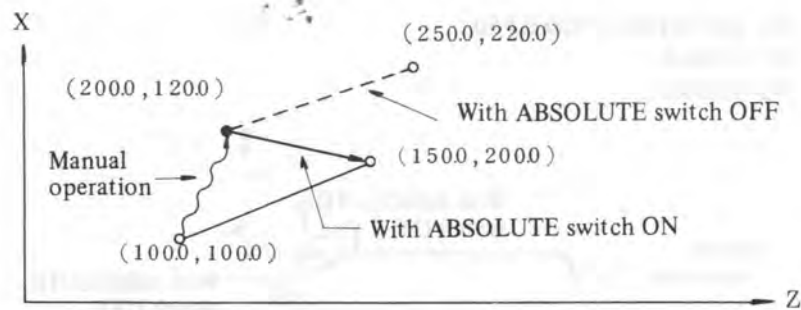


The coordinate value does not vary.

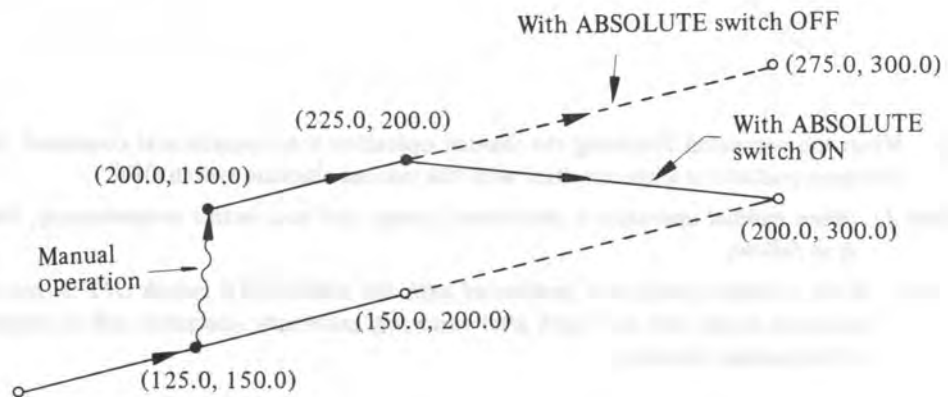
(Example) Suppose the next tape specifies:

```
G01X100.0Z100.0F010; ①
X150.0Z200.0 ; ②
X200.0Z300.0 ; ③
```

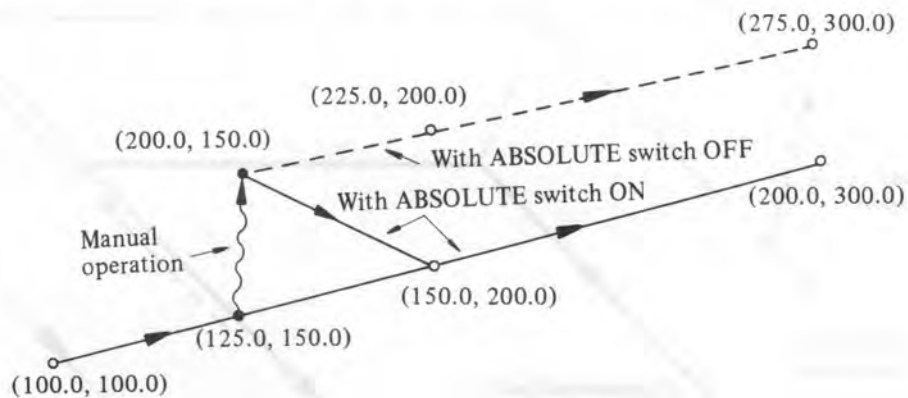
- (a) Upon completion of the shift in block ①, block ② is executed after a manual operation process (+100.0 in the X-axis direction, +20.0 in the Z-axis direction).



- (b) The FEED HOLD button is pushed during execution of block ②, a manual operation ($X + 75.0$) is performed and the CYCLE START button is pushed and released.

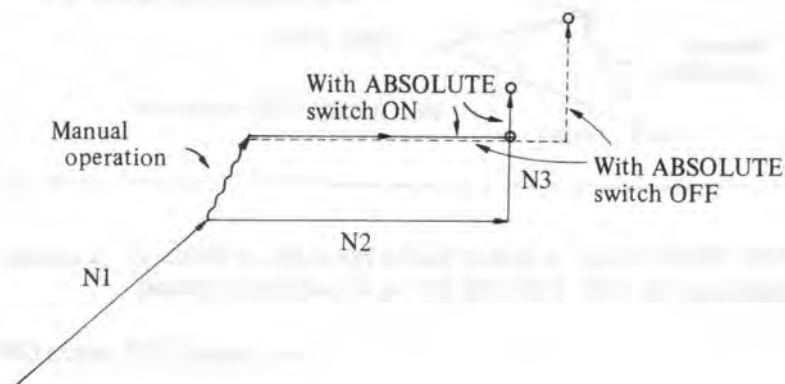


- (c) The FEED HOLD button is pushed during the execution of block ②, a reset is performed after a manual operation process ($X + 75.0$), and block ② is executed again.



- (d) When the command following the manual operation is a one axis command the tool returns to the programmed absolute position in only the specified axis.

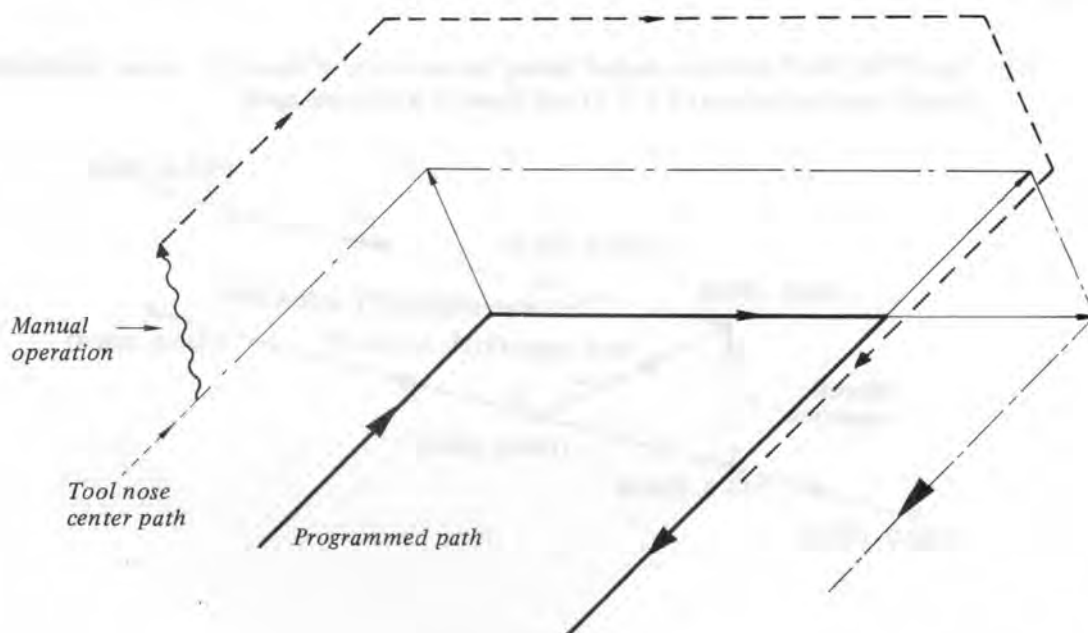
```
N1 G01 X100.0 Z100.0 F50;  
N2 Z200.0 ;  
N3 X150.0 ;
```



- (e) When the command following the manual operation is an incremental command, the tool moves to the same position as when specified with the manual absolute switch OFF.

(Note 1) When manual operation is performed during tool nose radius compensation, the actual tool path is as follows:

- (1) If the manual operation is performed with the ABSOLUTE switch OFF in tool nose radius compensation mode, the tool path after restarting automatic operation will be shifted by the amount of the manual operation



- (2) If manual operation is performed with the ABSOLUTE switch ON in the tool nose radius compensation mode, the tool path after restarting automatic operation by absolute command is as follows:

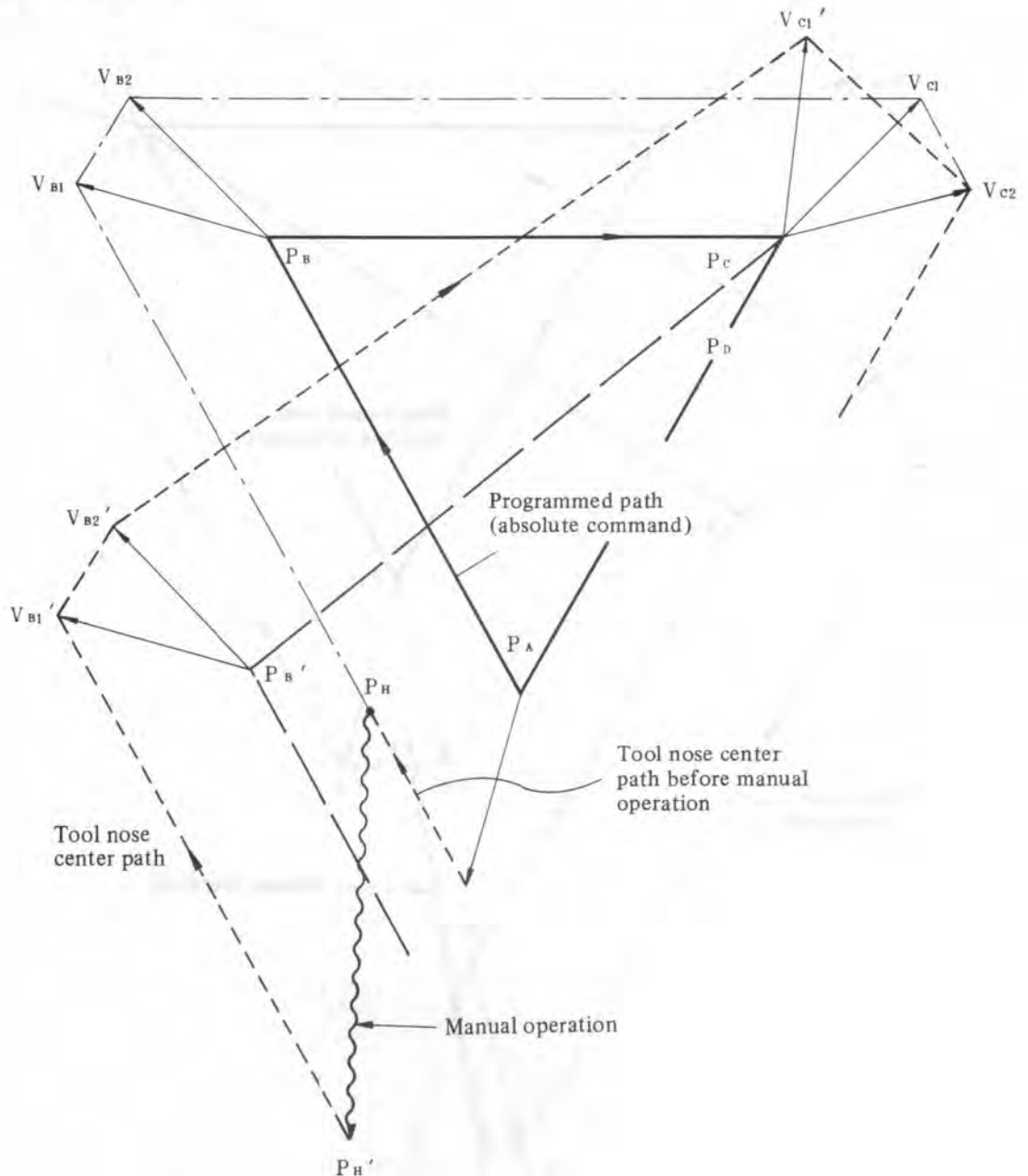
The tool path of the rest of the current block after the manual operation and the vectors at the start point of the next block are translated.

The tool path is calculated by the vector between next block and the block following it and by the movement amount attained by the manual operation.

If the manual operation is performed during corner movement, the tool path calculation is the same as above.

If the program is composed of incremental commands instead of absolute commands, the tool path is the same as when the ABSOLUTE switch is off.

- (a) A manual operation is performed during execution of a block.
(Example 1)



In the above programmed path ($P_A \rightarrow P_B \rightarrow P_C \rightarrow P_D$), suppose that the tool is translated from P_H between P_A and P_B to P_H' by manual operation (after pushing the feed hold button).

The end point P_B of the present block is translated to the point P_B' by the manual operation amount and the vectors (V_{B1} , V_{B2}) on P_B is also translated. (Translated vectors are V_{B1}' and V_{B2}' respectively).

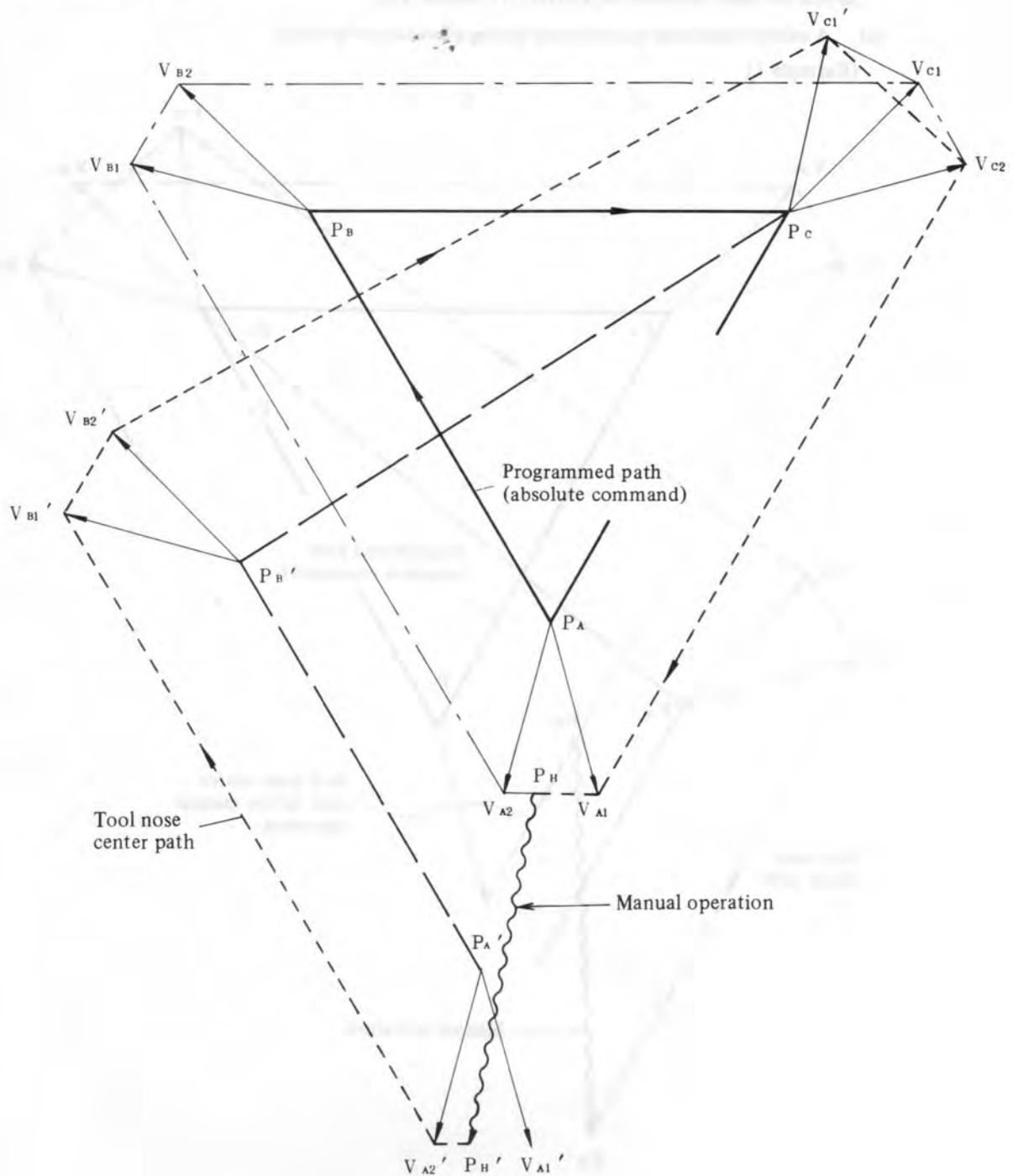
The vectors between the next block (tool path from P_B to P_C) and the following block (from P_C to P_D) are deleted.

The new vectors (V_{C1}' , V_{C2}') for compensation are formed by the relation between the two blocks (programmed path from P_B' to P_C and from P_C to P_D).

However, since the vector V_{B2}' was not recalculated, the tool nose radius compensation is not accurately performed for the tool path between P_B' and P_C .

The tool nose radius compensation is accurately performed for the blocks after point P_C .

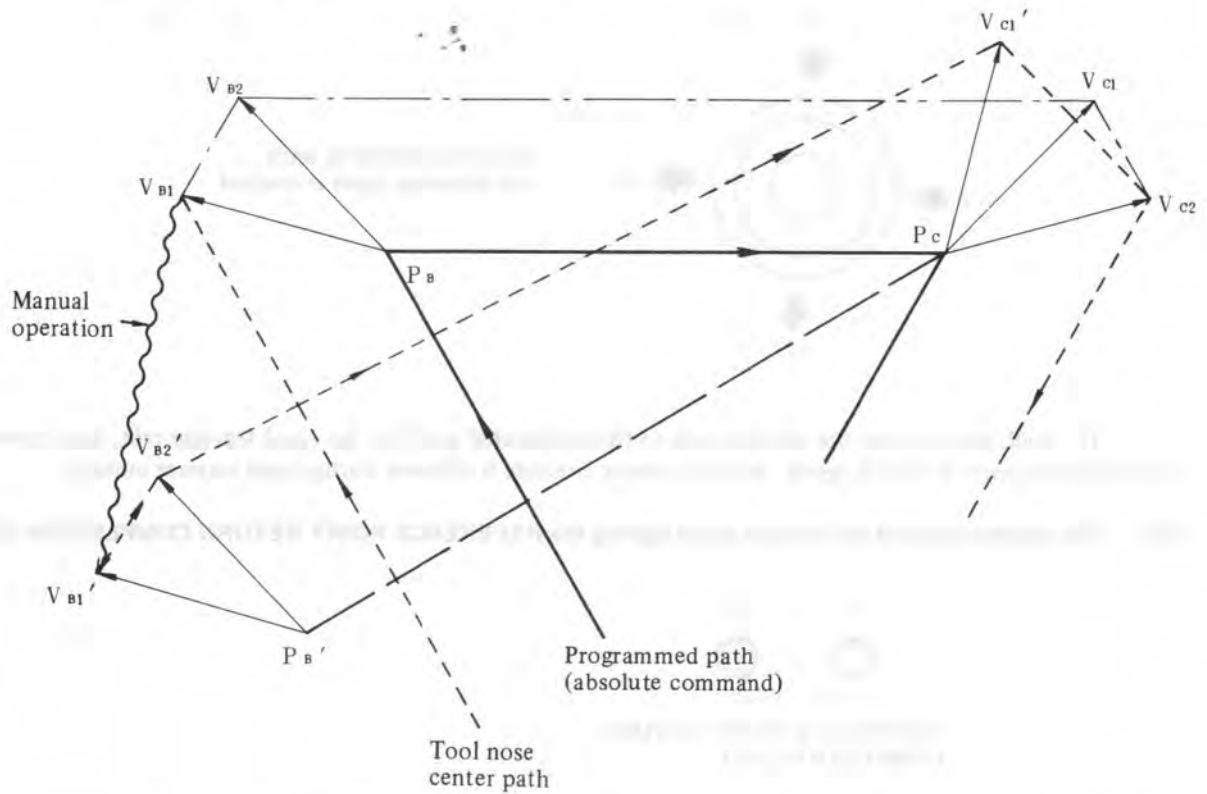
(Example 2)



If a manual operation is performed during corner machining in tool nose radius compensation, the tool path after that is determined in the same way as example 1.

That is, the vectors V_{A2}' , V_{B1}' and V_{B2}' in the above figure are calculated by translating the vectors V_{A2} , V_{B1} and V_{B2} by the manual operation amount respectively. New vectors are calculated by the vectors V_{C1}' and V_{C2}' . The tool nose radius compensation is accurately performed for blocks after point P_C .

- (b) When a manual operation is performed after a stop by the single block operation.

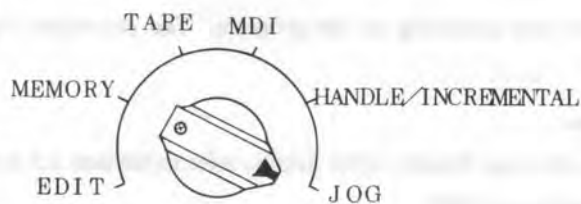


Vectors V_{B1} and V_{B2} at the end point of the present block are translated. The tool path after that is determined in the same way as mentioned in (a). An MDI command can be specified after a stop by the single block operation. And the move after the MDI command is the same as when a manual operation was performed.

4.5 Manual Reference Point Return

The tool can be returned to the reference point by manual operation.

- (1) Set the **MODE SELECT** switch to the **JOG** position.

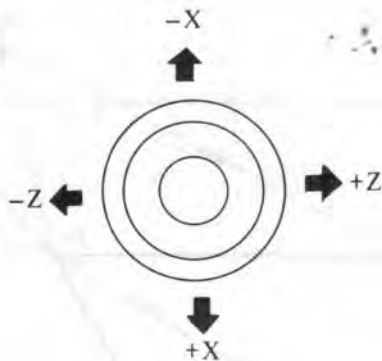


- (2) Turn on the **REFERENCE POINT RETURN** switch.



REFERENCE POINT RETURN Switch

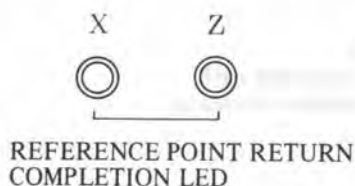
- (3) Select the desired axis.



Keep this switch on until the reference point is reached.

The tool moves along the selected axis to the decelerated point at the rapid traverse rate, then moves to the reference point at the FL speed. A rapid traverse override is effective during rapid traverse motion.

- (4) The machine stops at the referent point lighting the **REFERENCE POINT RETURN COMPLETION LED**.



(Note 1) Once the **REFERENCE POINT RETURN COMPLETION LED** lights at the completion of reference point return, the tool does not move unless the **REFERENCE POINT RETURN** switch is turned off.

(Note 2) The **REFERENCE POINT RETURN COMPLETION LED** is extinguished by either of the following operations:

- (i) Moving from the reference point.
- (ii) Entering a emergency stop status.

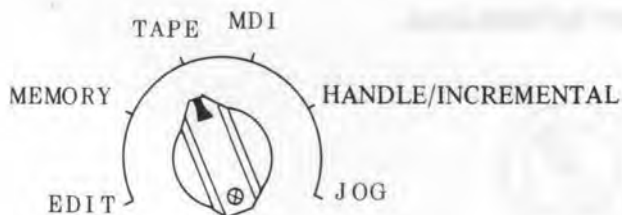
(Note 3) For the distance to return the tool to reference point, refer to the manual issued by the machine tool builder.

4.6 Automatic Operation

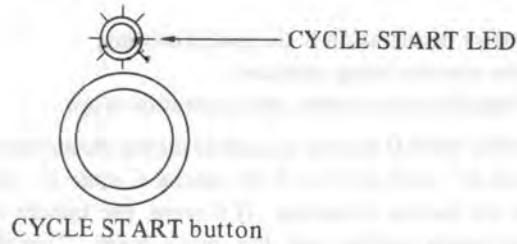
Automatic operation moves the tool according to the program. The procedure for automatic operation is as follows:

4.6.1 Starting automatic operation

- (i) Load the program tape on the Tape Reader. (For details, refer to Section 3.3 in Chapter IV)
- (ii) Set the **MODE SELECT** switch to **TAPE**.



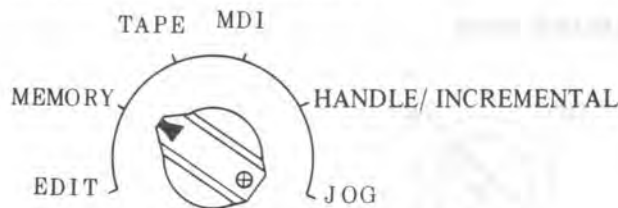
- (iii) Push the CYCLE START button.



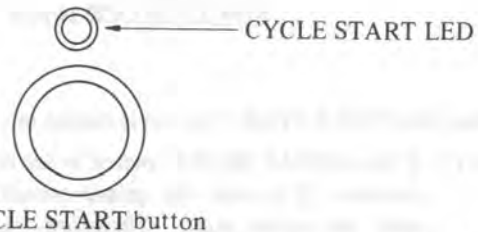
When the CYCLE START button is pushed, automatic operation starts and the CYCLE START LED lights.

(2) How to start operation according to a program stored in memory

- (i) Select the program. See Section 5.14.
(ii) Set the MODE SELECT switch to MEMORY



- (iii) Push the CYCLE START button



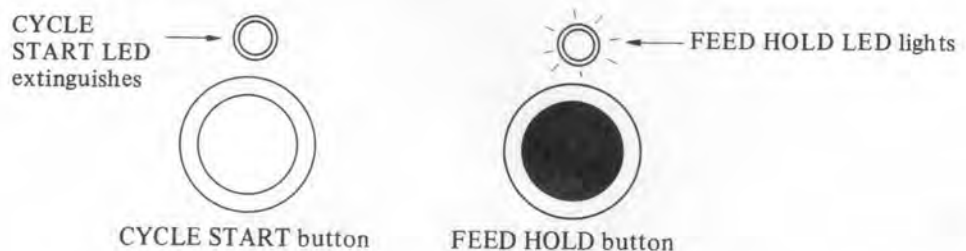
When the CYCLE START button is pushed, automatic operation starts and the CYCLE START LED lights.

(Note) In the following cases, the CYCLE START button is ignored:

- (a) When the FEED HOLD button is being pushed.
- (b) When the EMERGENCY STOP button is being pushed.
- (c) When the RESET signal is turned on. (For details, contact the machine tool builder.)
- (d) When the MODE SELECT switch is set to an inappropriate position. (Mode other than TAPE or MEMORY)
- (e) When a sequence number is being searched for.
- (f) When an alarm has occurred.
- (g) When automatic operation is being performed. (Not in feed hold or stop status).
- (h) When the NC is not READY.

4.6.2 Halting automatic operation

- (1) Push the FEED HOLD button**



When the FEED HOLD button is pushed, the CYCLE START LED is extinguished and the FEED HOLD LED is lit. And,

- (i) The feed is stopped after deceleration if the tool is moving.
- (ii) Dwell is stopped if the dwell is being executed.
- (iii) Effective M, S or T function is executed, then operation stops.

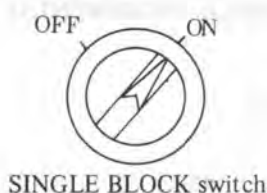
(Note) Even if the FEED HOLD button is pushed during thread cutting specified in G32, G76 or G92 (special G code B: G33, G76 or G78, special G code C: G33, G78 or G21), the feed is not stopped when the button is pushed. If it were, the spindle would continue to revolve and the thread portion would collide with the cutter point. Therefore, the feed is stopped after the block (cycle following the block (cycle) executing thread cutting is executed.

However, if the FEED HOLD button is pushed in the block (cycle) following the block (cycle) executing thread cutting, the tool stops at that position.

4.6.3 Single block

When this switch is on, control executes only one block of information each time the cycle start button is pressed.

(1) Turn on the SINGLE BLOCK switch



When the CYCLE START button is turned on, one block is executed, then execution stops.

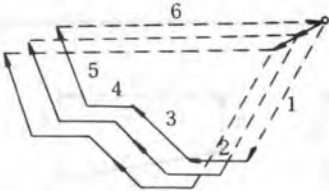
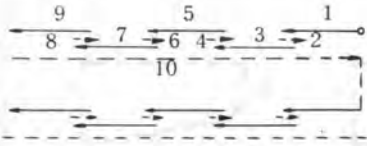
(Note 1) If the SINGLE BLOCK switch is set to on during thread cutting, the feed is not stopped at that position. If it were, the spindle would continue to revolve and the thread portion would collide with the cutter point. Therefore, the feed is stopped after the block following the block executing the thread cutting if the block is not a thread cutting block.

(Note 2) Movement also stops at the intermediate point when the SINGLE BLOCK switch is on for G28, G29 and G30 commands.

(Note 3) If the SINGLE BLOCK switch is set to on and a canned cycle G90, G92, G94, or G70 ~ G76 (G77, G78, G79, and G70 ~ G76 in the case of special G code B; G20, G21, G24 and G72 ~ G78 in the case of special G code C), the results of actual execution are shown below.

-----→ Rapid Traverse
 -----→ Feed

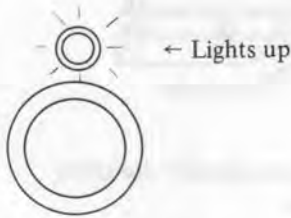
G code			Tool path	Explanation
Standard	Special B	Special C		
G90	G77	G20		Tool path 1 to 4 is assumed as one cycle. After 4 is finished a stop is made.
G92	G78	G21		
G94	G79	G24		
G70	G70	G72		Tool path 1 to 7 is assumed as one cycle. After 7 is finished, a stop is made.
G71 G72	G71 G72	G73 G74	(Note) This figure shows the case for G71 (or G73). G72 (G74) is the same.	Each tool path 1 to 4, 5 to 8, 9 to 12, 13 to 16 and 17 to 20 is assumed as one cycle. After each cycle is finished a stop is made.

G code			Tool path	Explanation
Standard	Special B	Special C		
G73	G73	G75		Tool path 1 to 6 is assumed as one cycle. After 6 is finished, a stop is made.
G74 G75	G74 G75	G76 G77	 Note) This figure shows the case for G74 (or G76). G75 (or G77) is the same.	Tool path 1 to 10 is assumed as one cycle. After 10 is finished, a stop is made.
G76	G76	G78		Tool path 1 to 4 is assumed as one cycle. After 4 is finished, a stop is made.

(Note 4) The single block stop function has no effect on M98P____; M99; G65, G66, or G67 blocks but effects M98 and M99 blocks which contain addresses other than O, N, L and P.

4.6.4 Restart after Feed Hold

- (1) Return the MODE SELECT switch to the mode under which a feed hold was made. (The mode will either TAPE or MEMORY mode)
- (2) Push the CYCLE START button.

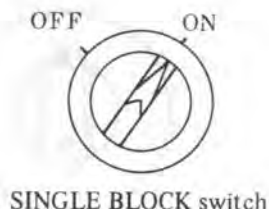


CYCLE START button

When the CYCLE START button is pushed, the FEED HOLD LED is extinguished, and the CYCLE START LED is lit.

4.6.5 Manual operation during automatic operation

- (1) Push the FEED HOLD button on the operator's panel, or set the SINGLE BLOCK switch to ON, to make a temporary stop.



- (2) Record the coordinate value of the stop position shown on the position display unit.
- (3) Perform the manual operation (see Section 4.4.4).
- (4) Return the tool to the recorded coordinate value (the start point of the manual operation).
- (5) In order to resume automatic operation, set the MODE SELECT switch to the position it was in prior to the manual operation.
- (6) Push the CYCLE START button.

4.6.6 MDI operation during automatic operation

- (1) Set the SINGLE BLOCK switch to ON. One block is executed, then execution stops.
- (2) Set the MODE SELECT switch to MDI.



- (3) Perform the operation from the Manual Data Input unit.
- (4) In order to resume automatic operation, set the MODE SELECT switch to its previous position.
- (5) Push the CYCLE START button.

(Note 1) Since old modal data specified in an NC command has been effective, new modal data (ex: G code, movement value in canned cycles) must be specified, if necessary, before the MDI operation is executed.

(Note 2) Tool nose radius compensation specified in MDI input is ignored.

(Note 3) The modal data specified during the MDI operation affects the automatic operation resumed after the MDI operation.

(Note 4) MDI operation cannot be performed during a temporary FEED HOLD stop.

4.6.7 Optional block skip

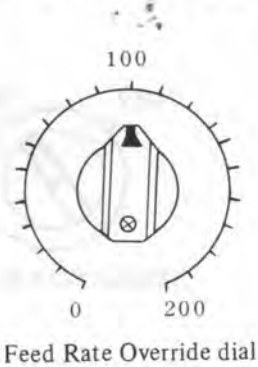
This function allows control to skip a block of information in which a slash "/" is punched as the first character.



(Note) Whether or not a block with a slash is skipped is determined when the program from the tape reader or memory is stored in the buffer storage. So if the block has already been stored in buffer storage, the block is not skipped even if this switch is set on.

4.6.8 Feedrate override

An override from 10 to 200% in increments of 10% can be specified for the feedrate specified in the F function.



- (Note 1) A JOG feedrate can also be set with this switch depending on the machine tool builder.
- (Note 2) The feedrate override set by this dial is ignored during thread cutting. That is, the override is always 100% during thread cutting.

4.6.9 Dry run

If this switch is set ON in the Cycle Operation, the feedrate specified by the program is ignored and becomes as follows:

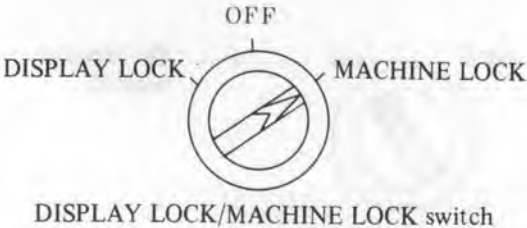


Rapid traverse button ON/OFF	Rapid traverse	Cutting feed
Rapid traverse button ON	Rapid traverse	Max. JOG feedrate
Rapid traverse button OFF	JOG feedrate (Note 1)	JOG feedrate

- (Note 1) This speed can be set to the rapid traverse rate by setting parameter RDRN.
- (Note 2) Dry run in thread cutting can be made effective or ineffective by parameter setting TDRN.

4.6.10 Machine lock

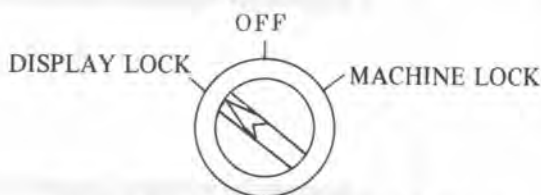
When the switch is set at the MACHINE LOCK position, move command pulses are suppressed. Consequently, the position display is updated as specified by the program, but the tool does not move. This function is used to check a program.



- (Note 1) Even if a G27, G28 or G30 command is specified, the tool is not moved to the reference point and the REFERENCE POINT RETURN COMPLETION LED does not go on.
- (Note 2) The M, S and T functions are executed.

4.6.11 Display lock

When the switch is set to DISPLAY LOCK, the relative coordinate value display is not updated even while the tool is moving.



DISPLAY LOCK/MACHINE LOCK switch

When shifting a coordinate system by manual operation, use this switch so that the shift amount does not affect the display.

4.6.12 Mirror image

The direction of movements on the X axis is opposite to those specified in the program during automatic operation when the Mirror Image switch is on.

Movements during manual operation and between the intermediate point and reference point in automatic reference point return are not inversed even while the Mirror Image switch is on. The X axis display is not inversed either.

This function is used when the machine has an opposite tool post.

4.6.13 Rapid traverse override

The optional rapid traverse override switch of 100%, 50%, 25% and Fo can be provided on the machine operator's panel. (Not shown in the example operator's panel.) When the feedrate is 10 m/min. and this switch is set to 50%, the actual feedrate is 5 m/min. Fo is the fixed value (feedrate) specified by each machine tool builder. This function is available in the following cases.

- (1) During rapid traverse motion specified in G00
- (2) During rapid traverse motion in a canned cycle
- (3) During rapid traverse motion specified in G27, G28, G29 or G30
- (4) During manual rapid traverse
- (5) During rapid traverse motion in manual reference point return

4.6.14 External workpiece number search function

The number of a workpiece to be machined can be specified by pressing a switch (Not shown in the example operator's panel in this manual) on the machine control panel. Then when machining programs have been stored previously in part program memory and the start button is pressed, the program corresponding to that workpiece number is automatically executed. Use of this function rids the operator of tape replacement and of searching memory for the program number, helping to reduce idle time and operating errors.

(1) Program preparation

To use this function, a program number must be defined corresponding to the number of a workpiece to be machined. Using an arbitrary number assigned to each workpiece to be machined, prepare programs with corresponding program numbers written as follows:

○ (:) ○ ○ □ □ (○ for EIA, : for ISO)

└────────── arbitrary workpiece number

And store them into the part program memory.

Each program must start with address O followed by a program number and end with M02, M30 or M99 as shown below.

Programs not related to workpiece numbers may also be stored.

(Example)

O0001; N001G00 -----; ----- ----- N120M02	} A program for workpiece number 01
O002; N001G00 -----; ----- ----- N300M30;	} A program for workpiece number 02
O004; N001G00 -----; ----- ----- N080M02;	} A program for workpiece number 04
O6247; N001G00 -----; ----- ----- N034M99;	} A program not related to workpiece numbers

- (Note 1) Each program must start with address O followed by a program number and end with M02, M30 or M99. But neither M02, M30 or M99 may be specified in the middle of a program. If they are, the subsequent program is assumed as another program. (The sequence number of the block immediately after M02, M30 or M99 is assumed to be the program number when the program is stored into memory.)
- (Note 2) The number of workpiece numbers allowed depends on the machine tool builder. Refer to the appropriate manual.
- (Note 3) When your machine tool builder supplies an external workpiece number search function A, the maximum workpiece number allowed is 31. In this case the upper two digits of the program numbers which correspond to workpiece numbers must be 00.

(2) Operation

Operation differs with the machine tool builder. The standard operation is described in this manual. Use the operation specified by your machine tool builder.

- Prepare a tape or a number of tapes programmed in the format mentioned above and store them in memory. Either one tape storing several programs or a number of tapes each storing one program will do. (See Section 5.15 to 5.17 for how to store a tape in memory.)
- Set the mode selector switch to MEMORY, set the number (01 ~ 31) corresponding to a workpiece to be machined with the rotary switch on the control panel, then press the start button. The program corresponding to the workpiece number will be searched and executed from the beginning.
- If the start button is pressed after the workpiece number has been set to 00, no program is searched for but execution starts from the then executable portion. To start from the middle of a program or to execute a program not related to the workpiece number, press the start button with the workpiece number set to 00 after the sequence number search or the program number search.

- (Note 4) This function cannot be used in Tape or MDI operation. It can only be used in Memory operation.
- (Note 5) When there is no program number corresponding to a workpiece number on the rotary switch in memory, alarm No. 59 will be generated when the start button is pressed.
- (Note 6) Even if a workpiece number is selected on the dial, the corresponding program is not always selected. Refer to the manual issued by your machine tool builder for the operation used to select the program. When workpiece number search function A is selected, the NC searches for a program at the start of cycle operation in the reset status.

4.7 Manual Handle Interruption Function

4.7.1 General

As regards a specific axis (which is fixed by a parameter) during automatic operation, handle movement can be performed in addition to its automatic movement.

4.7.2 Handle interruption operation

Handle interruption can be performed by turning the manual pulse generator in the following situations.

(1) Mode

TAPE mode, MEMORY mode or MDI mode

(2) Operating state

Handle interruption can be performed during linear or circular interpolation except following cases.

- (i) Alarm
- (ii) Start lock on
- (iii) Movement by positioning
- (iv) Interlock (when in effect)
- (v) There is no move command in either X or Z.

(c) Handle axis selection signal

The handle axis selection signal (HX or HZ) is on (the contact is closed) in the axis in which an attempt has been made to perform handle interruption.

4.7.3 Movement by handle interruption

(1) Amount of movement

The movement by handle interruption, which is the same as that of manual handle feed, is determined by the graduation on the manual pulse generator and the magnification of handle feed ($\times 1$, $\times 10$, and $\times 100$). This movement is added to the amount realized by automatic operation.

(2) Movement speed

The speed in an axis to which handle interruption was applied is the movement speed by automatic operation plus that of handle interruption. As a result, when the rapid traverse speed is exceeded in that axis, the movement speed is tied to the rapid traverse speed.

This discards the excess over the rapid traverse speed, resulting in mismatching between the amount of movement and the graduation on the manual pulse generator.

(3) Relation between movement and signals below

Signal	Movement
Machine lock	Affected, i.e., the tool will not move when machine lock is on.
Display lock	Affected, i.e., the relative coordinate value will not change when display lock is on.
X-axis mirror image	Not affected, i.e., the handle movement is made in the positive direction by rotating the handle in the pulse direction.

(4) Relation between movement and displays below.

Display	Movement
Absolute coordinate value	Not affected, i.e., pulses are not added to the absolute coordinate value by handle interruption.
Relative coordinate value	Added, i.e., pulses are added to the relative coordinate value.
Machine coordinate value	Added, i.e., pulses are added to the machine coordinate value.

(5) Display of amount of movement

Movement by handle interruption can be displayed by Diagnose (DGNOS).

	8	0	5
--	---	---	---

Movement in X by handle interruption

	8	0	6
--	---	---	---

Movement in Z by handle interruption

Unit: 0.001 mm (mm input)

0.0001 inch (inch input)

Note: Movement can be cleared by a clear operation.

5. DISPLAY AND OPERATION ON THE CRT CHARACTER DISPLAY MDI & DPL PANEL

The MDI & DPL panel, usually attached at the top front side of the control unit, consists of a CRT display and buttons as in the following figure.

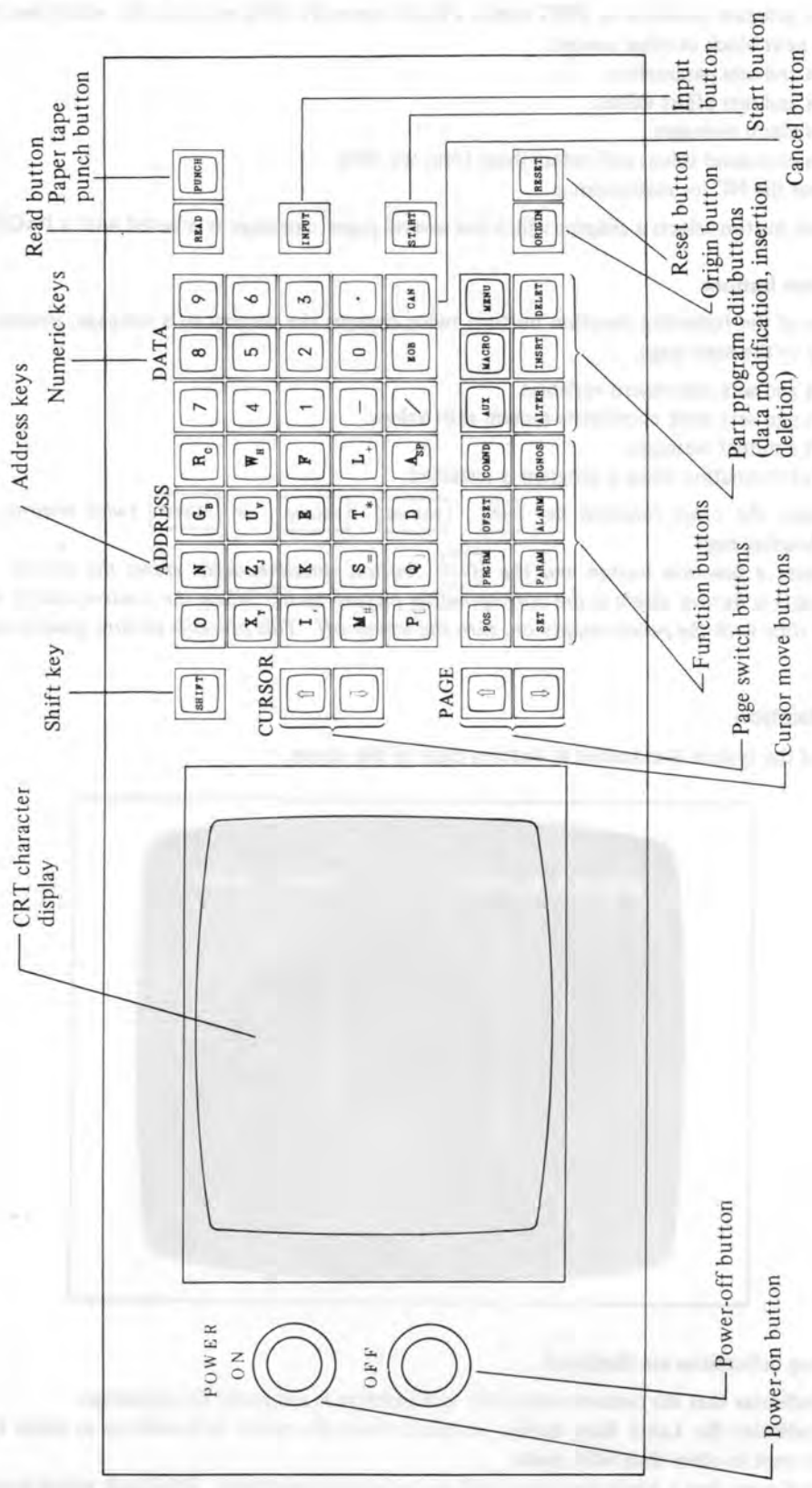


Fig. 5.1 CRT character display MDI & DPL panel

Function button

The function button indicates a large item such as a chapter in a book.

POS	Displays and resets the current position.
SET	Displays and sets setting data.
PRGRM	Displays program contents in EDIT mode; a block currently being executed (or which has been executed) and the next block in other modes.
PARAM	Displays and sets parameters.
OFFSET	Displays and sets offset values.
ALARM	Displays alarm messages.
COMND	Displays command values and enters them from the MDI.
DGNOS	Diagnoses the NC for maintenance.

Each function button selects a chapter which has several pages; one page is selected with a PAGE button.

Subpages of function buttons

Pressing one of the following function buttons twice changes the display to a subpage; pressing it once more returns the display to the main page.

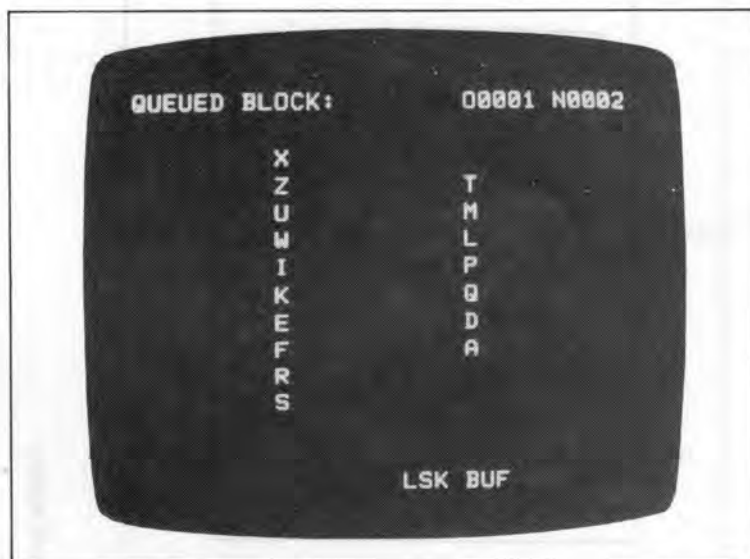
SET	Displays and sets user macro variables.
OFFSET	Displays and sets work coordinate system shift values.
ALARM	Displays external messages.
COMND	Displays information when a program is restarted.

(Note 1) Pressing the other function key **POS** , **PARAM** , **PRGRM** , or **DGNOS** twice returns the display to the heading page.

(Note 2) Pressing a function button and the **CAN** button simultaneously erases the screen. Then pressing a function button displays the corresponding picture again. When the control unit is not in use for a long time with the power supply on, turn the screen off. This prevents picture quality deterioration.

5.1 Status Indication

The status of the system is indicated at bottom right of the screen.



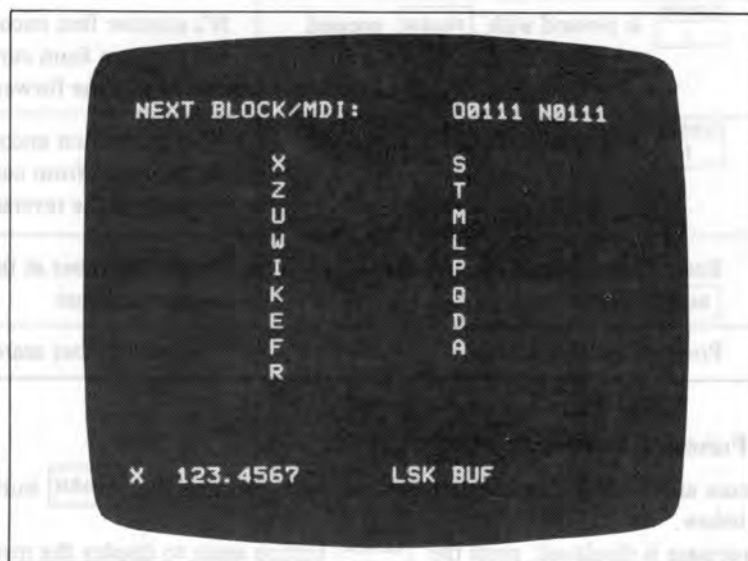
The following indications are displayed.

NOT READY	Indicates that the control unit or the servo system is not ready for operation.
LSK	Indicates the Label Skip status, produced when the power is turned on or when the control unit is reset in other than MDI mode.
BUF	Indicates that a block has been read in, but not yet executed. The block which is not yet executed disappears when a reset is executed in other than MDI mode.

- ALM Indicates that an alarm was generated. The type is displayed by pressing the **ALARM** button (this indication flickers).
- EDIT Indicates that editing is being performed (this indication flickers). Stop the editing operation while this indication is on.
- SRCH Indicates that a Sequence Number Search is being performed (this indication flickers).

5.2 Keyed Data Display

Input from an address key or from a numeric key is displayed at the bottom left.



Data cannot be keyed in when a picture displayed by pressing the **POS** or **ALARM** button is displayed.

When not editing a program, only one word consisting of an address and a number can be keyed in. Pressing the **CAN** key cancels the one word.

In program editing, one word, several words, one block, and any character string up to 32 characters can be keyed in.

Pressing the **CAN** key cancels the last character of keyed-in data. If the **CAN** key is pressed continuously, keyed-in characters are canceled one after another from the end.

(Note 1) Pressing the **PRGRM** button in the edit mode enables program editing.

5.3 Program Number and Sequence Number Display

A program number and a sequence number are displayed at the top, as seen in the photo.



Indications of program and sequence numbers are as follows.

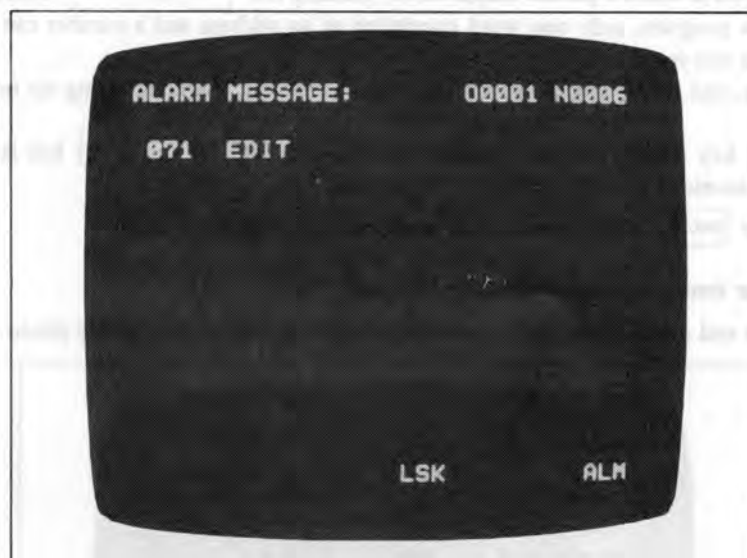
Mode	Operation	Displayed
Other than EDIT	Other than below	Sequence number executed last
	Sequence number search	Sequence number read moment by moment during search
MEMORY	 is pressed with  pressed	Program number at the head of a current program
EDIT	 is pressed with  pressed	N's number first encountered, looking at the program from current position of memory in the forward direction
	 is pressed with  pressed	N's number first encountered, looking at the program from current position of memory in the reverse direction
	Reset status entered by pressing  , etc.	Program number at the head of a current program
MEMORY	Program number search	Program number searched for

5.4 Alarm Display (Function button:)

When an alarm occurs and ALM is displayed at bottom right, pressing the  button displays the alarm message, as in the photo below.

When an operator message is displayed, press the  button again to display the message.

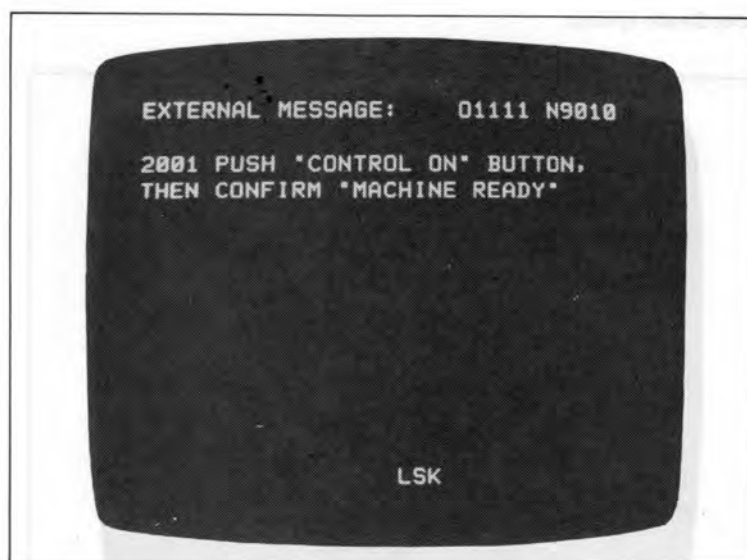
See Appendix 7 for the meaning of the alarm number.



(Note) When an alarm occurs, the alarm message is usually automatically displayed.

5.5 Operator Message

When an operator message is sent from the machine tool, it is automatically displayed on the screen.



When the operator message is to be displayed after selecting the other screen, push the **ALARM** button. And when the alarm message is displayed, press the **ALARM** button again.

5.6 Display and Reset of a Current Position (Function button: **POS**)

- (1) Press the **POS** button.
- (2) Press a **PAGE** button. Data will be displayed in one of the following three ways.
 - (i) In the relative coordinate system



The relative position is displayed where the position reset by an operator is zero.

Reset

When this is displayed, press the **U** or **W** key. The address will flicker. Then press the **ORIGIN** button. The relative position of the flickering address will be reset to zero.

- (ii) In the work coordinate system



The current value in a programmed coordinate system set with G50, Automatic Coordinate System Setting, or by the following action (Reset) is displayed.

Reset ()

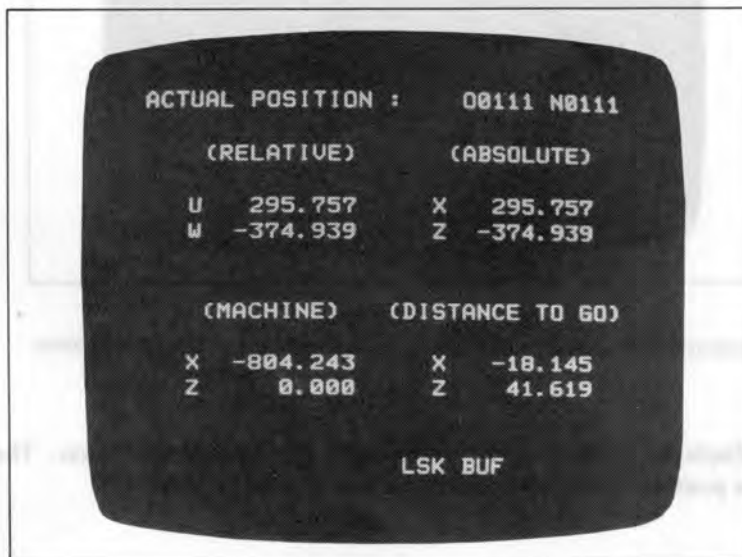
When this is displayed, press the or key. The address will flicker. Then press the button. The current position of the flickering address will be reset to zero

Note: Reset is possible only in an automatic operation stopped or a reset status.

- (iii) Overall display

The current position in the following coordinate systems is displayed simultaneously:

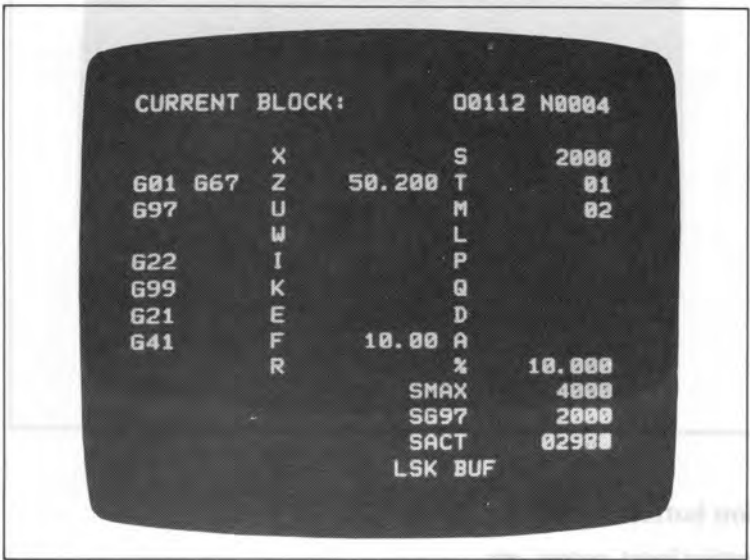
- (a) Position in the relative coordinate system (RELATIVE)
- (b) Position in the work coordinate system (ABSOLUTE)
- (c) Position in the machine coordinate system (MACHINE)
- (d) Residual movement amount (DISTANCE TO GO)



The residual movement amount indicates the movement amount remaining in one block of a command. When the overall display is on, the position in a particular coordinate system cannot be reset. The unit of the machine coordinate system is the same as that of the machine system.

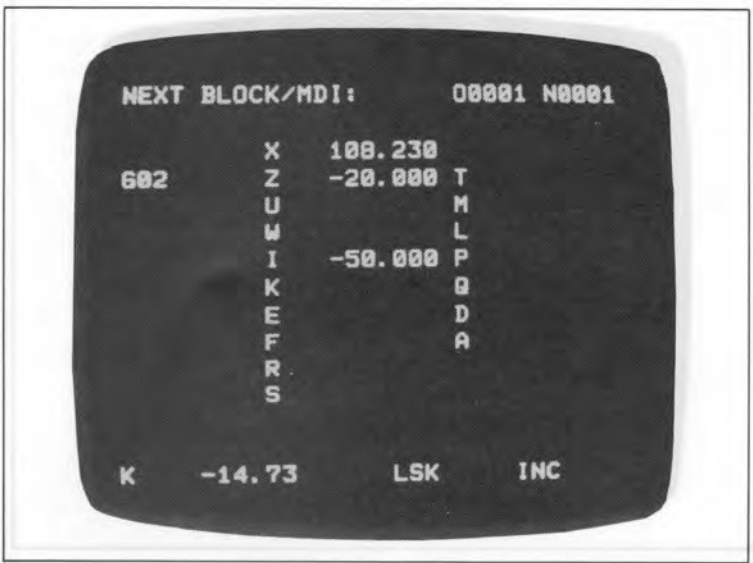
5.7 Command Value Display (Function button: COMND)

- (1) Press the COMND button.
- (2) Press a PGAE button. Data will be displayed in the following three ways.
 - (i) Command values being executed, and modal values previously specified are displayed.

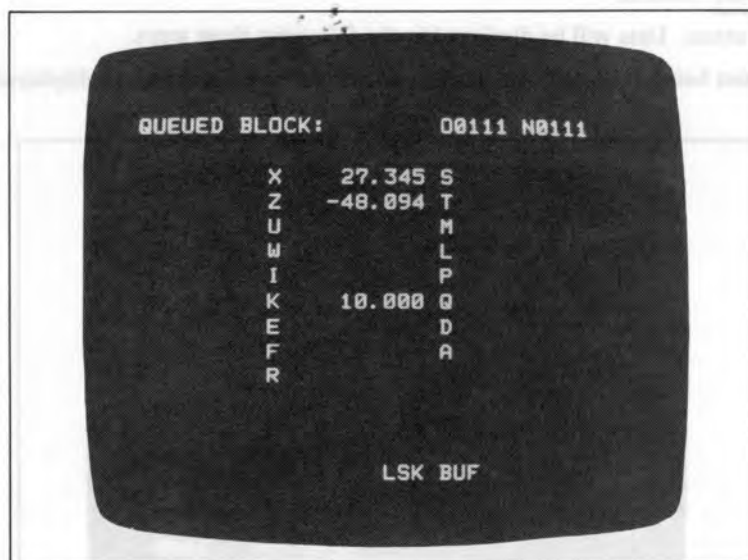


(Note) In the above photo, a numeral following the character % indicates the feedrate overridden.

- (ii) Command values entered from the MDI or command values to be executed next are displayed.



- (iii) The command to be executed next but one is displayed in the offset mode by tool nose radius compensation.



5.8 Setting (Function button:)

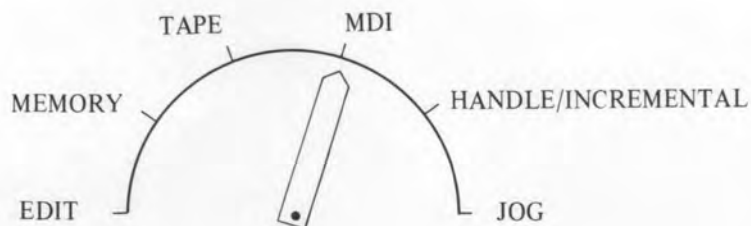
5.8.1 Display and setting input, output, etc.

- (1) Press the button.
- (2) Press a PAGE button. Setting and display is performed in the following two ways.
 - (i) Input and output



Setting ( /no key)

- (a) Set the mode selector switch to MDI.



- (b) Move the cursor to the item to be changed by pressing a CURSOR button.
The cursor does not move using N key.
- (c) Enter 1 or 0 according to the following table, using the P key.

Item \ Display	0	1
TV CHECK	No	Yes
PUNCH CODE	EIA	ISO
INPUT UNIT	MM	INCH
INPUT DEVICE 1	Tape reader	RMT
INPUT DEVICE 2	Either tape reader or RMT which is set to INPUT DEVICE 1, when registering a program tape into memory.	Input device by data number 340, when registering a program tape into memory.

Key in and or keys and the button in this order.

- (Note 1) Setting using an option which is not selected is impossible. If no inch/metric switch option is available, for example, INPUT UNIT = 1 cannot be set for a metric machine. Also if no ISO code input option is available, PUNCH CODE = 1 cannot be set.
- (Note 2) When G20 (INCH input)/G21 (MM input) is executed, the INPUT UNIT is automatically rewritten.
- (Note 3) The PUNCH CODE specifies ISO code or EIA code in punching, and is not related to input. In input, ISO or EIA code is automatically determined.
- (Note 4) A device to be used for punches is set in data number 341.

(ii) Others

SETTING DATA 02 : 00112 N0001			
NO.	DATA	NO.	DATA
0057	0002	0152	1750000
0058	0058	0153	1000000
0059	0028	0154	1500000
0064	0010	0155	9000000
0065	2000	0156	9000000
0066	0200	0157	8000000
0067	1000	0158	8000000
0068	1000	0319	00000000
0141	0002	0340	0000
0151	3000000	0341	0001
P		LSK	

Displayed numbers and their meanings are as follows:

Data number	Contents
057*	Run hour (in hours) (TMHOR)
058*	Run hour (in minutes) (TMMIN)
059*	Run hour (in seconds) (TMSEC)
064*	Chamfered width of screw (THDCH)
065*	Return value for G74 and G75 (GROVE)
066*	Finishing allowance for G76 (THDFN)
067*	Relief value for G71 and G72 (MRCDT)
068*	Minimum allowance for G76 (THCLM)
141*	Run hour (TIME 1)
151*	1st-vertex X-axis value of 2nd stored stroke limit
152*	1st-vertex Z-axis value of 2nd stored stroke limit
153*	2nd-vertex X-axis value of 2nd stored stroke limit
154*	2nd-vertex Z-axis value of 2nd stored stroke limit
155*	1st-vertex X-axis value of 3rd stored stroke limit
156*	1st-vertex Z-axis value of 3rd stored stroke limit
157*	2nd-vertex X-axis value of 3rd stored stroke limit
158*	2nd-vertex Z-axis value of 3rd stored stroke limit
319*	Various settings (PRG8, MSBL)
340*	Input device selection at program registration (IDVICE)
341*	Output device selection at punch out (ODVICE)

(Note 1) A blank is displayed in data numbers other than the above.

(Note 2) A parameter with the same data number can be set at address PRM.

(Note 3) Refer to the explanation of the parameter of the same data number.

(Note 4) Data numbers 340 and 341 are detailed as follows.

	3	4	0
--	---	---	---

IDVICE

	3	4	1
--	---	---	---

ODVICE

IDVICE: selects an input device to register a program in memory (is effective when INPUT DEVICE 2 = 1 (RS232C) is set.)

ODVICE: selects an output device for punching.

Setting	I/O device
0	Tape reader for input FACIT PUNCHER for output
1	ASR33/ASR43 for both input and output. Sets baud rate and other parameters to 310.
2	RS232C for both input and output. Sets baud rate and other parameters to 311.
3	RS232C for both input and output. Sets baud rate and other parameters to 312.
4	RS232C for both input and output. Sets baud rate and other parameters to 313.

Both can also be set as parameters.

Setting ( /no key)

- Set the mode selector switch to MDI.
- Move the cursor to the item to be changed by pressing a CURSOR button. The cursor does not move by using N key.
- Press the and numeric keys and the button in this order.

5.8.2 Display and setting of custom macro variable values

The values of all common variables and of local variables in custom macro body currently being called can be displayed on the CRT.



When a variable value is (vacant) (see Section 10.2.3), a blank is displayed. When an absolute value exceeds 99999999, OVER FLOW is displayed. When an absolute value is not 0 and under 0.0000001, UNDR FLOW is displayed.

Display

- Select subpage of SETTING
Press the button to display a SETTING page and press it again.
- Display covers six pages. Press the PAGE button to display a particular page.
Page 1 local variables #1 to #20 in current degree of multicalling
Page 2 local variables #21 to #33 in current degree of multicalling
Page 3 common variables #100 to #119
Page 4 common variables #120 to #139
Page 5 common variables #140 to #149
Page 6 common variables #500 to #509
- Move the cursor to the variable number to be displayed.
Method 1 Press the CURSOR button to move the cursor in sequence. Move the cursor beyond the page to shift the screen to the next page.
Method 2 Key in and the and press the button.

Setting ()

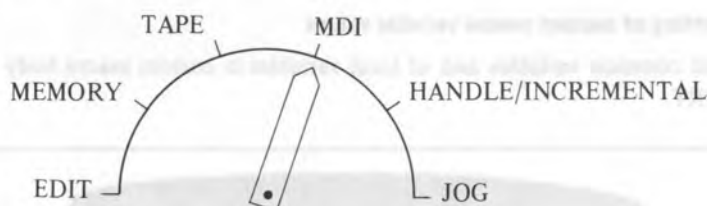
- Select MDI mode.
- After displaying variables and moving the cursor to the number of the variable to be altered, key in and the and press the button.

5.9 Operation from the MDI (Function button: COMND)

One command block can be entered from the MDI for execution.

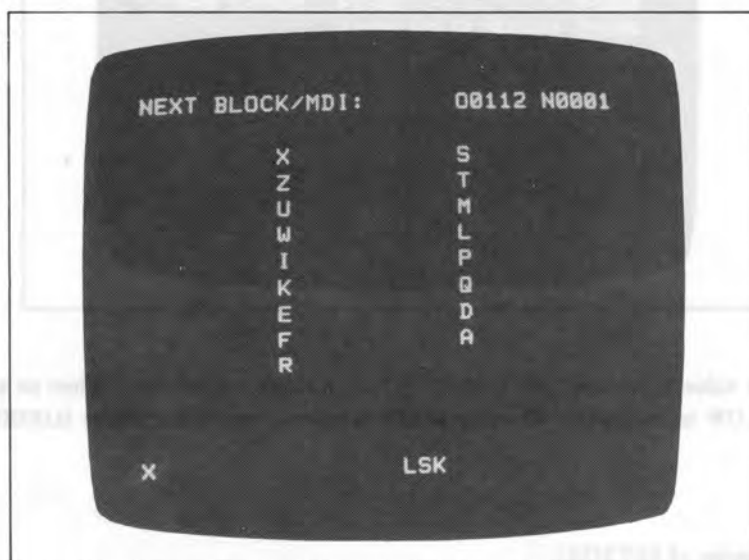
(1) For example, X10.5Z200.5;

- (a) Set the mode selector switch to MDI

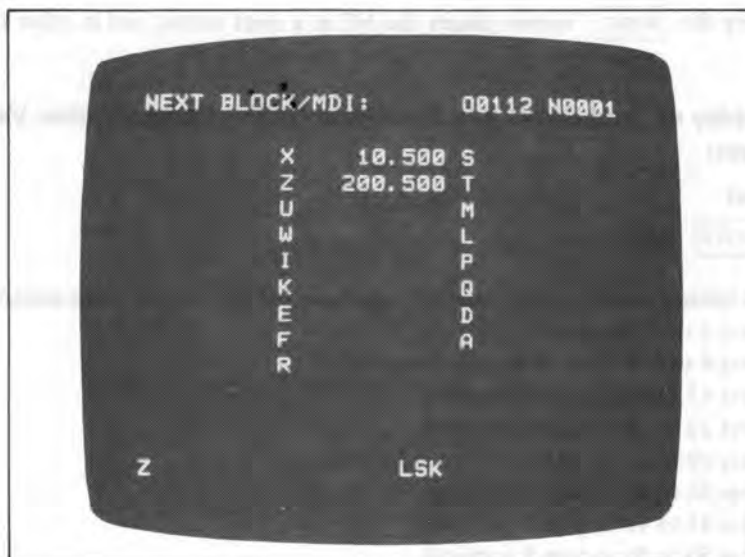


- (b) Press the COMND button.

- (c) Press a PAGE button to display NEXT BLOCK/MDI at the top left.



- (d) Press the X, 1, 0, ., 5 keys and the INPUT button in this order. If you are aware of an error in the keyed-in number before pressing the INPUT button, press the CAN button and key in the correct number again. If you notice an error after the INPUT button is pressed, key in the correct number again from the beginning.
- (e) Press the Z, 2, 0, 0, ., 5 keys and the INPUT button in this order. If you make a mistake keying in the number, follow the above instruction.



(f) Press the button. Press the cycle start button on the control panel (depending on the machine tool).

(2) Canceling Z200.5; of X10.5Z200.5; before pressing the button (cycle start button)

(a) Press the key and the and buttons in this order.

(b) Press the button or the cycle start button on the control panel.

(3) Cancelling modal data

Modal G codes, F, E, and T cannot be cancelled. Enter the correct data again.

5.10 Starting MDI Operation

Pressing executes commands entered from the MDI.

Note: Depending on the machine tool, pressing the button may start both tape operation and memory operation. Also depending on the machine tool, not pressing this button but pressing the start button on the control panel instead may start MDI operation.

5.11 Reset

Pressing executes a reset. This is generally used to release an alarm status.

Pressing the button places the NC in the following status.

Status before reset		Status after reset
Move commands are being executed.		The machine stops with deceleration and the residual movement disappears.
M, S, or T is being transmitted.		The transmission sequence stops. Refer to the description issued by the machine tool builder as to what happens to the machine tool at this time.
Buffer storage stores one block	MDI mode	Contents in buffer storage are not deleted.
	Mode other than MDI	Contents in buffer storage are deleted. BUF goes out.

In any case, pressing the **RESET** button places the NC in a reset status; and in other than MDI mode, in the Label Skip status.

5.12 Setting and Display of Tool Offset and Tool Nose Radius Compensation Values (Function button: **OFFSET**)

5.12.1 Absolute input

- (1) Press the **OFFSET** button.
- (2) Press a PAGE button to display the required page from the two to five pages available.
 Offset numbers 1 to 7 in page 1
 Offset numbers 8 to 9 or 8 to 14 in page 2 (option)
 Offset numbers 15 to 21 in page 3 (option)
 Offset numbers 22 to 28 in page 4 (option)
 Offset numbers 29 to 32 or 29 to 35 (option) in page 5
 Offset numbers 36 to 42 in page 6 (option)
 Offset numbers 43 to 49 in page 7 (option)
 Offset numbers 50 to 56 in page 8 (option)
 Offset numbers 57 to 63 in page 9 (option)
 Offset numbers 64 in page 10 (option)

NO.	X	Z	R	T
01	026.882	001.200	002.000	1
02	026.882	000.200	000.400	0
03	016.130	001.000	001.300	0
04	012.800	-004.000	000.000	0
05	010.500	000.990	002.000	2
06	006.000	000.400	002.200	0
07	006.600	000.666	002.200	1

ACTUAL POSITION : (RELATIVE)
 U 12.800 W -4.000

W LSK

Offset Values in Page 1

- (3) Move the cursor to the offset number to be changed.
Method 1 Continuously press the CURSOR button. The cursor will move in sequence. Moving the cursor beyond the page changes the screen to the next page.
Method 2 Key in N and an offset number and press the INPUT button.
- (4) Set the mode selector switch to a position other than EDIT.
- (5) Key in **X**, **Z**, **R** or **T**, and an offset number and press the **INPUT** button.
X, **Z** : Offset amount in X and Z axes respectively (Absolute)
R : Tool nose radius compensation value (Absolute)
T : Imaginary tool nose number

The following photo shows the screen when **Z**, **1**, **5**, **.**, **4** and **INPUT** are pressed at offset number 4.



5.12.2 Incremental input

Incremental Offset Input can be optionally selected. and select Incremental Offset Input. Key in as follows:

X	}	Offset value and INPUT .
Z		
U		
W		
R		
T		

X	,	Z	;	X and Z axes offset values (Absolute input)
U		W	;	X and Z axes offset values (Incremental input)
R			;	Tool nose radius compensation value (Absolute/incremental input)
T			;	Imaginary tool nose number

Whether R indicates absolute input or incremental input depends on the parameter IOF. Incremental Offset Input enters an offset increment or decrement value.

(1) Entering an incremental offset value

For the current tool nose R offset value 5.678, pressing the , , , keys and button in this order sets the offset value to 7.178.

(2) When desired to input the offset value with the absolute value when the incremental offset input is effected.

For the current offset value 5.678, pressing the , , and buttons makes the offset value zero. Then pressing , , , keys and the button in this order makes the offset value 1.5.

(Note 1) When the offset value is changed during cycle operation, whether the new offset value becomes valid with the next block or with the next T code, depends on the parameter TLCC.

5.12.3 When tool offset is divided between tool geometry offset and tool wear offset

(1) Display of tool wear offsets

Tool wear offsets are displayed in the 1st chapter of offset screen. (Press offset function button several times)

OFFSET/WEAR 01 01234 N3456				
NO.	X	Z	R	T
W01	00.435	000.050	002.000	1
W02	00.250	-000.320	000.025	0
W03	00.042	-001.400	000.100	0
W04	00.138	000.125	000.300	1
W05	-00.080	000.325	000.150	1
W06	00.400	-000.810	000.000	0
W07	-00.315	000.000	000.040	2
ACTUAL POSITION : (RELATIVE)				
U	10.000	W	12.000	
X	1.05	LSK		

(2) Display of tool geometry offsets

Tool geometry offset are displayed in the 3rd chapter of the offset screen. (Press offset function button several times.)

OFFSET/GEOMETRIC 01 01234 N3456				
NO.	X	Z	R	T
G01	009.520	045.500	008.000	1
G02	-017.400	028.400	001.500	0
G03	-012.000	074.000	002.000	0
G04	-015.300	032.250	002.000	1
G05	-005.100	052.000	001.000	1
G06	034.000	-008.500	001.500	0
G07	027.500	-026.100	001.800	2
ACTUAL POSITION : (RELATIVE)				
U	10.000	W	12.000	
X	22.300	LSK		

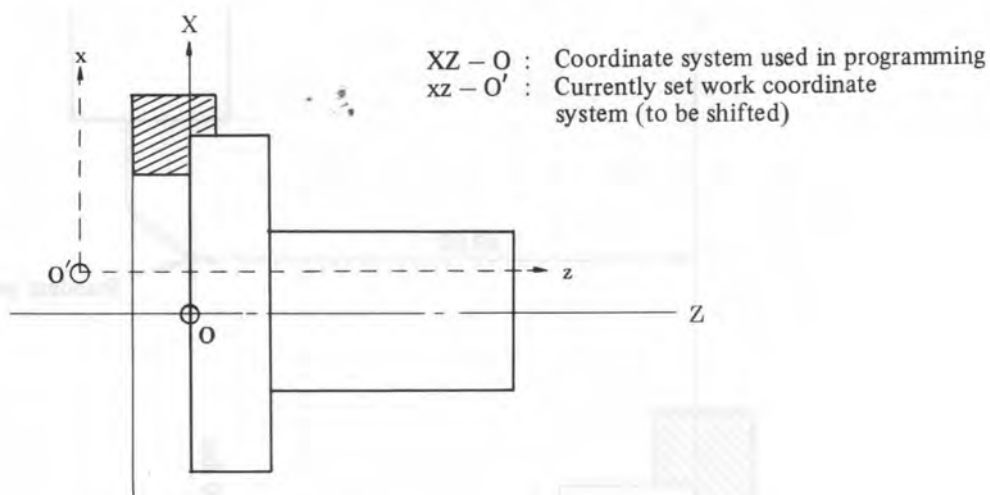
Setting

- Display tool wear offset or the tool geometry offset screen.
- Set as in item 5.12.1 or 5.12.2.

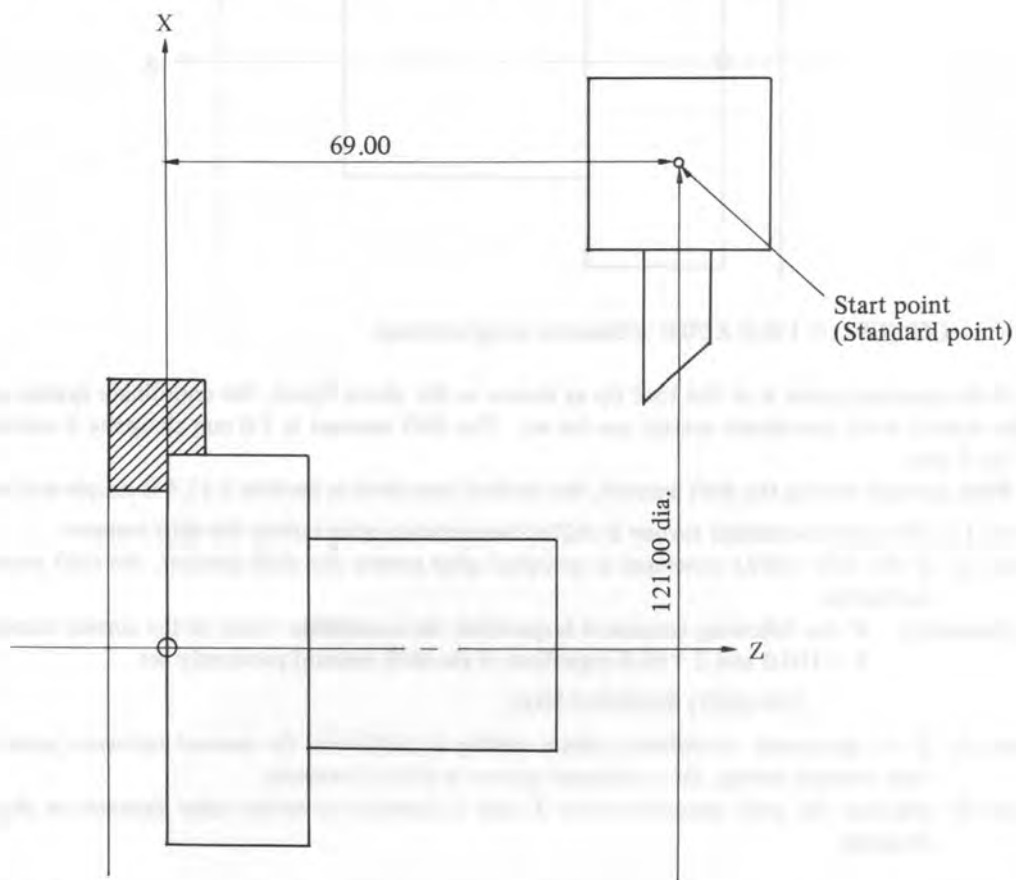
5.12.4 Setting work coordinate system shift amount

If the work coordinate system set with G50 (G92) or automatic coordinate system setting function is different from the coordinate system used in programming, the coordinate system set in advance can be shifted.

Store the shift amount in offset memory corresponding to offset number 00. The setting procedure for the shift amount is the same as that for the tool offset amount.



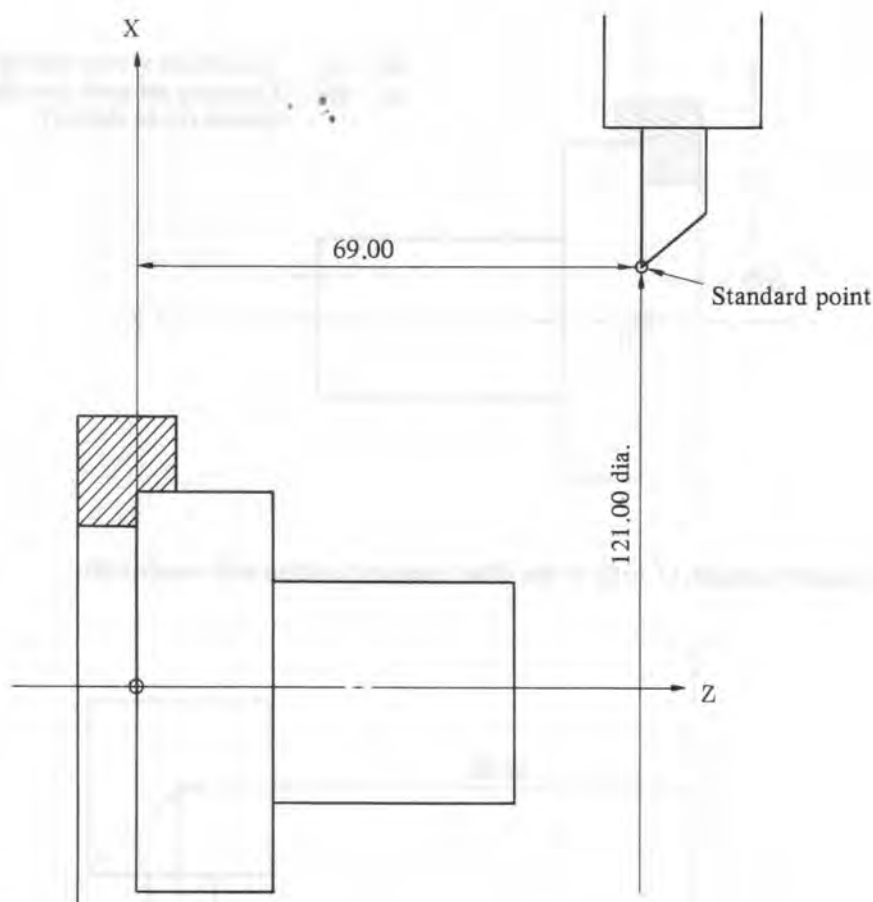
Set the shift amount, O' to O, in the offset memory location with number 00.



When the actual standard point is at the position of 121 mm (diameter value) on the X axis and 69 mm on the Z axis even though the following G50 (G92) command is executed,

G50 (G92) X 120.0 Z 70.0 ;

the desired work coordinate system can be set by shifting the current coordinate system 1.0 mm along the X axis and -1.0 mm along the Z axis.



G50 (G92) X 120.0 Z 70.0 ; (Diameter programming)

If the standard point is at the tool tip as shown in the above figure, the coordinate system can be shifted and the desired work coordinate system can be set. The shift amount is 1.0 mm along the X axis and -1.0 mm along the Z axis.

When actually setting the shift amount, the method described in Section 5.12.4 is simple and convenient.

(Note 1) The work coordinate system is shifted immediately after setting the shift amount.

(Note 2) If the G50 (G92) command is specified after setting the shift amount, the shift amount becomes ineffective.

(Example) If the following command is specified, the coordinate values of the current standard point are X = 100.0 and Z = 80.0 regardless of the shift amount previously set.

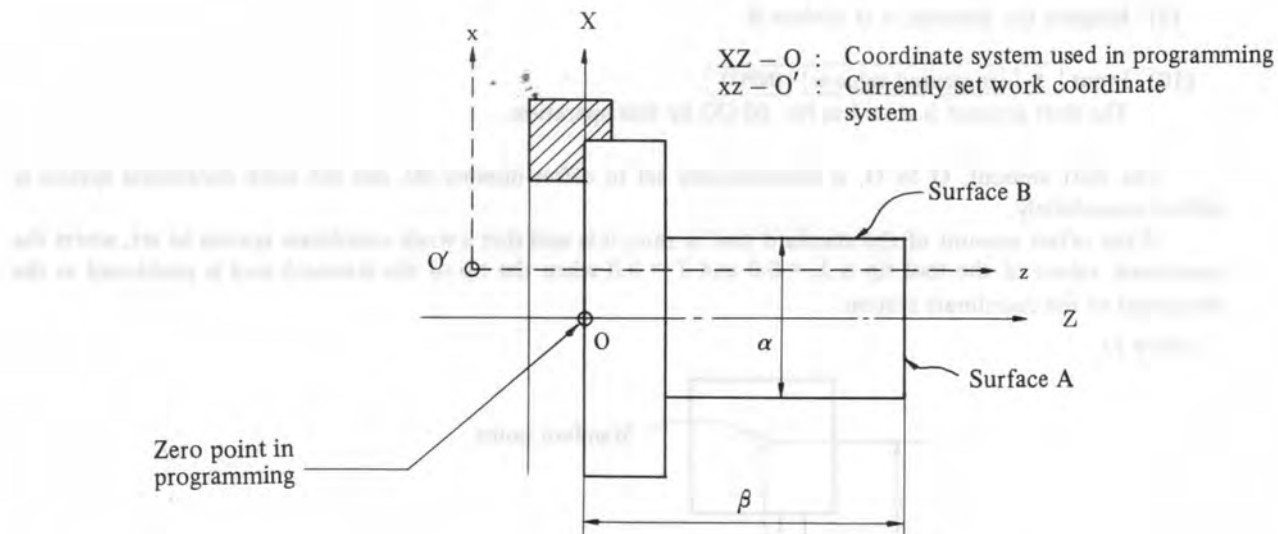
G50 (G92) X 100.0 Z 80.0 ;

(Note 3) If the automatic coordinate system setting is performed by manual reference point return after shift amount setting, the coordinate system is shifted instantly.

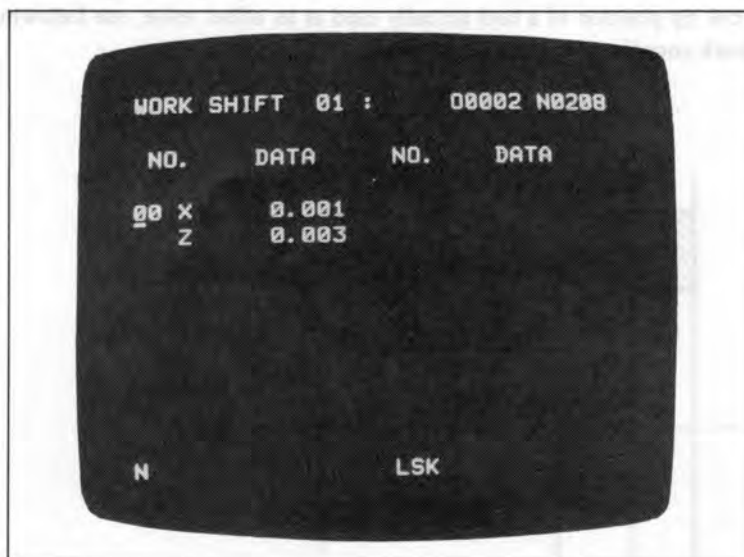
(Note 4) Whether the shift amount on the X axis is diameter or radius value depends on that specified in program.

5.12.5 Direct measured value input for work coordinate system shift

When the work coordinate system set with a G50 (G92) command or the automatic coordinate system setting function is different from the coordinate system used in programming, the coordinate system can be shifted by storing the measured distance directly instead of the shift amount as follows.



- (1) Cut the workpiece along surface A using the standard tool in manual operation.
- (2) Release the tool after pressing the tool measuring button on the machine operator's panel and stop the spindle.
- (3) Measure distance " β " in the figure above.
- (4) Press **OFFSET** button
When a page with offset number 0 (see the photo below) is not selected, press the **OFFSET** button once more.
- (5) Input **N** **1** **0** **0** **INPUT**
The page with offset number 0 is displayed and the cursor indicates number 0. Keyed-in N blinks.



- (6) Input **Z** **measured distance β** **INPUT**.
The shift amount is entered in Z of number 0.
- (7) Cut surface B by manual operation.
- (8) Release the tool after pressing the tool measuring button on the machine operator's panel and stop the spindle.

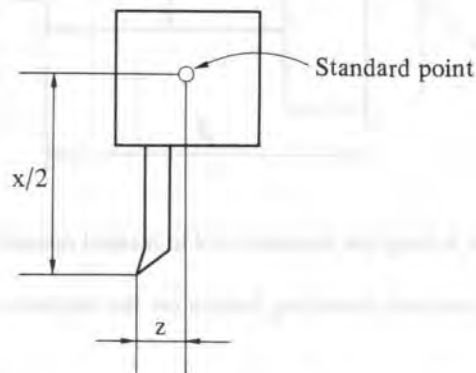
(9) Measure the diameter α at surface B.

- (10) Input
- The shift amount is stored in No. 00 (X) by this operation.

The shift amount, O' to O , is automatically set to offset number 00, and the work coordinate system is shifted immediately.

If the offset amount of the standard tool is zero, it is said that a work coordinate system be set, where the coordinate values of the tool tip is $X = 0.0$ and $Z = 0.0$ when the tip of the standard tool is positioned at the zero point of the coordinate system.

(Note 1)



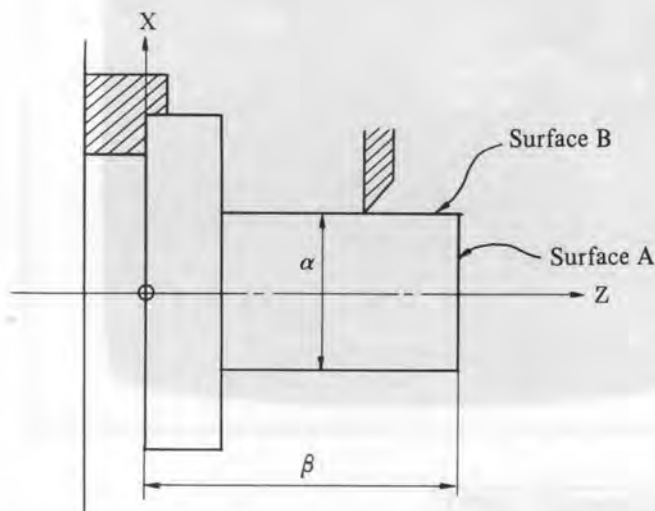
When the tool offset amount (X, Z) has already been set on the standard tool as in the figure above and the tool offset function is effective and the direct measured value input for the work coordinate system shift is performed, a work coordinate system is set in which the coordinate values of the standard point are $X = 0.0$ and $Z = 0.0$ when the standard point is positioned at the zero point.

(Note 2) The X axis data should be measured with a diameter value.

5.12.6 Direct input of tool offset values

To set the difference between the tool reference point used in programming (the nose of a basic tool, turret center, etc.) and the tool tip position of a tool actually used as an offset value, use following convenient method.

Suppose that a work coordinate system has been set.



- Cut surface A in manual mode with an actual tool.
- Press the tool measurement button on the machine operator's panel and retract the tool. Stop the spindle.
- Measure distance β from the zero point in the work coordinate system to surface A.

- (d) Press the **OFFSET** button.
- (e) Key in **N** and the **offset number + No. 100** and press the **INPUT** button.
The page with the desired offset-number will be selected and the cursor indicates that offset number.
The keyed-in N will flicker.



- (f) Key in **Z** and the **measured value β** and press the **INPUT** button. The calculated Z offset value will be entered in the desired offset number.
- (g) Cut surface B in manual mode.
- (h) Press the tool measurement button and retract the tool. Stop the spindle.
- (i) Measure the diameter α of surface B.
- (j) Key in **X** and the **measured value α** and press the **INPUT** button.
The calculated X offset value will be entered in the desired offset number.
Repeat the above sequence the same number of times as the number of required tools.

(Note 1) Measure the X-axis value with a diameter value.

5.12.7 Input an offset value to the counter (option)

Pressing the INPUT button when an address (X, Z, U or W) but not a number has been keyed in to set an offset value, sets as an offset value the value of relative coordinate value U (when the address is X or U) or W (when the address is Z or W).

This is used as follows:

- (a) Press the **OFFSET** button to select the offset page.
- (b) Press the **CURSOR** button to select the page with the number of an offset value to be set.
Alternatively, key in **N** and the **offset number** and press the **INPUT** button.
- (c) Move a standard tool to the standard point manually.
- (d) Reset the relative coordinate values U and W to 0.

U **ORIGIN** **W** **ORIGIN**



- (e) Move the tool whose offset value is to be set to the standard point. The offset value will be displayed in relative position.
- (f) sets the value in relative position display as the offset value.

5.13 Program Display (Function button:)

(1) In EDIT mode

Pressing the button displays a page containing a selected word in a currently selected program.



See Section 5.14 for which program to display. Pressing a PAGE button displays program contents in sequence. Pressing the button displays the contents in the normal direction; the button, in the reverse direction.

(Note 1) Switching from another mode to the EDIT mode and pressing the button displays the contents of a program starting with a block currently being executed or which has been executed. When returning to the head of a program (see Section 5.29.4), however, the contents are displayed from the head of the program.

(2) **Memory operation**

Pressing the **PRGRM** button displays the screen containing the block currently being executed.



The cursor indicates (in MEMORY operation)

- (a) When flickering,
the block to be executed next.
- (b) When not
the block currently being executed or a block which has been executed.

(Note 1) Strictly, the cursor flickers in neither automatic nor feed hold operation when the buffer storage is empty, indicating that a block should be read in the buffer storage next for execution.

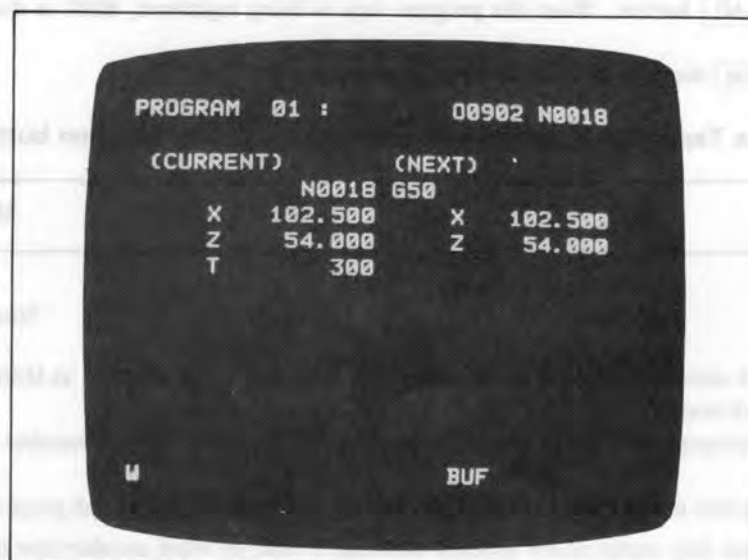
(Note 2) When a subprogram is being executed, the same as above is also displayed in TAPE mode.

(Note 3) When a program is started in MEMORY mode after the cursor is moved by pressing the PAGE or CURSOR button in EDIT mode, the block at the cursor position in EDIT mode is read into the buffer storage next.

(3) **In modes other than the EDIT mode**

Pressing the **PRGRM** button displays the contents of the block currently being executed or which has been executed on the left, and the contents of the block to be executed next on the right side.

(Note 1) As in G28, G29, G90, G92, G94, and movement at a corner in tool nose R compensation, when one command block creates movements for several blocks, the same contents are sometimes displayed on both right and left sides.



5.14 Program Number Search (Function button **PRGRM**)

One of the programs in memory can be searched for.

O1001	O3054	O1972
-------	-------	-------

⇒ Search for a program number.

(1) Method 1

- Select a mode (EDIT or MEMORY).
- Press the **PRGRM** button.
- Key in **O** and the program number to be searched for and press the **CURSOR** button. At the end of the search, the page representing the first of the program will be displayed.

(2) Method 2

- Select a MEMORY mode.
- Press the **PRGRM** button.
- Key in **O** and press the **CAN** and **CURSOR** buttons in this order.
The next registered program will be displayed.

(3) Method 3

- Select an EDIT mode.
- Press the **PRGRM** button.
- Key in **O** and **CURSOR** buttons in this order.

The next program registered will be displayed.

If you keep pressing the **CURSOR** button, numbers of all registered programs are displayed in succession.

This method is used to examine registered program numbers.

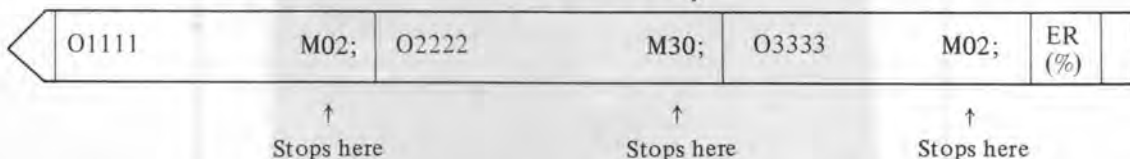
(Note 1) When all registered program numbers have been displayed, the first one is displayed again.

(Note 2) Searching for a program number clears the contents of buffer storage.

5.15 Registering a Program Tape in Memory (, any function button)

- Select a mode (EDIT or MEMORY).
- Load the program tape into the tape reader.
- When the tape has no program number or when the program number is to be changed, enter the program number. (When the tape has a program number and the program number is not changed, this is not required.)
Key in **O** and the program number.
- Press the **READ** button. When the program tape is being registered, EDIT is displayed at the bottom of the screen.
- Pressing **PRGRM** displays the first of the registered program.

5.16 Registering One Tape with Programs into Memory (, any function buttons.)

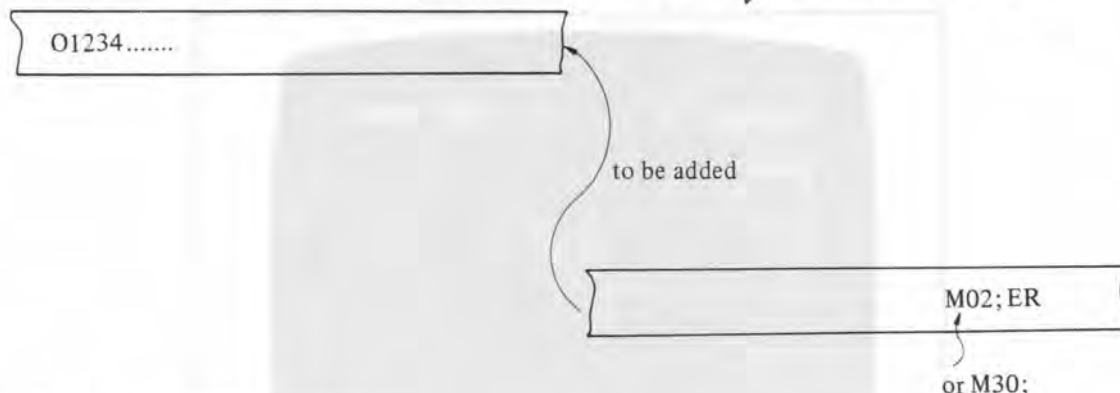


- Operation is the same as described in Section 5.15. The tape stops at M02;, at M30;. Press the **READ** button after each stop.
- To change the program number or set a new one, key in **O** and a program number and press the **READ** button.
- To register programs in succession as far as ER, key in **O** and **-9999** and press the **READ** button.

(Note 1) The Label Skip status is not entered until ER is read, so when another tape is to be loaded into the tape reader after reading up to M02;, M30; or M99;, press the **RESET** button.

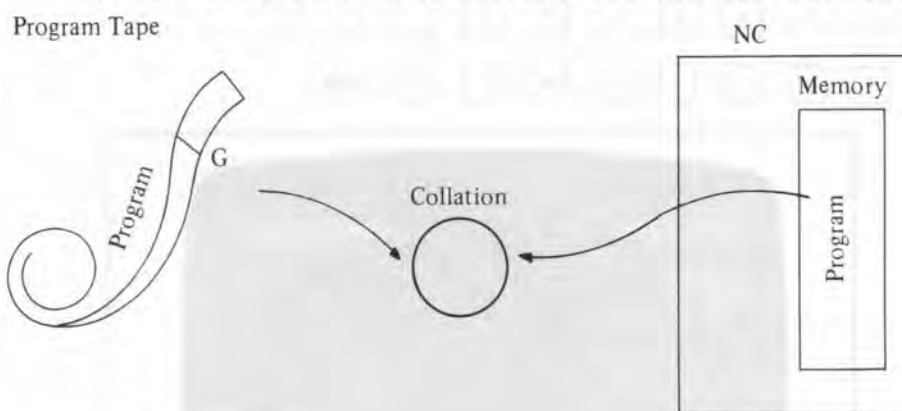
(Note 2) When reading a tape via the RS232C interface, the tape does not stop right after M02, M30 or M99, but it stops after a while. In this case, since the NC loads the information beyond the code M02, M30 or M99, push **READ** button subsequently to read the next program on a tape. However, if the NC is reset or the mode is changed before pushing **READ** button, the information beyond the code M02, M30 or M99 will be cancelled.

5.17 Adding a Program after an Already Registered Program (💡, any function button.)



- Select a mode (EDIT or MEMORY).
- Press the **PRGRM** button.
- Set the program tape into the tape reader.
- Search for the program which has already been registered.
Key in **O** and the program number and press the **CURSOR** button. (This is not required when the program has been found.)
- Press the **CAN** and **READ** buttons in this order.

5.18 Collating Programs in Memory and Those on Tape (Option) (Function button: **PRGRM**)



- Select a mode (EDIT or MEMORY).
- Press the **PRGRM** button.
- Load the program tape to be collated into the tape reader.
- Press the **/** key and the **READ** button in this order (a slash cannot be keyed in without pressing the **PRGRM** button).
- When the tape contains several programs, collation is performed until ER (%) is encountered.

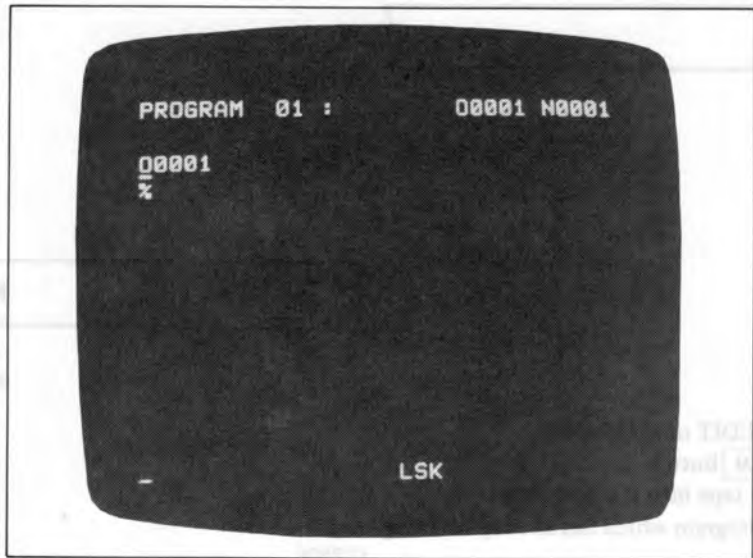
(Note 1) Pressing the **EOB** and **READ** buttons in this order collates the currently indicated word by the cursor and subsequent ones in memory and the tape. To collate a tape with no program number and a program registered with a program number entered from the MDI, return to the head of memory, press the **CURSOR** button once to display the word following the program number, then press the **EOB** and **READ** buttons.

(Note 2) A mismatch during collation generates an alarm. Pressing the **PRGRM** button displays the program and the cursor indicates the location of the mismatch. Pressing the **CAN** button releases the alarm, starting collation with the next character.

5.19 Inputting Programs by Keys

A program written on a process sheet can be directly registered into memory using MDI keys.

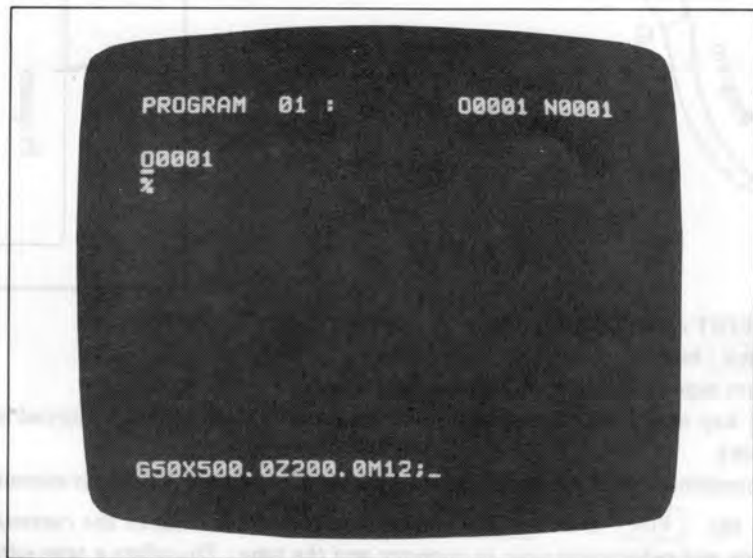
- (a) Select EDIT mode.
- (b) Press the **PRGRM** button. The current program is displayed.
- (c) Enter the program number of the program to be registered.
Pushing **O**, **program number** and **INSRT** displays a new screen.



- (d) Key in one block according to the process sheet.

(Example) To input G50X500.0Z200.0M12;

G	5	0	X	5	0	0	.	0	Z
2	0	0	.	0	M	1	2	EOB	



- (e) When a keying error occurs, press the **CAN** key. The character keyed in last will disappear. Keep pressing the **CAN** key to erase characters one by one from the last. A block of more than 32 characters is too long. Separate the block at an appropriate breakpoint.

- (f) When the program has been keyed in correctly, press the **INSRT** button.



- (g) Enter blocks one by one in this way.
 (h) To correct an entered block, proceed as directed for program editing in Section 5.29.
 (i) For restart, move the cursor to the word keyed in last for continuous input. This operation is exactly the same as the inserting operation.
 (j) When the entire program has been input, the operation is complete. To return to the beginning, press the **RESET** button.

5.20 Deleting a Program (, function button: **PRGRM**)

To delete a program registered in memory.

- (a) Set the mode selector switch to EDIT.
 (b) Press the **PRGRM** button.
 (c) Key in **O** and the program number and press the **DELET** button. The program with the keyed in number will be deleted.

5.21 Deleting All Programs (, function button: **PRGRM**)

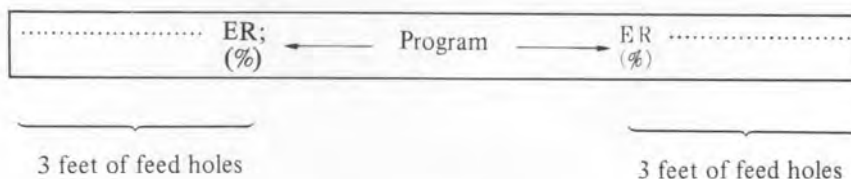
To delete all programs registered in memory.

- (a) Set the mode selector switch to EDIT.
 (b) Press the **PRGRM** button.
 (c) Key in **O**, **-**, **9**, **9**, **9**, and **9** and press the **DELET** button.

5.22 Punching a Program (Option) (Any function button)

To punch out a program from memory.

- (a) Make the tape punch unit ready.
 (b) Press the **SET** button to check that the desired code is set.
 (c) Set the mode selector switch to EDIT.
 (d) Press the **PRGRM** button (this is not always required).
 (e) Key in **O** and the program number and press the **PUNCH** button. The program with the keyed in number will be punched out.



(Note 1) The space code for the TV check is automatically punched.

(Note 2) Two CRs are punched after LF in ISO code.

.....LFCRCR

(Note 3) To stop punching feed holes midway, press the **CAN** button.

(Note 4) Pressing the **RESET** button stops punching.

(Note 5) When a punch unit is connected or disconnected to the NC, both NC and punch unit power must be turned off.

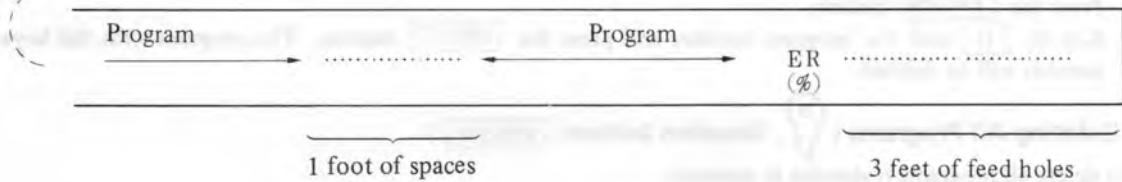
5.23 Punching All Programs (Option) (Any function button)

To punch out all programs from memory.

- Make the tape punch unit ready.
- Press the **SET** button to check that the desired code is set.
- Set the mode selector switch to EDIT.
- Press the **PRGRM** button (this is not always required).
- Key in **O**, **-**, **9**, **9**, **9**, and **9** and press the **PUNCH** button.

3 feet of feed holes

1 foot of spaces



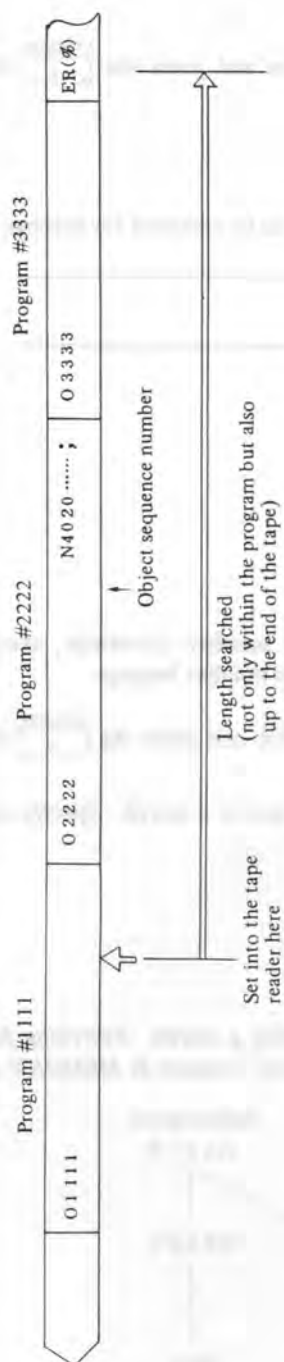
(Note 1) The order of punched programs is irregular.

(Note 2) When a punch unit is connected or disconnected to the NC, both NC and punch unit power must be turned off.

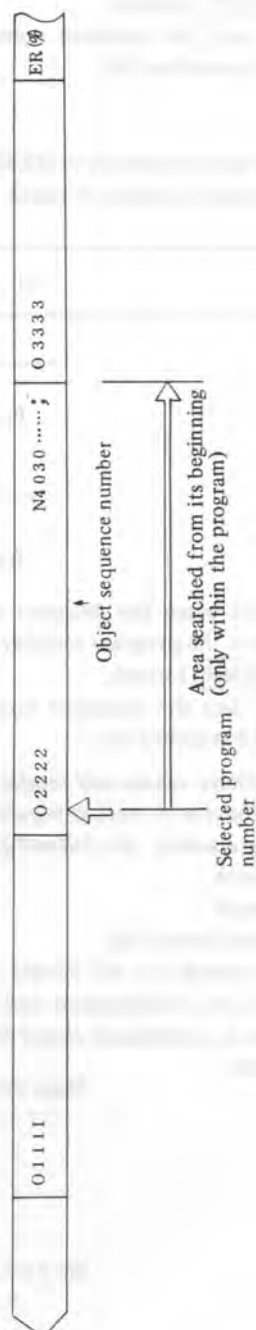
5.24 Sequence Number Search (Function button: PRGRM)

Sequence Number Search finds a sequence number halfway in a program usually used to start or restart the program at the block of that sequence number. A block or blocks skipped during the search does not influence the NC; the coordinate values, M, S, T and G codes, etc. in skipped blocks do not change the coordinate values and modal values of the NC. When the user macro option is provided, sequence numbers are not displayed during search. So specify necessary M, S, T and G codes, coordinate system setting, etc., in the first block at which the program is to be started or restarted with the Sequence Number Search. The block searched for by sequence number search is usually at a breakpoint in processing. To find a block halfway in restart processing, examine the condition of the machine tool and NC and specify M, S, T and G codes, coordinate system setting, etc. from the MDI & DPL panel as required.

Sequence number search in tape operation

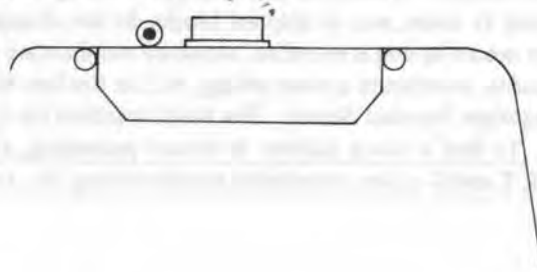


Sequence number search in memory operation



(1) Tape

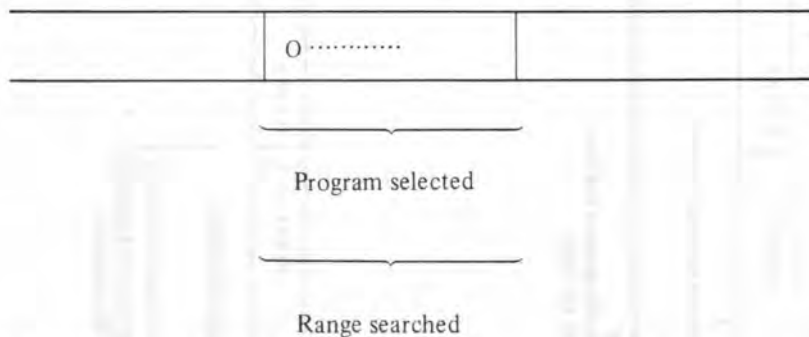
- Set the mode selector switch to TAPE.
- Load the program tape into the tape reader.



- Press the **PRGRM** button.
- Key in **N** and the sequence number to be searched for and press the **CURSOR** button. The sequence number will be searched for.

(2) Memory

- Set the mode selector switch to MEMORY.
- Select the program number to which the sequence number to be searched for belongs.



Proceed to (c) when the program contains the sequence number; Otherwise, execute Program Number Search to select the program number to which the sequence number belongs.

- Press the **PRGRM** button.
- Key in **N** and the sequence number to be searched for and press the **CURSOR** button. The sequence number will be searched for.

(Note 1) Coordinate values and modal data are not altered during a search. Specify this data from the MDI after the search ends as required.

(Note 2) During a search, the following items are checked.

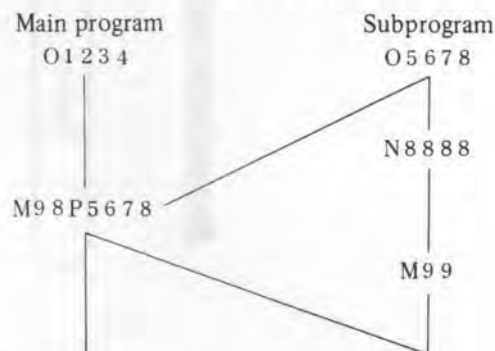
TH check

TV check

Optional Block Skip

Alarm check (03, 04, 05 and 10)

(Note 3) M98Pxxxx (subprogram call) is not executed during a search. Searching for a sequence number within a subprogram called from a currently selected program in MEMORY mode generates alarm No. 060.



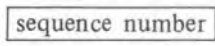
In the above example, searching for N8888 generates an alarm.

5.25 Program Restart

When the tool has been damaged or when machining is to be restarted after a rest, this function restarts machining from the specified block by specifying its sequence number.

In addition to this, this function can be used for high speed tape checking.

(1) The tool has been damaged (Ptype)

- (a) Press the feed hold button, retract the tool, and replace the tool with a new one. Alter the offset values, when required.
- (b) Set the program restart button on the machine operator's panel to ON.
- (c) Press the **PRGRM** button to display the current program.
- (d) Return the program to its beginning.
 - (i) In memory operation, press the **CURSOR**  button in memory mode.
 - (ii) In tape operation, load the tape from the beginning on the tape reader and select the tape mode.
- (e) Pressing **P**,  and **CURSOR**  searches for the block to be restarted from. When the same sequence number appears many times (for example, when a sequence number is searched for in a subprogram that is called several times), specify the number of times the block with that sequence number appears in the high-order four digits and the sequence number in the low-order four digits.

P 1 2 3 4 0 1 2 3 **CURSOR** 
 
 Number Sequence number
 of times

When the number of times is 1, the high-order four digits may be omitted. The leading zero may also be omitted, except for the sequence number when the number of times is specified.

- (f) When completion of the search, the program restart screen is displayed on the CRT



DESTINATION indicates the position from which machining is restarted.

DISTANCE TO GO indicates the distance between the current tool position and the position from which machining is restarted.

M shows the last 35 M codes specified.

T shows the last two T codes specified.

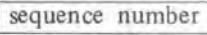
S shows the last S code specified.

The earliest specified code is displayed at the beginning. A program restart command or a cycle start command in the reset status clears each code.

- (g) Set the program restart switch to OFF.
- (h) Look at the screen and output M, S, and T codes to be output, if required, from the MDI panel in MDI mode. In this case, the M, S, or T code issued is not displayed on the PROGRAM RESTART page.

- (i) Return to the memory mode for memory operation; to the tape mode, for tape operation. Check if the distance in DISTANCE TO GO is correct, and if the tool will collide with the workpiece when it moves to the position for machining restart. After moving the tool manually to the position without a collision, if required, press the cycle start button.
The tool will move to this position for machining restart under dry run in Z and X in this order and restart machining.

(2) Machining is restarted after (Q type):

- (i) The power was turned off.
 - (ii) The emergency stop button was pressed.
 - (iii) The machine was stopped instantaneously by the stored stroke limit alarm.
 - (iv) The coordinate system was changed after the last automatic operation.
- For example,
- a) G50 was specified from the MDI panel.
 - b) The coordinate system was shifted.
 - c) Automatic Coordinate System Setting by reference point return was performed.
 - d) The ORIGIN button was pressed.
- (a) Perform required action such as reference point return before machining restart after the power was turned on or emergency stop or stroke limit alarm were released (see Notes).
 - (b) Move the tool manually to the machining start point of the program and set modal information and coordinates to the same status as when machining restarts.
 - (c) Set or change offset values if required.
 - (d) Set the program restart button on the machine operator's panel to ON.
 - (e) Press the **PRGRM** button to display the program. When not desired, search for the desired program.
 - (f) Return the program to its beginning.
- (i) In memory operation, press the  button in memory mode.
 - (ii) In tape operation, load the tape from its beginning on the tape reader and select the tape mode.
- (g)    searches for the sequence number of the block to be restarted from. When the same sequence number appears many times during a search, specify the number of times that sequence number appears in the high-order four digits and the sequence number in the low-order four digits.
 - (h) Upon completion of the search, the program restart screen is displayed on the CRT.
 - (i) Set the program restart switch to OFF.
 - (j) Look at the screen and output M, S, and T codes to be output, if present, from the MDI panel in MDI mode. In this case, the M, S or T code issued is not displayed on the PROGRAM RESTART page.
 - (k) Check if the tool will collide with the workpiece when the tool moves to the position for machining restart. If it will collide, move the tool manually to the position without a collision.
 - (l) Check if the distance in DISTANCE TO GO is adequate.
 - (m) In memory operation, return to the memory mode; in tape operation, to the tape mode. Press the cycle start button. The tool will move to the position for machining restart under dry run in Z and X in this order and restart machining.

(Note 1)    does not execute program restart
When

- (i) No cycle operation is executed after the power was turned on.
- (ii) No cycle operation is executed after the emergency stop or stroke limit alarm (instantaneous stop) was released.
- (iii) No cycle operation is executed after the coordinate system was set or shifted.

Of the above conditions,

If the P type program restart is commanded under condition (i) or (ii), alarm 97 will occur.

Also if it is specified after resetting alarm 94, 96 or 97, alarm 97 will occur.

If it is commanded under the condition that no cycle operation is yet executed after the coordinate system was set or shifted, alarm 94 or 96 will occur respectively.

Machining can be restarted with P type from blocks which follow the block in which the coordinate system was last set before machining was interrupted.

(Note 2) In both P type and Q type, when the tool moves to the position for machining restart along one axis at a time, Single Block Stop is possible after completion of motion along one axis. But MDI operation can not be inserted. Manual operation can be inserted, but no further movement is made on the axis along which the tool has returned.

- (Note 3) During a search, when the input signals, offset values, and other conditions are not the same as before, the tool cannot return to the same machining start point as before. Even if the single block switch is turned on or if the memory mode is changed to tape mode or tape mode is changed to memory mode; the search is continued.
- (Note 4) When a feed hold is executed during a search or when a reset is performed during or after a search, retry the program restart operation from the beginning. However, the reset in the MDI mode after the search ends depends on parameter (006: CLER).
- (Note 5) When the program restart switch is ON, cycle start is ignored.
- (Note 6) Perform manual operation with Manual Absolute ON at all times, before or after machining. However, when a Program Restart is specified without resetting after manual operation or when a manual operation is performed along an axis which has not yet been returned to the machining restart position, the motion concerned is regarded as that under Manual Absolute ON regardless of whether the Manual Absolute switch is ON or OFF.
- (Note 7) In principle the tool cannot be returned to the correct position when:
- (i) Manual motion was performed with Manual Absolute OFF.
 - (ii) Motion was performed with the Machine Lock ON.
 - (iii) An external mirror image was used.
 - (iv) No coordinate system was set at the beginning of incremental programming.
 - (v) Mirror Image was programmed.
 - (vi) Manual operation was inserted while returning along an axis.
 - (vii) When the machine lock was released after program restart was executed in machine lock status.
 - (viii) Program restart was specified in a block between a skip cutting block and a block with an absolute command.
 - (xi) Program restart was specified in a block which belongs to a multiple repetitive cycle.
 - (x) A coordinate system was created or shifted after a search.
- In (iii), however, P-type return is possible for blocks following a block in which ON/OFF was switched last. In this case, keep Mirror Image in the same status as when interrupted. In (v), P-type return is possible. In any case, note that no alarm is generated.
- (Note 8) When the specified block includes only M98, M99, a macro call command (G65, G66, G67) or separate block of macro statement, or when no specified block is searched for, alarm No. 60 is generated.
- (Note 9) When program restart is specified with no reference point return after the power was turned on and G28 is detected, alarm 98 occurs.
- (Note 10) When a move command is executed by an MDI operation before axis movement is performed after a search has ended, alarm 99 occurs.
- (Note 11) "RSTR" flickers at the bottom of the CRT screen until return is made along the final axis (X) after program restart is executed.

5.26 Sequence Number Comparison and Stop

This function is provided to stop processing after executing commands up to the preset sequence number.

- (a) Select the MDI mode.
- (b) Depress **SET** button to obtain the setting page.
(Set the cursor to setting number 180 by depressing **CURSOR**  or **CURSOR**  key.
The cursor cannot move using address N.)
- (c) Input commands in the order of **P**, sequence number to which check is to be stopped, and **INPUT**.
- (d) Select either TAPE or MEMORY mode.

Set the machine tool to be ready for automatic operation.

- (e) Depress the cycle start button.

By step (c), the machine tool stops operating after executing data in the block having the preset sequence number.

The preset sequence number is cleared simultaneously when the machine tool stops operating.

For another comparison stop, if desired, repeat the above procedure, starting with (a).

(Note 1) Sequence number 0 cannot be used as the sequence number for comparison stop.

(Note 2) The preset sequence number is cleared by resetting.

5.27 Entering Offset Values from Tape (option) (Any function button)

(1) Format

G10 P___ X___ Z___ R___ Q___; or

G10 P___ U___ W___ R___ Q___;

Where P: Offset number

X: X-axis offset value (absolute)

Z: Z-axis offset value (absolute)

U: X-axis offset value (incremental)

W: Z-axis offset value (incremental)

R: Tool nose offset value (absolute)

Q: Imaginary tool nose number

G10 is required for each block. The tape ends with % (ISO) or ER (EIA).

; G10 P- ; G10 P- ; ; %

(2) Operation

- Load a tape in the above-format into the tape reader.
- Set the mode selector switch to TAPE.
- Press the **START** button (depending on the machine tool start button on the machine operator's panel must be pressed.)

Note: The absolute input/incremental input switch is set by addresses X, Z, U, and W and is not concerned with the presence or absence of the Incremental Offset option.

5.28 Punching Offset Values onto Tape (option) (Function button: **OFFSET**)

- Make the tape punch unit ready.
- Set the mode selector switch to EDIT.
- Press the **OFFSET** button.
- Key in **P**, **-**, **9**, **9**, **9**, and **9** and press the **PUNCH** button. The offset values of all the offset numbers will be punched in the same format as input.

(Note) When a punch unit is connected or disconnected to the NC, both NC and punch unit power must be turned off.

5.29 Parameter Display (Function button: **PARAM**)

Pressing the **PARAM** button displays parameters. Parameters cover several pages in display. Press a **PAGE** button to display a particular parameter. See Appendix 6 for the meanings of parameters.

5.30 Program Editing (Function button: **PRGRM**)

This function is used to modify a program in memory.

- Set the mode selector switch to EDIT.
- Press the **PRGRM** button.
- Select a program. When it has been selected, proceed to (4); otherwise, perform a Program Number Search.
- Search for the word to be modified. A scan or a word search can be used.
- Modify, insert, or delete the word.

Note 1: Words and editing unit.

A word includes an address and the number following it. But in user macros, the concept of a word is not clear. Consequently, "editing unit" is used. This is the object of modification and deletion in one operation. One scan causes the cursor to indicate the beginning of the editing unit. In insertion, data is inserted after the editing unit.

Definition of editing unit

(1) From one address to the next address exclusive.

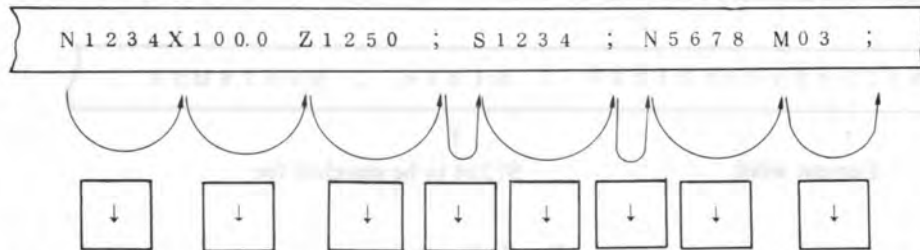
(2) The address is an alphabetic character, IF, WHILE, GOTO, END, DO, =, or; (EOB).

According to this definition, a word is also one editing unit. The word referred to in the following explanation on editing should strictly be termed the editing unit.

(Note 2) It is not allowable to continue a program again after changing, inserting, or deleting data to the program while temporarily suspending the machining by single block stop, feed hold, etc. during the execution of the program, otherwise the program is not always executed correctly as specified by the program data being displayed on CRT after the continuance of machining.
If it is desired to change memory data by part program editing, change them under the reset condition, or perform a reset operation after editing without fail, before executing the program.

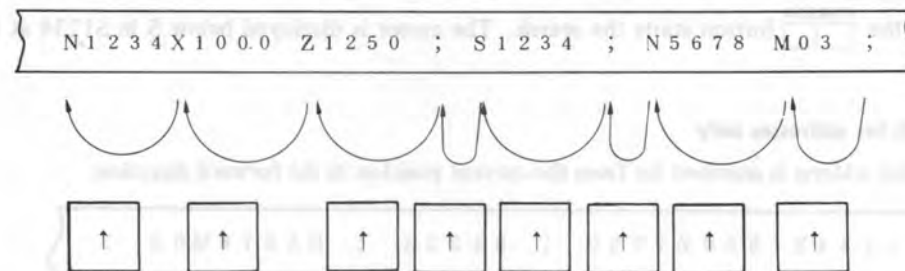
5.30.1 Word scan

(1) Press the  button



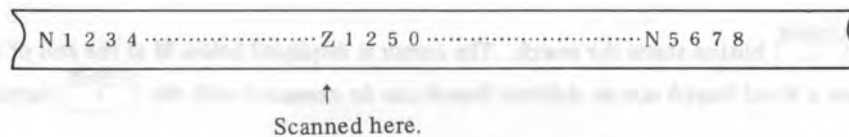
The cursor moves forward word by word on the screen; the cursor is displayed below the address character of a selected word.

(2) Press the  button



The cursor moves backward word by word on the screen; the cursor is displayed below the address character of a selected word.

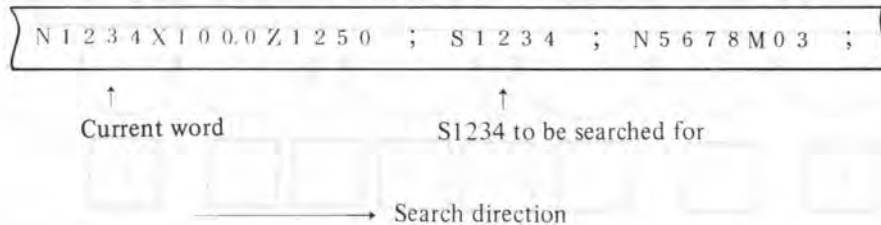
(Example)



- (3) Keep pressing the  or the  button to make a continuous search.
- (4) Press the  button to display the following page and search from the first of that page.
- (5) Press the  button to display the preceding page and search from the first of front page.
- (6) Keep pressing the  or  button to display one page after another.

5.30.2 Word search

A specified word is searched for from the current position in the forward direction.



- (1) Key in , , , , and .

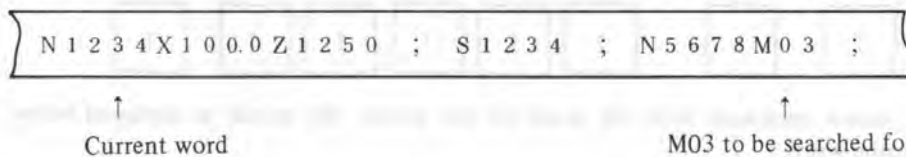
Note 1: Keying in only S123 does not execute a search for S1234.

2: Keying in S9 does not execute a search for S009. Key in 009 to search for S009.

- (2) Pressing the  button starts the search. The cursor is displayed below S in S1234 at the end of the search.

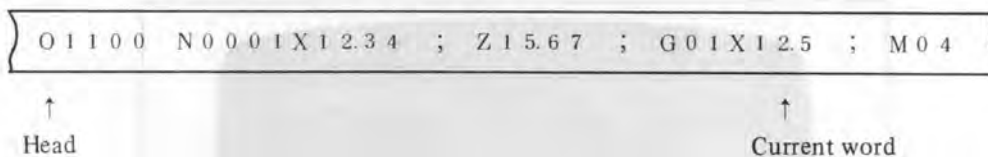
5.30.3 Search for addresses only

A specified address is searched for from the current position in the forward direction.



- (1) Key in .
 - (2) Pressing the  button starts the search. The cursor is displayed below M at the end of the search.
- (Note 1) Neither a Word Search nor an Address Search can be executed with the  button.*

5.30.4 Returning the cursor to the Head of a Program



- (1) **Method 1**
Press the  button in EDIT mode. The program will be displayed from its head.
- (2) **Method 2**
Execute a Program Number Search.
- (3) **Method 3**
 - (a) Set the mode selector switch to MEMORY.
 - (b) Press the  button.
 - (c) Press the  button. For part program editing, return to the EDIT mode.

5.30.5 Inserting a Word



T105 to be inserted

N 1 2 3 4 X 1 0 0 . 0 Z 1 2 5 0 ; S 1 2 3 4 ; N 5 6 7 8 M 3 0 ;

↑
To be searched for

- (1) Search for or scan the word immediately before the insertion location
 - (a) Scan
See Section 5.30.1.
 - (b) Word Search
See Section 5.30.2. When Z1250 is before the current position, return the cursor to the head of the program first.
- (2) Key in **T** , **1** , **0** , and **5** and press the **INSRT** button.

```

PROGRAM 01 :      04444 N1234

04444 ;
N1234 X100.0 Z1250 ;
S1234 ;
N5678 M03 ;
%

T105_
  
```

Before

```

PROGRAM 01 :      04444 N1234

04444 ;
N1234 X100.0 Z1250 I105 ;
S1234 ;
N5678 M03 ;
%

-                               LSK      INC
  
```

After

- (Note 1) When a number but no address is inserted, the inserted number is added to the current word (editing unit). In the above example, inserting 2.5 when the cursor is below Z in Z1250, produces Z12502.5.
- (Note 2) Similarly, a number can be added after all addresses such as EOB, IF, etc. Inserting 23 when the cursor is below ;, produces ;23. But this is meaningless in programming.

5.30.6 Changing a word (💡)

```
N 1 2 3 4 X 1 0 0 . 0   Z 1 2 5 0 T 1 0 5   ;   S 1 2 3 4   ;
```

↑
To be changed to M15

- (1) Search for/scan the word to be changed.
- (2) Key in , , and and press the button.

```
N 1 2 3 4 X 1 0 0 . 0   Z 1 2 5 0 M 1 5   ;   S 1 2 3 4   ;
```

Modified program

5.30.7 Insertion or correction of words, blocks, and character strings

More than one word, block or character string can be inserted (up to 32 characters). In the previous example (5.30.5), when

T105M20

is to be inserted,

key in T105M20 and press the INSRT button.

```

PROGRAM 01 :           04444 N1234

04444 ;
N1234 X100.0 Z1250 ;
S1234 ;
N5678 M03 ;
%

T105M20_

```

Before inserted



After inserted

Similarly, a word indicated by the cursor can be changed to more than one word, block or character string.

(Note 1) When the cursor is below Z in Z1250, inserting 2.5M20 produces Z12502.5M20.

(Note 2) When the cursor is below T in Z1250T105, replacing 2.5M20 produces Z12502.5M20.

5.30.8 Deleting a word (💡)

N 1 2 3 4 X 1 0 0 . 0 Z 1 2 5 0 T 1 0 5 ; S 1 2 3 4 ;

↑
Z1250 to be deleted

- (1) Search for/scan the word to be deleted.
- (2) Press the **DELET** button.

N 1 2 3 4 X 1 0 0 . 0 T 1 0 5 ; S 1 2 3 4 ;

Program after deletion

5.30.9 Deleting up to an EOB (💡)

Current word (Search or scan)



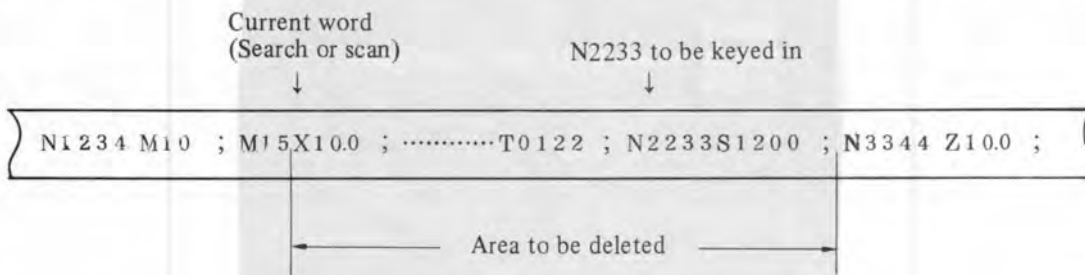
N 1 2 3 4 X 1 0 0 . 0 Z 1 2 5 0 T 1 0 5 M 1 3 ; S 1 2 3 4 ;

← Area to be deleted →

Pressing the **EOB** and the **DELET** button deletes up to an EOB and causes the cursor to move to below the address character of the next word.

5.30.10 Deleting blocks (💡)

The area from the currently displayed word to the block with a specified sequence number is deleted.



- (1) Key in the sequence number of the last block to be deleted.
Key in , , , , and in this example.
- (2) Press the button.

5.30.11 Memory arrangement

Frequent part program editing sometimes causes memory to be used unconconomically so that the specified length tape cannot be accumulated. This requires Memory Arrangement.

- (1) Press the and the buttons in this order. After arrangement, the number of characters that can still be accommodated is displayed at the bottom of the screen.

(Note 1) When there is one program on memory, the memory stores the specified length. When many programs are on memory, some areas in memory are used for their identification.

(Note 2) For speedier part program editing, memory area in excess of characters actually modified or inserted is sometimes consumed. Memory Arrangement eliminates wasteful use of memory.

5.30.12 Displaying program numbers of all registered programs

Arranging the memory contents as discussed in Section 5.30.11 displays the program numbers of all registered programs.



5.30.13 Custom macro editing (💡)

Custom macros can be edited using shift key.

Key operation is done as usual under edit mode with the program protect key released. However, note that the following points differ:

(a) Shift key

When  key is depressed, cursor (key input cursor; at the character position following the last input data) changes from the ordinary  to . If a key with a character at the right lower corner of the key-top is depressed under this state, the character at the right lower corner will be input.

After one character is input, the cursor reverts to the ordinary . The cursor also reverts to the ordinary  when shift key is depressed twice.

< Example >



(b) How to delete, alter and insert program

On editing custom macro programs already input, cursor can move only to the following:

- (i) Address
- (ii) / of optional block skip
- (iii) # at the head of the left side of substitution statement
- (iv) (, =, ;
- (v) Head character of IF, WHILE, GOTO, END, DO

There is a blank space of one character in front of the above on the CRT screen.

Deleting, altering and inserting of program is done in a unit from the first position of cursor to one character before the next position of cursor.

(Example) Position of cursor

```

N001 X-#100 ;
#1 =123 ;
N002 /2 X[12/#3] ;
N003 X-SQRT[#3/3*[#4+1]] ;
N004 X-#2 Z#1 ;
N005 #5 =1+2-#10 ;
IF [#1 NE 0] GOTO 10 ;
WHILE [#2 LE 5] DO 1 ;
#[2000+#2] =#2*10 ;
#2 =#2+1 ;
END 1 ;

```

(Note 1) Cursor does not stop between ().

↑ ↑
Control out/in

(Example)

(#1=100) ;
↑

Cursor does not stop here.

(Note 2) Position of cursor may change according to alteration in program.

(Example) When

X100 Z200 ;

if Z200 is altered to 100, using  key it becomes

X100100 ;

(c) Abbreviation of custom macro words

When altering or inserting custom macro words, the top two characters can be used as abbreviation for the word.

The underlined part of each word below is the abbreviation which can be used as substitute for the whole word:

WHILE, GOTO, END, XOR, AND, SIN, COS, TAN, ATAN, SQRT, ABS, BCD, BIN, FIX, FUP, ROUND.

(Example)

When

WH[TA[#1*AB[#2]] LERO[#3]]

is input as key input data, it is accepted as

WHILE[TAN[#1*ABS[#2]] LE ROUN[ROUND[#3]]].

5.31 Run Hour Display

Automatic operating times are integrated and displayed on the screen in hours, minutes, and seconds (in units of two seconds).

Pressing the  button displays the time as in the photo. When another page appears, press the PAGE button.



(Note 1) Times to be integrated includes those of automatic operation and not those of single block stop, feed hold stop, etc.

(Note 2) Powering off immediately after stopping automatic operation sometimes generates an error of up to six minutes when powering on again.

(Note 3) Run hour can be preset from MDI, if required. See section 5.8. The data number is 57, 58 and 59.

5.32 Menu Switch Function

Switch functions can be turned on and off in NC memory by using CRT instead of switches on the machine tool operator's panel.

By using this function, the number of switches on the machine tool operator's panel can be reduced.

The following signals can be turned on and off on CRT.

- (1) Single block (SBK)
- (2) Machine lock (MLK)
- (3) Dry run (DRN)
- (4) Optional block skip 1 ~ 9 (BDT1 ~ 9)
- (5) Mirror image (MIX, MIZ)
- (6) Display lock (DLK)
- (7) Auxiliary function lock (AFL)
- (8) Manual absolute (ABS)

The signals turned on and off on CRT remain unchanged, even if NC power supply is turned off once after their data have been stored into bubble memory.

These signals are not fully replaced with those on CRT but, they are regarded as being turned on when either machine tool signals or signals being set on CRT are turned on.

Accordingly, these signals can be turned on and off on CRT by deleting optional signals out of the above listed signals from the machine tool operator's panel and opening the circuit for these signal lines.

Setting and display

The CRT status of the above signals can be displayed by the following operation.

Display

- (i) Select chapter 3 in setting.

Depress for setting display, and then, depress twice.

- (ii) Display a necessary page by depressing PAGE button by selecting it from the display on 2 pages.

Page 1 Those other than optional block skip

Page 2 Optional block skip 1 ~ 9

Setting

Observe the following procedure after display procedure.

- (iii) Shift the cursor to the item to be changed.

Set the cursor to the item to be changed by depressing or key.

- (iv) Input 1 for turning on or 0 for turning off after depressing address .

Depress keys in the order of { } (INPUT)

Depress keys in the order of { } (INPUT)



6. POSITION DISPLAY BY POSITION DISPLAY UNIT

The position display unit displays the present position. Position display reset buttons are located on the position display unit with one button for each axis. When one of these buttons is pushed, the position display for the corresponding axis is cleared. This position is referred to as zero point in subsequent representation.

When the coordinate system setting G50 (G92) is specified by the program or MDI, the current coordinate values can be set on the position display depending on the system parameter PPD.

(Note 1) When the DISPLAY LOCK switch is set to ON, position pulses are not applied to the position display unit.

When shifting a coordinate system by manual operation, use this switch so that the shift amount does not affect the display.

(Note 2) For inch input, position display is displayed in inch format.

For metric input, position display is displayed using the metric system.

When the input system is switched from inch to metric system or from metric to inch, the reset buttons should be pushed to make the indication on the position display zero.

If the input system is changed while the data is being displayed, only the decimal point is shifted, the data remains unchanged.

(Note 3) Compensation data on the machine tool such as the backlash compensation amount are not displayed on the position display unit.



7. OPERATION OF FANUC CASSETTE

7.1 General

By adding an input/output interface option, various data can be transferred between NC and FANUC CASSETTE (hereinafter referred to as cassette) via RS232C interface. The following data can be transferred in the same manner as in data transfer to and from paper tape reader/puncher.

- (1) NC command data
- (2) Offset data
- (3) NC parameters (including pitch error compensation data)

7.2 Input/Output Operation

Data are transferred to and from the cassette by just the same manner as in paper tape reader/puncher, in principle, and you may presume that the cassette is attached to the RS232C interface instead of paper tape reader/puncher; provided that the cassette involves a concept as a file, and the following filing operation is added.

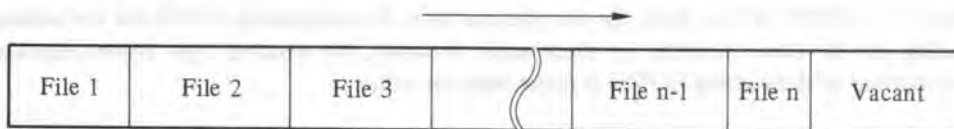
- (1) Heading of file
- (2) Deletion of file

7.3 What Is a File?

The unit of data to be transferred between NC and cassette by each input/output operation (by depressing **READ** or **PUNCH** button) is called "file". Let's consider an NC program output. If O100 **PUNCH** operation was made, a program having program No. 100 composes a file inside the cassette, and if O-9999 **PUNCH** (all program output) operation was made, some programs in NC memory compose a file.

Respective files are numbered with ascending file numbers as 2, 3, 4 . . . in the output order, starting with 1 as the start file No. These files correspond to data in the cassette by these file numbers hereafter.

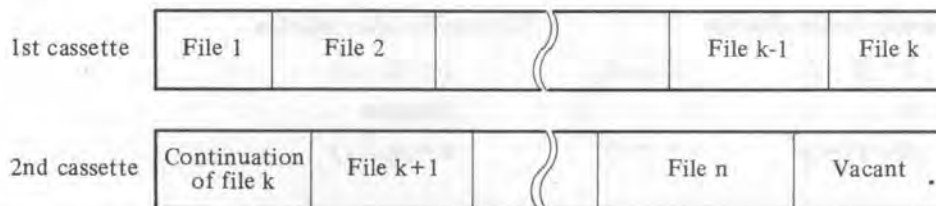
Since it is impossible to visually confirm the correspondence between file numbers and loaded data, it is recommended to enter a file number and its data contents into the "MEMO" column of the cassette at once, each time a data was outputted to the cassette.



MEMO (Example)

File 1	NC parameter
File 2	Offset data
File 3	NC program O0100
⋮	⋮
⋮	⋮
File n-1	NC program O0500
File n	NC program O0600

One file can extend over 2 cassettes as illustrated below.



7.4 Heading of File

When data are transferred to and from the cassette, it is necessary to designate a cassette data to be inputted or outputted. This designation is done by the file number, and this designating operation is called "heading of file".

Heading of a file is executed by the following procedure.

- (1) Select the EDIT mode (or MEMORY mode).
- (2) Select **PRGRM** function button.
- (3) Input **N**, **File No.**, and **INPUT** in this order.

Heading is executed as shown in the following table according to the file numbers in (3).

	File number	Heading contents
1	0 (zero)	The cassette start is headed.
2	1 ~ 9999	A file having file No. 1 to 9999 is headed.
3	-9999	A file next to the file accessed just before is headed.
4	-9998	By this designation, N-9999 is automatically inserted before input/output when data is transferred to and from the cassette. In other words, the next file is headed without fail before data transfer, and data is transferred to and from the file. This designation is modal, and this state is cancelled by designating (1), (2), or (3) or by reset operation.

(Note 1) *N0 and N1 designation brings the same result, if a file exists in the cassette. If no file exists, correct heading is not done by N1 designation because of the absence of the first file, while correct heading is done by N0 designation, since the start is headed, irrespective of whether a file is present or absent in the cassette.*

If a new cassette is employed or if data are outputted from the start after making all existing files invalid, the heading is done by N0 designation.

(Note 2) *No alarm is displayed on the CRT screen, even if the heading is not done correctly because of the absence of files, etc.*

If data transfer operation (described later) is done hereafter, alarm 086 is displayed.

(Note 3) *When continuously heading files, they can be headed every time by designating N1 ~ N9999 or by designating N1 ~ N9999 first to head the first file and then, by designating N-9999 for the subsequent files. Heading can be done correctly by these ways. However, the required time differs between the two. (The method of designating N-9999 is faster than the other.)*

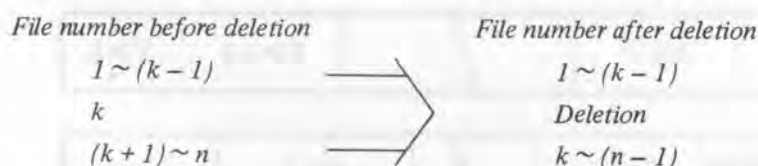
7.5 Deletion of File

A cassette file can be deleted by the following procedure.

- (1) Select the EDIT mode.
- (2) Select **PRGRM** of function buttons.
- (3) Release the lock key ().
- (4) Input **N**, **file number**, and **START** in this order. (Input a file number (1 ~ 9999) of the file to be deleted.)

By this operation, the k-th file inputted in (4) is deleted.

(Note 1) *Be careful since the file number after deletion is added by -1 about old (k + 1) ~ n files to be k ~ (n - 1) files when the k-th file was deleted. (Change the description in the MEMO column described in 7.3)*



(Note 2) *The cassette is provided with a write protect slide. Set this write protect slide upward, and make sure that the red lamp of the adapter lights when the cassette was inserted into the adapter, before strating the deletion.*

No alarm is displayed, even if the operation for deletion is made without releasing the write protect slide. However, the deletion is not executed in this case.

7.6 Data Transfer to and from Cassette

Data can be transferred to and from the cassette in the same manner as in paper tape reader/puncher. However, be careful with the following items.

- (1) Release the write protect slide of the cassette described in 7.5 (Note 2) without fail when data are outputted to the cassette. Alarm 086 is displayed, if data are outputted without releasing it.
- (2) When data are inputted from the cassette, make sure that INPUT DEVICE 2 on page 1 is set to "1" (input from RS232C) when the function button **SET** is depressed.
If INPUT DEVICE 2 is set to "0", data are inputted from paper tape reader.

7.6.1 NC program transfer to and from cassette

(1) NC program output to cassette

An NC program can be outputted to the cassette by just the same operation as in punching to the paper tape puncher.

- (1) Select the EDIT mode.
- (2) Select **PRGRM** function button.
- (3) Input **O**, **program No** and **PUNCH** (1 program output) or **O**, **-9999** and **PUNCH** (all programs output)

in this order.

By this operation, an NC program is outputted as a new file, after existing files in the cassette.

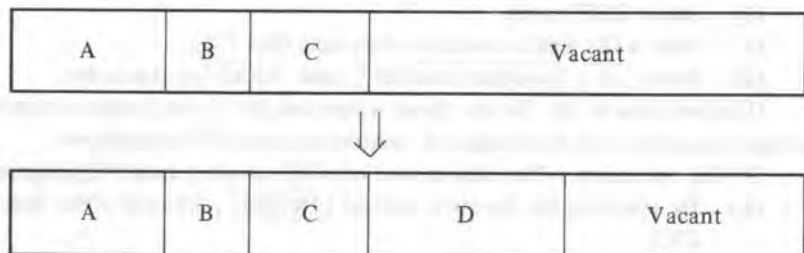
If the output operation to the cassette was made after heading of a file as described in 7.4, an NC program is outputted as a new file to the headed n-th file.

In this case, the previous state is reserved in 1 ~ (n - 1) files. However, old n-th file and its subsequent files are all deleted.

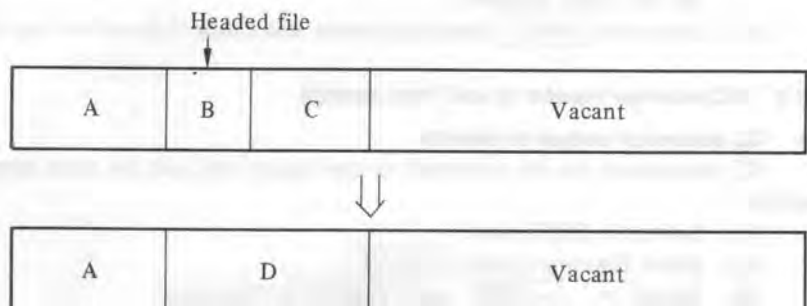
If an NC program is outputted to new cassette or it is outputted to the start of old cassette after making all existing files invalid, perform the above output operation after heading by N0 designation.

(Example) For **D** output;

- (i) When no heading was made;



- (ii) When heading was made;



(Note 1) If alarm 086 was produced during the output operation to a cassette, cassette data are restored to the state before output.

If alarm 086 occurred during output operation after heading the n-th file, cassette data are restored up to 1 ~ (n - 1) files.

(2) NC program input from cassette

An NC program can be inputted from the cassette by just the same operation as in reading from paper tape reader.

- (i) Select EDIT mode (or MEMORY mode)
- (ii) Head a file in which a desired program is loaded (See 7.4).
- (iii) Input **O**, **-9999**, and **READ** (normal) or **O**, **program number**, and **READ** [when to change the order of inputting programs] in this order.

By this operation, an NC program in the headed file is read into NC memory.

(iv) By selecting the function button **PRGRM**, the read program is displayed on CRT.

(Example) For reading files 2 to 4 in cassette into memory;

(Operation example 1)	(Operation example 2)	(Operation example 3)
N2 INPUT	N2 INPUT	N2 INPUT ——— Heading of file 2
O-9999 READ	O-9999 READ	O-9999 READ ——— Reading of file 2
N3 INPUT	N-9999 INPUT	N-9998 INPUT ——— Heading of file 3
O-9999 READ	O-9999 READ	O-9999 READ ——— Reading of file 3
N4 INPUT	N-9999 INPUT	————— Heading of file 4
O-9999 READ	O-9999 READ	O-9999 READ ——— Reading of file 4
Slow	Quick	Quick ——— Heading time

7.6.2 Offset data transfer to and from cassette

(1) Offset data output to cassette

Offset data can be outputted to the cassette by just the same operation as in punching to paper tape puncher.

- (i) Select EDIT mode.
- (ii) Select function button **OFFSET**.
- (iii) Input **P**, **-9999**, and **PUNCH** in this order.

By this operation, offset data are outputted to the cassette. The file, to which offset data are outputted, is determined in the same manner as in NC program output. (See 7.6.1 (1))

(2) Offset data input from cassette

Offset data can be inputted from the cassette, and set to the NC offset memory by the following operation.

- (i) Select EDIT mode.
- (ii) Head a file which contains offset data (See 7.4.).
- (iii) Input **O**, **program number**, and **READ** in this order.

If offset data in the file are those outputted in (1), no program number is loaded at the start of data, and a program number must be designated, otherwise alarm 075 is displayed.

By this operation, offset data is read into NC memory as an NC program.

- (iv) By selecting the function button **PRGRM**, the read offset data (G10P_R_ ; format) is displayed on CRT.
- (v) When this program is executed by selecting the MEMORY mode and cycle start, offset data is set to the NC offset memory.
- (vi) Select the **OFFSET** function button, and check if the offset data has been set correctly or not.

7.6.3 NC parameter transfer to and from cassette

(1) NC parameter output to cassette

NC parameters can be outputted to the cassette by just the same operation as in punching to paper tape puncher.

- (i) Select the EDIT mode.
- (ii) Select function button **PARAM**.
- (iii) Input **P**, **-9999**, and **PUNCH** in this order.

The NC parameter is outputted to the cassette by this operation. The file, to which the NC parameter is to be outputted, is determined in the same manner as in NC program output. (See 7.6.1 (1))

(2) NC parameter input from cassette

NC parameters can be inputted from the cassette by just the same operation as in reading from paper tape reader.

- (i) Head a file which loads the NC parameter. (See 7.4)
- (ii) Open the front door of NC, and select the parameter protect switch on the master printed circuit board to ENABLE.

Alarm 100 is displayed.

- (iii) Depress the EMERGENCY STOP button on the machine operator's panel.
- (iv) Select function button **PARAM**.

(v) Input **P**, **-9999**, and **READ** in this order.

By this operation, the NC parameter is read from the cassette, and set into NC parameter memory. After reading the parameters, alarm 000 is produced usually.

(vi) Reset the parameter protect switch to DISABLE.

(vii) Release the EMERGENCY STOP button on the machine operator's panel.

(Note 1) Don't execute this operation by end user, in principle.

7.7 Cassette Exchange Request

If data transfer to and from the first cassette is terminated when one file extends over 2 cassettes (See 7.3), the red and green lamps of the cassette adapter alternately flicker to inform the operator of the cassette exchange request.

When these lamps flicker, unload the first cassette, and insert the second cassette. After the second cassette has been inserted, data transfer is automatically continued.

This cassette exchange request is issued, if the second cassette is required halfway in the course of the following operation.

- (1) Heading of file
- (2) Deletion of file
- (3) Data transfer between NC and cassette

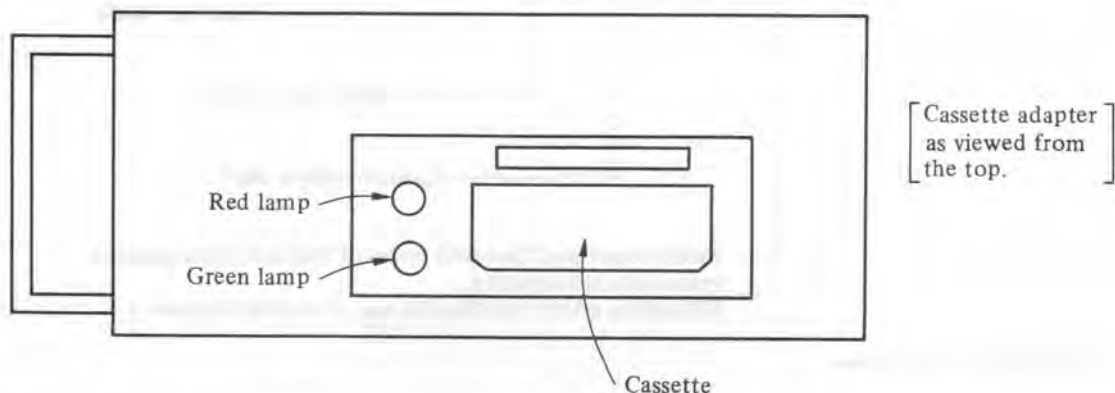
(Note 1) Since all cassette exchange processing is done on the cassette adapter side, no particular operation is needed on the NC. You have only to replace the cassette.

(During this time, data transfer processing is being interrupted on the NC.)

(Note 2) Even if NC is reset during the cassette exchange request, it is not reset at once, but it is reset after cassette exchange. The cassette exchange is an only means of releasing the cassette exchange request, accordingly.

7.8 Cassette Adapter Lamp Conditions

The red and green lamps of the cassette adapter indicate operating conditions.



	Lamp conditions	Operating conditions
1	Red lamp lights	Writable: This lamp lights when cassette is inserted after setting the write protect slide of cassette upward.
2	Green lamp lights	Readable: This lamp lights when cassette is inserted.
3	Red lamp flickers	Write in progress.
4	Green lamp flickers	Read or heading of file in progress.
5	Red and green lamps alternately flicker.	Cassette is not inserted, or cassette exchange request is in progress.
6	Red and green lamps concurrently flickers.	File deletion in progress.

- (Note 1) All alarms regarding data transfer to and from cassette are displayed as alarm 086. Locate a cause of an alarm, and release it by examining the cassette conditions and operation.
- (Note 2) When a cassette adapter is connected to or disconnected from NC, both NC and cassette adapter power must be turned off.

7.9 Parameters and Setting on Data Transfer to and from Cassette

Set parameters and setting as described below when data are transferred to and from the cassette.

(1) Selection of input device when data are inputted by RS232C interface

340

3

(2) Selection of output device when data are outputted by RS232C interface

341

3

Set input/output device No. 3 by setting in (1) and (2).

(3) Input/output parameters of input/output device 3 (cassette adapter)

312

1

0

0

1

1

0

0

1

7

6

5

4

3

2

1

0

Baud rate 4800

Stop bit (2 bits)

Control code is used.

Neither start and final feed codes of data nor inter-program space code is outputted.
(This bit is added for effective use of cassette memory.)

(3) is set by a parameter.

Parameter	Setting	Unit
Serial output level	1	
Serial input level	1	
Serial output speed	3	
Serial input speed	3	
Serial output flow control	1	
Serial input flow control	1	

V. MAINTENANCE

REMARKS

1. DAILY MAINTENANCE

1.1 Tape Reader Cleaning

(a) Tape reader without reels

Item	Cleaning point	Reference drawing	Cleaning period	Cleaning method
1	Surface of read head (light sensing part)	Fig. 1.1 (a) ①	Daily	Clean with gauze or a thin brush with absolute alcohol.
2	Surface of read head (light emitting part)	Fig. 1.1 (a) ②	Daily	
3	Tape retainer	Fig. 1.1 (a) ③	Daily	
4	Tape path	Fig. 1.1 (a) ④	Daily	
5	Capstan roller	Fig. 1.1 (a) ⑤	Weekly	
6	Guide roller	Fig. 1.1 (a) ⑥	Weekly	
7	Pinch roller	Fig. 1.1 (a) ⑦	Weekly	
8	Assembly under tape path plate	Fig. 1.1 (a) ⑧	Monthly	Clean with a cloth or brush.
9	Inside tape reader cover	Fig. 1.1 (b) ⑨	Monthly	

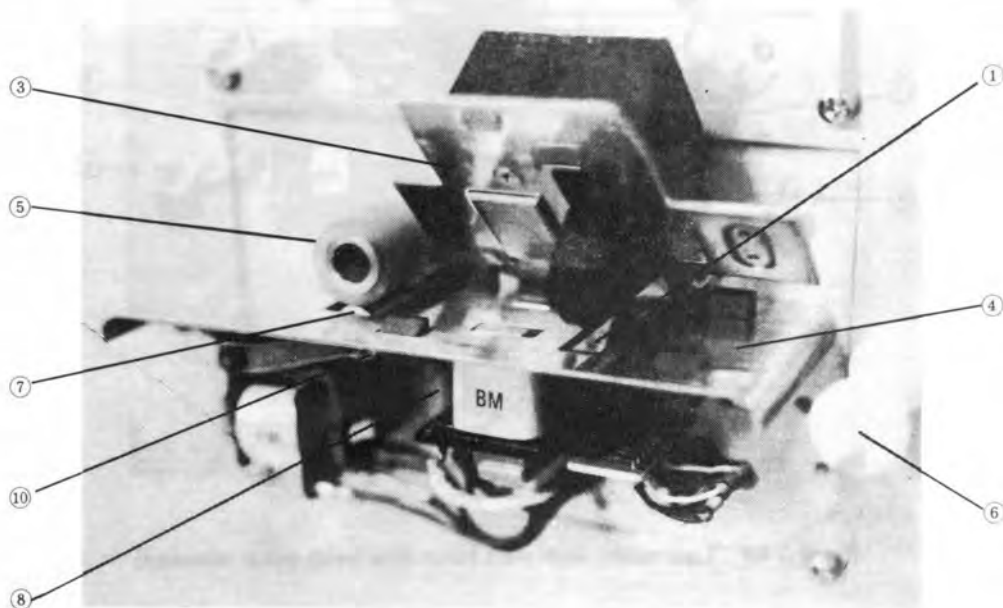


Fig. 1.1 (a) Tape reader without reels front view (with cover removed)

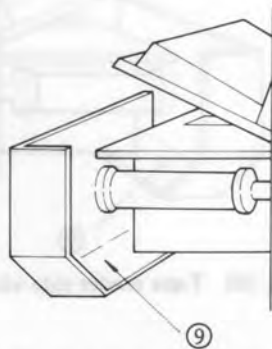


Fig. 1.1 (b) Tape reader side view

(b) Tape reader with reels

Item	Cleaning point	Reference drawing	Cleaning period	Cleaning method
1	Surface of read head (light sensing part)	Fig. 1.1 (c) ①	Daily	Clean with gauze or a thin brush with absolute alcohol.
2	Surface of read head (light emitting part)	Fig. 1.1 (c) ②	Daily	
3	Tape retainer	Fig. 1.1 (c) ③	Daily	
4	Surface of tape path	Fig. 1.1 (c) ④	Daily	
5	Capstan roller	Fig. 1.1 (c) ⑤	Weekly	
6	Guide roller	Fig. 1.1 (c) ⑥	Weekly	
7	Pinch roller	Fig. 1.1 (c) ⑦	Weekly	
8	Assembly under tape path plate.	Fig. 1.1 (c) ⑧	Monthly	Clean with a cloth or a brush.
9	Inside tape reader cover	Fig. 1.1 (d) ⑨	Monthly	

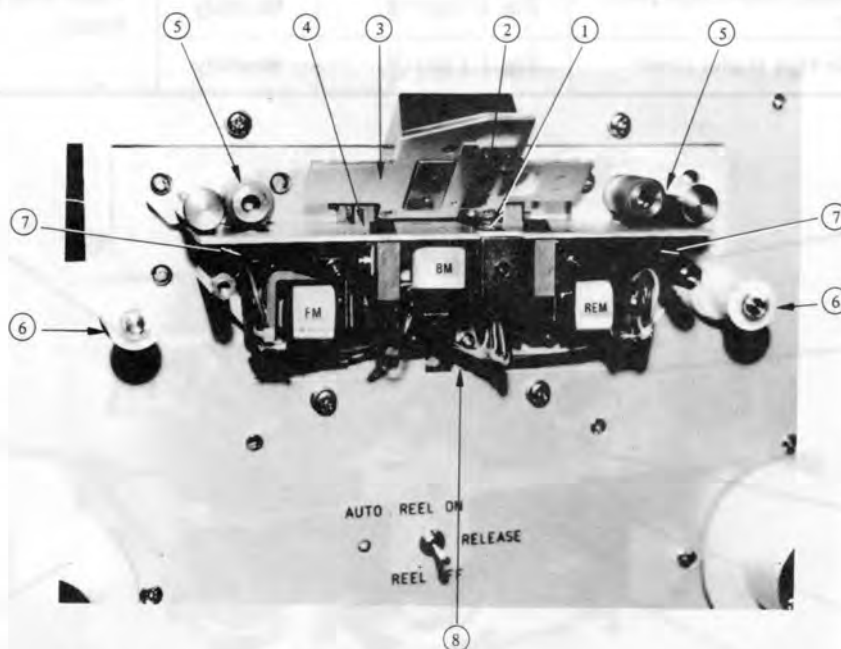


Fig. 1.1 (c) Tape reader with reels front view (with cover removed)

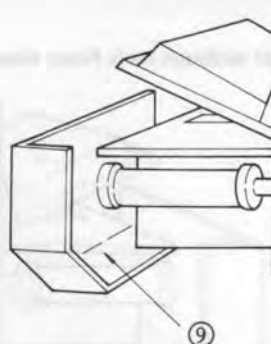
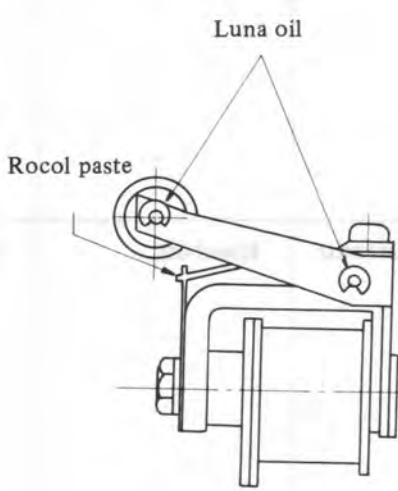


Fig. (d) Tape reader side view

1.2 Tape Reader Lubrication

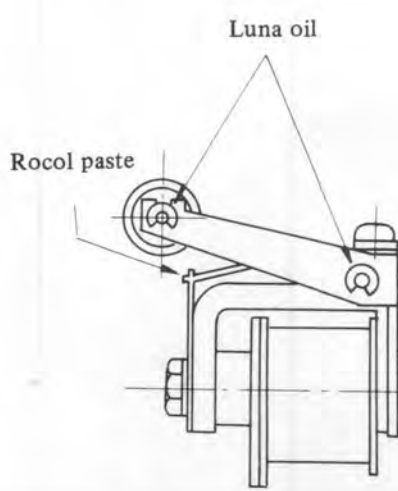
(a) Tape reader without reels

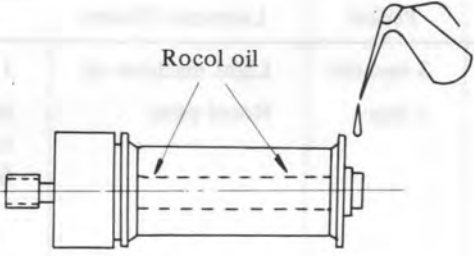
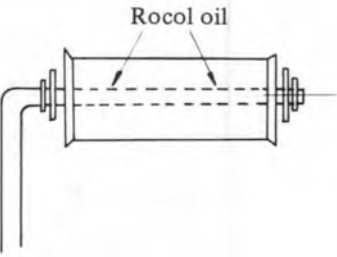
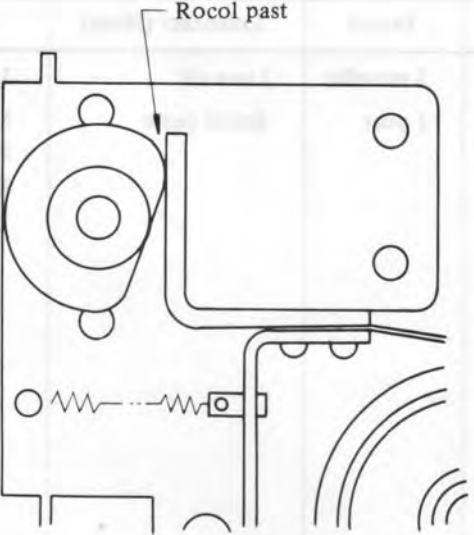
Routine lubrication points and lubrication periods are as follows:

Item	Lubrication point	Period	Lubricant (Note)	Amount
1	Magnet section  See Fig. 1.1 (a) ⑩	3 months 1 year	Light machine oil Rocol paste	1 drop Sufficient to form a thin film

(b) Tape reader with reels

Routine lubrication points and lubrication periods are as follows:

Item	Lubrication point	Period	Lubricant (Note)	Amount
1	Magnet section  See Fig. 1.2 (b) ①	3 months 1 year	Luna oil Rocol paste	1 drop Sufficient to form a thin film

Item	Lubrication point	Period	Lubricant (Note)	Amount
2	Guide roller  See Fig. 1.2 (b) ②	6 months	Rocol oil	2 ~ 3 drops
3	Tension arm guide roller  See Fig. 1.2 (c) ③	6 months	Rocol oil	2 ~ 3 drops
4	Cam  See Fig. 1.2 (d) ④	3 months	Rocol paste	Sufficient to form a thin film

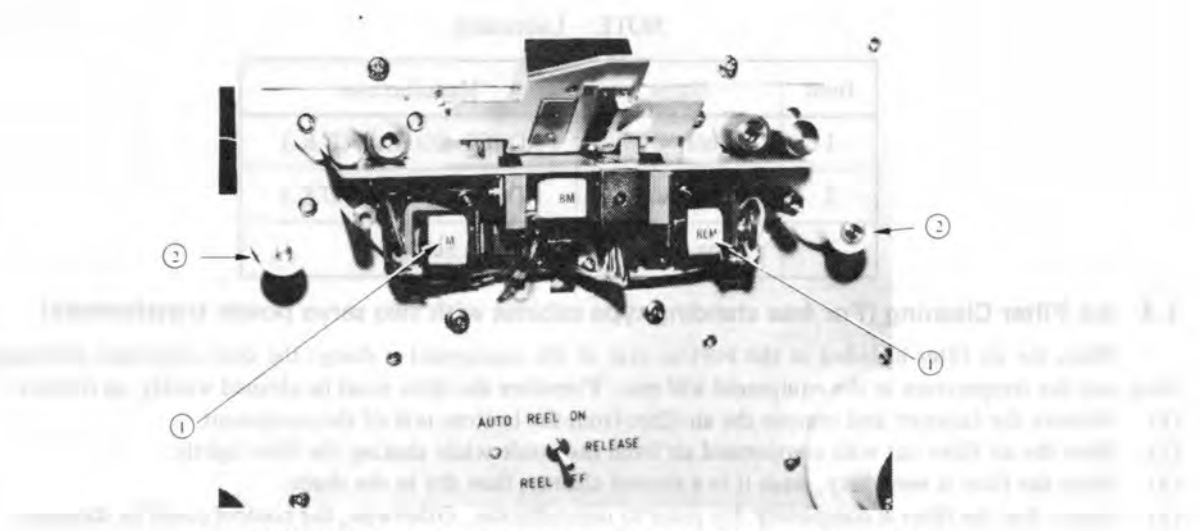


Fig. 1.2 (b) Tape reader with reels front view (with cover removed)

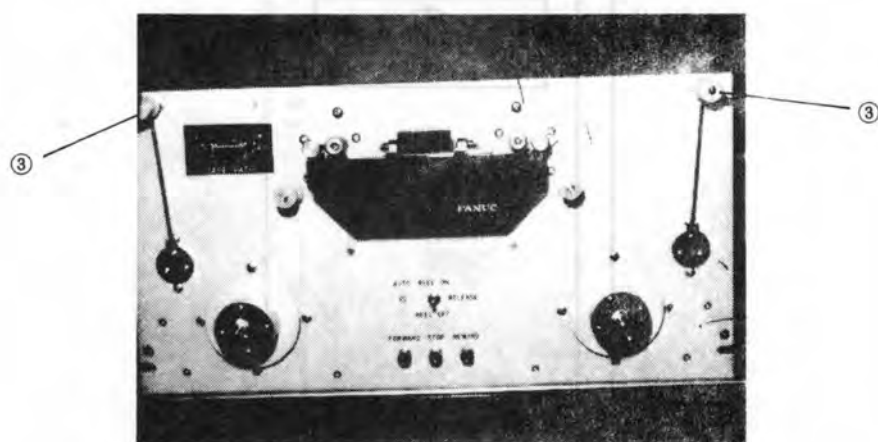


Fig. 1.2 (c) Tape reader with reels front view

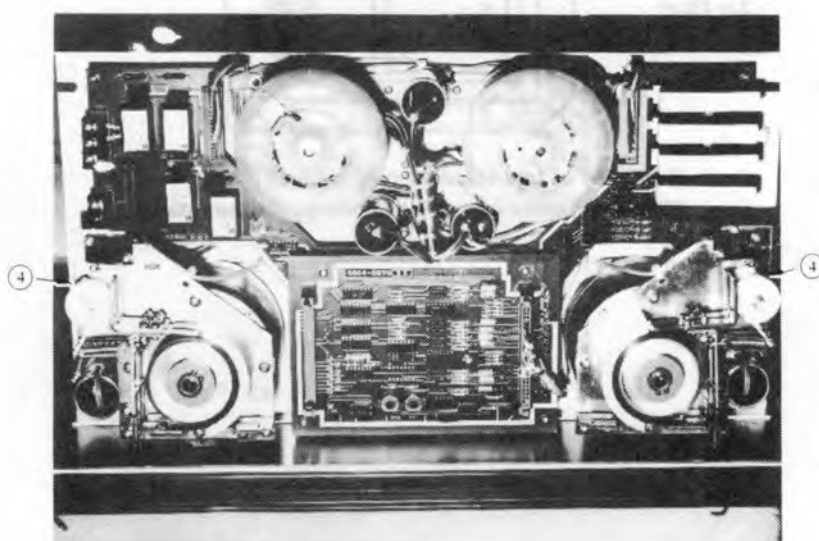


Fig. 1.2 (d) Tape reader with reels rear view

NOTE: Lubricants

Item	Name	Manufacturer
1	Rocol oil	ROCOL CO. Ltd. (U.K.)
2	Rocol paste	ROCOL CO. Ltd. (U.K.)
3	Luna oil	Nippon Sekiyu

1.3 Air Filter Cleaning (For free standing type cabinet with two servo power transformers)

When the air filter installed at the bottom rear of the equipment is dusty, the dust collection efficiency will drop, and the temperature in the equipment will rise. Therefore the filter must be cleaned weekly, as follows:

- (1) Remove the fastener, and remove the air filter from the bottom rear of the equipment.
- (2) Blow the air filter out with compressed air from the inside while shaking the filter lightly.
- (3) When the filter is very dirty, wash it in a neutral cleaner, then dry in the shade.
- (4) Insure that the filter is completely dry prior to reinstallation. Otherwise, the control could be damaged.



Rear view of free-standing type cabinet
(When two servo power transformers are mounted.)

2. FUSE CHECK AND REPLACEMENT

When a fuse blows inside the NC unit, find and correct the cause of the failure and replace the fuse. Fuses used in the NC unit are as follows:

2.1 Input Unit

The following two kinds of input units are provided.

- (1) For control unit Fig. 2.1 (a)
- (2) For control unit and servo Fig. 2.1 (b)

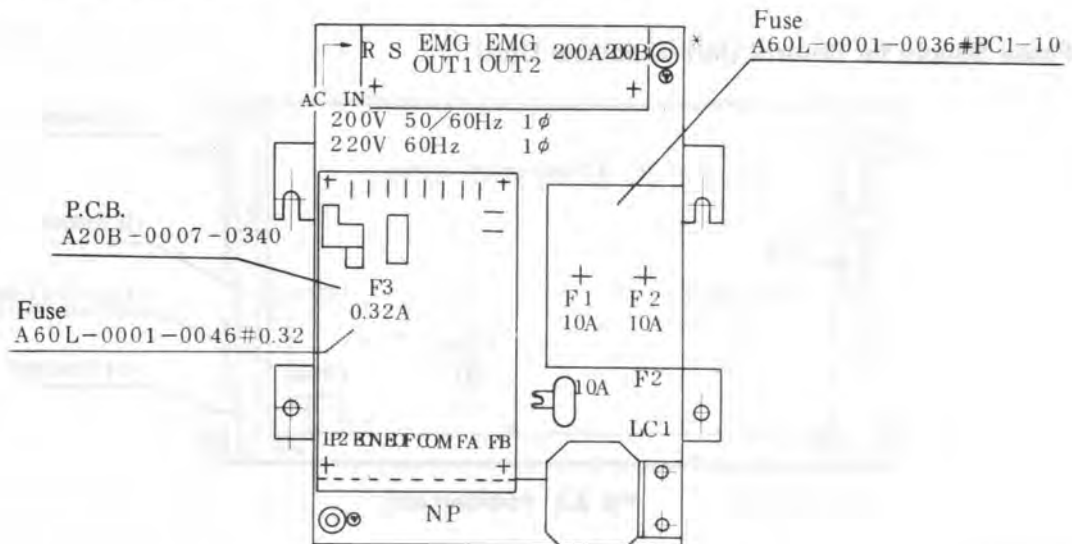


Fig. 2.1 (a) Input unit for control unit

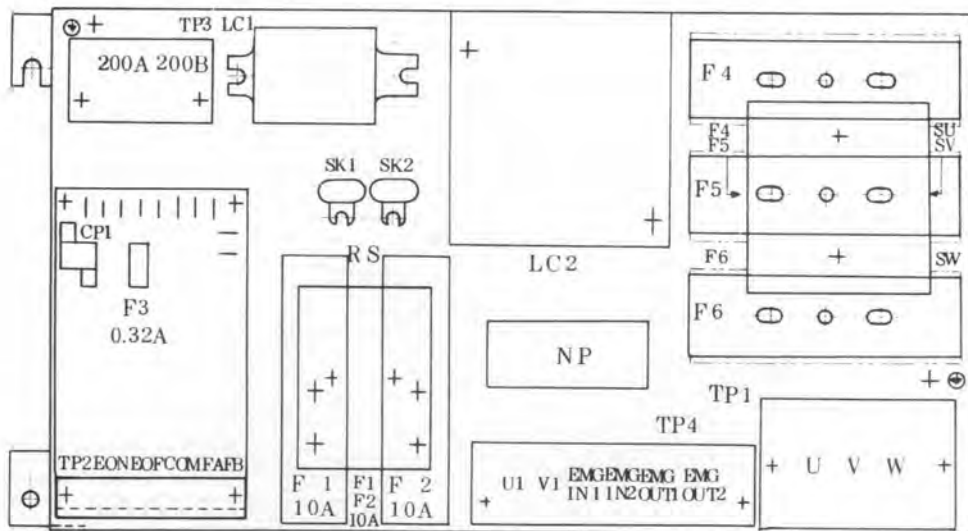


Fig. 2.1 (b) Input unit for control unit and servo unit

- | | |
|---------------|---|
| F1, F2 | Input fuses for control unit 10A |
| F3 | Fuse for power on/off control circuit 0.32A |
| F4 ~ F6 | Input fuses for servo transformer |

The rating of fuses used for the servo transformer primary input are listed below.

Power voltage	Fuse type	Utsunomiya Electric Co., Ltd. PC type	Fuji Electric Co., Ltd. FCF type
	Transformer capacity (KVA)		
200V	1.5KVA	15A	20A
220V	2.5	20	30
2	5	30	30
500V	10	40	40

2.2 Power Supply for Control Unit (Stabilizer Unit)

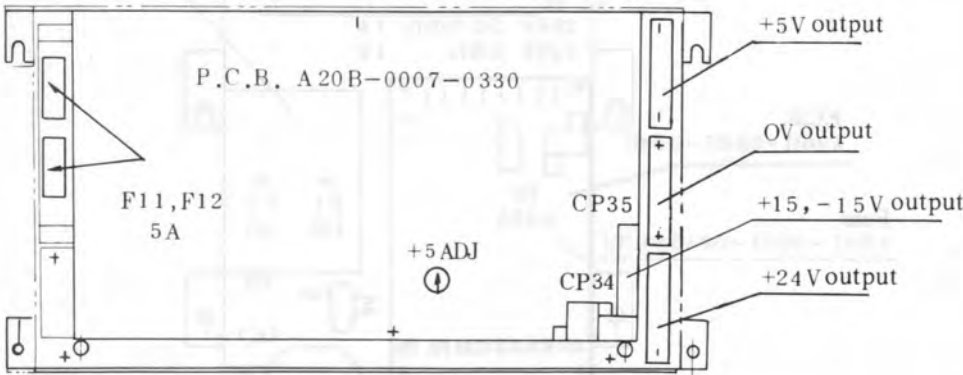


Fig. 2.2 Stabilizer unit

F11, F12 Input fuses for stabilizer unit 5A



3. CHECK AND REPLACEMENT OF DC MOTOR BRUSH

A periodic check and replacement of the brushes should be performed in the following way.

(1) Check interval

General machine tools

(Lathe, Milling machine, Machining center, etc.) Every year

Machine tools with high frequency of acceleration/deceleration

(Turret punch press, etc.) Every two months

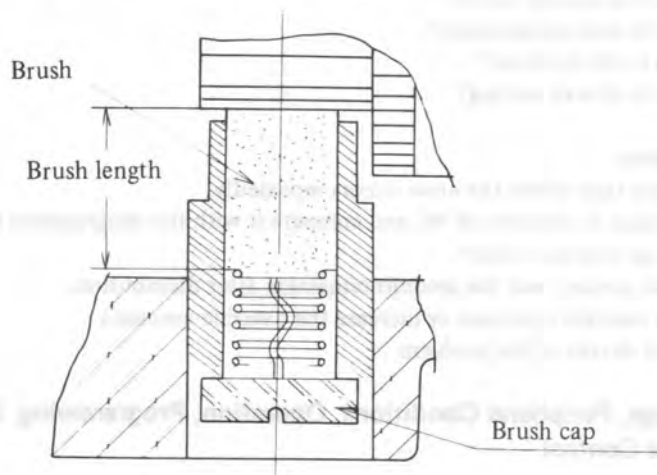
(2) Standard of brush replacement

When the length of the brush becomes shorter than 10 mm, the brush should be changed.

(3) Method of replacement

Remove the brush cap (screw) and take out the brush.

Tighten the brush cap fully when replacing it.



(Note) Replacement brushes can be obtained from our sales offices.

Part number: A290-0641-V001 (For model 0M and 5M).

A290-0651-V001 (For model 10M, 20M and 30M)

4. TROUBLESHOOTING

4.1 General

(1) Check the type of problem:

In what mode is the controller?

What is displayed on the MDI & CRT?

Is there a positioning error? If so, on which axis and by what amount?

Is there a tool path error? If so, by what amount?

Is the speed abnormal?

Is the problem in an auxiliary function?

What is the alarm number?

(2) Check the frequency of occurrence:

When did the problem occur? What is its frequency? (Was another machine also being operated?)

What is the frequency on the same workpieces?

Which program is it; What is the sequence number?

Is the problem related to a specific mode?

Is the problem related to tool replacement?

Is the problem related to the feedrate?

Is the problem related to thread cutting?

(3) For re-occurring problems:

Go through the program tape where the error occurs repeatedly.

Check the numerical value in memory of NC and compare it with the programmed numerical value.

Is the problem due to an external cause?

Check the stored offset amount and the amount remaining after distribution.

Check the response to override (decrease or increase the override amount).

Ask the operator about details of the problem.

4.2 Checking Input Voltage, Peripheral Conditions, Operation, Programming, Drives, Machine and Interface Control

(1) Check the input voltage:

Are there fluctuations in the input voltage?

Is there a drop in the input voltage?

Is front or rear door opened (door interlock)?

Is there some other device using large amounts of current?

Is there an electric discharge machine or welding machine nearby?

(2) Check peripheral conditions:

What is the temperature of the controller? Did the temperature change? Is it excessive?

Is the filter dirty?

Is the tape reader dirty?

Is there oil or cutting fluid about?

Are there any vibrations?

Is the unit in direct sunlight?

(3) Check for any external causes:

Has the machine recently been repaired or adjusted?

Has the magnetics cabinet recently been repaired or adjusted?

Has the NC unit recently been repaired or adjusted?

Is there a source of noise nearby?

(Example: Cranes, High-frequency machines, Electric discharge machines)

Has a new machine been mounted nearby?

Is there another NC with the same problem?

Has the user adjusted the NC?

Has the same problem occurred before?

(4) Check phases of operation

Has the operator been properly instructed?
Has the operator been replaced?
Is the operator familiar with the program?
Does the program finish too early or was it interrupted?
Does the program contain an incremental command?
Is the tool compensation value correctly set?
If the tool compensation value was changed, was it done correctly?
Does the machine change to another mode of operation?
Is the block skip function used correctly?
Is the tape set correctly?
Are there any tape coding errors?
Has the machine tool been operated incorrectly.

(5) Check punched tapes?

Is there dirt on the tape?
Are there any folds or wrinkles on the tape?
Are joints normal?
Did the program previously run normally?
Was the tape copied from a master tape?
Was it the punched tape?
Was the tape correctly punched?
Is the tape puncher normal?
Was the wrong tape used? (For example, colored tape?)
Is the puncher in good condition?
Was black tape used?

(6) Check the program

Is the program new?
Was the program created according to the OPERATOR'S MANUAL?
Are addresses in the right order?
Does the problem occur in any specific block?
Are the correct speed and lead values set for thread cutting?
Is there space at the beginning and end of thread cutting?
Does the problem occur in the sub-program?
Was the list of tapes created for checking?

(7) Check for changes in operation

Has any change or adjustment been made in the operation procedure?
Has a fuse been blown?
Is the NC in the emergency stop status?
Is the machine tool ready?
Is the NC in the alarm status?
Is the MODE SELECT switch set correctly?
Is the switch on the tape reader set correctly?
Is the override switch set to the zero position?
Is the NC in the machine lock status?
Is the feed hold button pushed?

(8) Check the machine itself

Is the machine properly installed?
Does vibration occur during operation?
Is the tool tip normal?
Is there any offset due to tool exchange?
Is there sufficient backlash compensation?
Are there distortions in any part of the machine due to temperature changes?
Was the workpiece measured correctly?
Was the measurement made at a constant temperature? (1 meter of steel changes 10μ in length at a temperature change of 1°C)

Are the cables normal (bent, broken or damaged)?
Are the signal lines and power lines separated?

(9) Check the interface control

Are power lines and NC cables mounted separately?
Is the shield normal? Is a noise suppressor attached to the relay, solenoid, and motor?

4.3 NC System Check (No tools required)

(1) Check control unit external conditions

Is there damage to the cabinet?
Is the MDI & DPL unit normal?
Is the filter clean?
Is the reader clean?
Is the door of the reader closed?
Was operation made with the door open?
Check that chips accumulated on the cabinet did not fall inside when the door was opened.

(2) Check the tape reader

Is the tape reader dirty?
Do the feed and the brake magnets operate normally?

(3) Check inside the control unit

Is there dirt in the control unit?
Is the fan motor normal?
Is there corrosion?

(4) Check the power unit

Is the unit correctly connected?
Are all fuses OK?
Is the circuit breaker normal?
Is the voltage within the allowable range?
Are the shield and cable duct grounded correctly?
Is the wiring path OK?
Are all terminals fully tightened?

(5) The grounding

Is the grounding connection OK?
Is the shield ground OK?

(6) Check all cables

Are cable connectors fully pushed in?
Are there any abnormalities in internal cables?
Are there any abnormalities in external cables?
Are there any scratches, bends or breaks?

(7) Check printed circuit boards

Are all PCBs mounted properly?
Is the plug connector OK?
Are physical conditions normal (no distortions, etc.)?
What is the PCB edition?
Are connections between printed circuit boards good?

(8) Check the MDI & DPL unit:

Do the push buttons operate normally?
Is the tape cable normal?

4.4 NC Status

When it seems that the NC is executing nothing even during cycle operating status, the current status of the NC is displayed on CRT screen at the diagnostic number 700 or 701 by selecting DIAGNOSTIC page using DGNOS function button.

	7	0	0		CSCT	CITL	COVZ	CINP	CDWL	CMTN	CFIN
	7	6	5	4	3	2	1	0			

When a digit is a 1, the corresponding status is effective.

- CFIN: The M, S, or T function is being executed.
- CMTN: A move command in the cycle operation is being executed.
- CDWL: Dwell is being executed.
- CINP: An in-position check is being executed.
- COVZ: Override is at 0%.
- CITL: STLK or Interlock is on.
- CSCT: The controller is waiting for the speed arrival signal of the spindle to turn on.

	7	0	1			CRST					CTRD	
	7	6	5	4	3	2	1	0				

- CTRD: The controller is reading the NC command from the tape reader.
- CRST: One of the following: emergency stop, remote reset, reset & rewind or the reset button on the MDI & DPL panel is on.

A4 NC Status

When it seems that the NC is executing nothing even during cycle operating status, the current status of the NC is displayed on CRT screen at the diagnostic number 700 by selecting DIAGNOSTIC page using **BCNOS** function button.

		7	0	0
--	--	---	---	---

	CRST	CITL	COVS	CINP	CDWL	CMTN	CTRD
--	------	------	------	------	------	------	------

7 6 5 4 3 2 1 0

When a digit is 1, the corresponding status is effective.

- CRST: The controller is waiting for the speed actual signal of the spindle to turn on.
- CITL: STLK or Interlock is on.
- COVS: Override is at 0%.
- CINP: An in-position check is being executed.
- CDWL: Dwell is being executed.
- CMTN: A move command in the cycle operation is being executed.
- CTRD: The M, S, or T function is being executed.

		7	0	1
--	--	---	---	---

	CRST					CTRD	
--	------	--	--	--	--	------	--

7 6 5 4 3 2 1 0

- CTRD: The controller is reading the NC command from the tape reader.
- CRST: One of the following: emergency stop, manual reset, reset & rework or the reset button on the MDI & DPL panel is on.

APPENDIXES

APPENDIX

APPENDIXES	319
APPENDIX 1 TAPE CODES USED IN PROGRAMMING	323
APPENDIX 2 G FUNCTION TABLE :	325
APPENDIX 3 TABLE OF COMMAND VALUE RANGES	327
APPENDIX 4 NOMOGRAPHS	328
APPENDIX 5 TAPE JOINING	332
APPENDIX 6 PARAMETERS	333
APPENDIX 7 ALARM LIST	364
APPENDIX 8 STATUS WHEN TURNING THE POWER ON, WHEN CLEARED AND WHEN RESET ..	372
APPENDIX 9 STORED PITCH ERROR COMPENSATION FUNCTION	374
APPENDIX 10 LIST OF OPERATIONS	379
APPENDIX 11 PROGRAM KEY LOCK	382
APPENDIX 12 CUSTOM MACRO INTERRUPT FUNCTION	384

319	APPENDIXES
323	APPENDIX 1 TAPE CODES USED IN PROGRAMMING
325	APPENDIX 2 G FUNCTION TABLE
327	APPENDIX 3 TABLE OF COMMAND VALUE RANGES
328	APPENDIX 4 NOMOGRAMS
331	APPENDIX 5 TAPE JOINING
333	APPENDIX 6 PARAMETERS
364	APPENDIX 7 ALARM LIST
371	APPENDIX 8 STATUS WHEN TURNING THE POWER ON, WHEN CLEARED AND WHEN RESET
374	APPENDIX 9 STORED PITCH ERROR COMPENSATION FUNCTION
379	APPENDIX 10 LIST OF OPERATIONS
383	APPENDIX 11 PROGRAM KEY LOCK
384	APPENDIX 12 CUSTOM MACRO INTERRUPT FUNCTION

APPENDIX 1 TAPE CODES USED IN PROGRAMMING

ISO code									EIA code									Meaning
Character	8	7	6	5	4	3	2	1	Character	8	7	6	5	4	3	2	1	
0			○	○	○				0			○		○				Numeral 0
1	○		○	○	○	○		○	1					○			○	Numeral 1
2	○		○	○	○	○	○		2					○		○		Numeral 2
3			○	○	○	○		○	3				○	○		○	○	Numeral 3
4	○		○	○	○	○	○		4					○	○			Numeral 4
5			○	○	○	○	○	○	5				○	○	○		○	Numeral 5
6			○	○	○	○	○	○	6				○	○	○	○		Numeral 6
7	○		○	○	○	○	○	○	7					○	○	○	○	Numeral 7
8	○		○	○	○	○			8					○	○			Numeral 8
9			○	○	○	○		○	9				○	○			○	Numeral 9
A		○				○		○	a		○	○		○			○	Address A
B		○				○		○	b		○	○		○		○	?	Address B
C	○	○				○		○	c		○	○	○		○		○	Address C
D	○					○	○		d		○	○		○	○			Address D
E	○	○				○	○	○	e		○	○	○	○	○		○	Address E
F	○	○				○	○	○	f		○	○	○	○	○		○	Address F
G		○				○	○	○	g		○	○		○	○	○	○	Address G
H		○			○	○			h		○	○		○	○		?	Address H
I	○	○			○	○		○	i		○	○	○	○	○		○	Address I
J	○	○			○	○		○	j		○	○		○			○	Address J
K		○			○	○		○	k		○	○				○		Address K
L	○	○			○	○	○		l		○			○	○		○	Address L
M		○			○	○	○	○	m		○	○		○	○			Address M
N		○			○	○	○	○	n		○			○	○		○	Address N
O	○	○			○	○	○	○	o		○			○	○			Address O
P		○	○		○				p		○	○		○	○	○	○	Address P
Q	○	○	○		○			○	q		○	○	○	○				Address Q
R	○	○	○		○		○		r		○		○	○			○	Address R
S		○	○		○		○	○	s			○	○	○		○		Address S
T	○	○		○		○			t			○		○		○	○	Address T
U		○	○		○	○		○	u			○	○		○			Address U
V		○	○		○	○	○		v			○		○	○		○	Address V
W	○	○	○		○	○	○	○	w		○			○	○	○		Address W
X	○	○	○	○		○			x		○	○		○	○	○	○	Address X
Y		○	○	○	○			○	y			○	○	○			?	Address Y
Z		○	○	○	○			○	z			○	○	○			○	Address Z
DEL	○	○	○	○	○	○	○	○	Del		○	○	○	○	○	○	○	* Delete (cancel an error punch).
NUL						○			Blank					○				* Not punched. Can not be used in significant section in EIA code.
BS	○				○	○			BS			○	○	○		○		* Back space
HT					○	○		○	Tab			○	○	○	○	○		* Tabulator
LF or NL					○	○		○	CR or EOB	○				○				* End of block
CR	○				○	○	○	○										* Carriage return
SP	○	○			○				SP				○	○				* Space
%	○	○	○		○			○	ER					○	○	○	○	* Absolute rewind stop
(		○	○	○	○				(2-4-5)				○	○	○		○	* Control out (a comment is started)
)	○	○	○	○	○			○	(2-4-7)		○			○	○	○	○	* Control in (the end of a comment)
+			○	○	○		○	○	+			○	○	○	○			* Positive sign
-			○	○	○	○	○	○	-			○			○			* Negative sign
:			○	○	○	○	○	○	:									* Colon
/	○		○	○	○	○	○	○	/			○	○	○			○	* Optional block skip
.			○	○	○	○	○	○	.			○	○	○	○	○	○	* Period (A decimal point)
#	○	○			○		○	○	#									* Sharpe
\$		○			○	○			\$									* Dollar sign
&	○	○			○	○	○	○	&					○	○	○	○	* Ampersand
'		○			○	○	○	○	'									* Apostrophe
*	○	○	○		○	○		○	*									* Asterisk
,	○	○	○		○	○		○	,			○	○	○	○		○	* Comma
;	○	○	○	○	○	○		○	;									* Semicolon
<			○	○	○	○	○		<									* Left angle bracket
=	○	○	○	○	○	○		○	=									* Equal
>	○		○	○	○	○	○	○	>									* Right angle bracket
?			○	○	○	○	○	○	?									* Question mark
@	○	○			○				@									* Commercial at mark
"			○		○		○		"									* Quotation
[	○	○		○	○	○		○	[									* Left brace
]	○	○		○	○	○		○	]									* Right brace

APPENDIX 2 G FUNCTION TABLE

The following G codes can be used. The special G code can also be used by setting a parameter.

Standard G code	Special G code B	Special G code C	Group	Function	Basic/Option
G00	G00	G00	01	Positioning (Rapid traverse)	B
G01	G01	G01		Linear interpolation (Feed)	B
G02	G02	G02		Circular interpolation CW	B
G03	G03	G03		Circular interpolation CCW	B
G04	G04	G04	00	Dwell	B
G07	G07	G07		Feedrate sine curve control	
G10	G10	G10		Offset value setting	O
G20	G20	G70	06	Inch data input	O
G21	G21	G71		Metric data input	O
G22	G22	G22	04	Stored stroke limit ON	O
G23	G23	G23		Stored stroke limit OFF	O
G27	G27	G27	00	Reference point return check	O
G28	G28	G28		Return to reference point	O
G29	G29	G29		Return from reference point	O
G30	G30	G30		Return to 2nd reference point	O
G31	G31	G31		Skip cutting	O
G32	G33	G33	01	Thread cutting (Straight, taper)	B
G34	G34	G34		Variable lead thread cutting	O
G36	G36	G36	00	Automatic tool compensation X	O
G37	G37	G37		Automatic tool compensation Z	O
G40	G40	G40	07	Tool nose radius compensation cancel	O
G41	G41	G41		Tool nose radius compensation left	O
G42	G42	G42		Tool nose radius compensation right	O
G50	G92	G92	00	Programming of absolute zero point Maximum spindle speed setting	B, O
G65	G65	G65		User macro simple calling	O
G66	G66	G66	12	User macro modal calling	O
G67	G67	G67		User macro modal call cancellation	O
G68	G68	G68	13	Mirror image for double turrets ON	O
G69	G69	G69		Mirror image for double turrets OFF	O
G70	G70	G72	00	Finishing cycle	O
G71	G71	G73		Stock removal in turning	O
G72	G72	G74		Stock removal in facing	O
G73	G73	G75		Pattern repeating	O
G74	G74	G76		Peck drilling in Z axis	O
G75	G75	G77		Grooving in X axis	O
G76	G76	G78		Thread cutting cycle	O
G90	G77	G20		Cutting cycle A	O
G92	G78	G21	01	Thread cutting cycle	O
G94	G79	G24		Cutting cycle B	O
G96	G96	G96	02	Constant surface speed control	O
G97	G97	G97		Constant surface speed control cancel	O
G98	G94	G94	05	Per minute feed	B
G99	G95	G95		Per revolution feed	B
—	G90	G90	03	Absolute programming	B
—	G91	G91		Incremental programming	B

(Note 1) The G codes marked with ∇ are initial G codes in each group. That is, when the power is turned on or when the reset button is pressed when the system parameter specifying initialization of G codes at reset is effective, those G codes are set.

For G22 and G23, G22 is selected after the power was turned on. After resetting, G22 or G23, the one effective before resetting is set.

For G00 and G01, G98 and G99, or G90 and G91, either is selected for the initial G codes by setting a parameter (G00, G98, G90).

For G20 and G21, the one in effect before cutting power or pressing the reset button is selected.

(Note 2) The G codes in group 00 are not modal. They are effective only in the block in which they are specified.

(Note 3) An alarm (No. 010) occurs when a G code not listed in the above table is specified. When an optional G code not stored in the controller is specified, alarm (No. 010) occurs. However, G60 and G61 are ignored.

(Note 4) A number of G codes can be specified in a block even if they do not belong to the same group. When a number of G codes in the same group are specified, the G code specified last is effective.

(Note 5) A G code from each group is displayed.

Group	G Code	Initial	Parameter	Initial	Parameter	Initial	Parameter
00	G00	∇	0	0	0	0	0
01	G01	∇	0	0	0	0	0
02	G02	∇	0	0	0	0	0
03	G03	∇	0	0	0	0	0
04	G04	∇	0	0	0	0	0
05	G05	∇	0	0	0	0	0
06	G06	∇	0	0	0	0	0
07	G07	∇	0	0	0	0	0
08	G08	∇	0	0	0	0	0
09	G09	∇	0	0	0	0	0
10	G10	∇	0	0	0	0	0
11	G11	∇	0	0	0	0	0
12	G12	∇	0	0	0	0	0
13	G13	∇	0	0	0	0	0
14	G14	∇	0	0	0	0	0
15	G15	∇	0	0	0	0	0
16	G16	∇	0	0	0	0	0
17	G17	∇	0	0	0	0	0
18	G18	∇	0	0	0	0	0
19	G19	∇	0	0	0	0	0
20	G20	∇	0	0	0	0	0
21	G21	∇	0	0	0	0	0
22	G22	∇	0	0	0	0	0
23	G23	∇	0	0	0	0	0
24	G24	∇	0	0	0	0	0
25	G25	∇	0	0	0	0	0
26	G26	∇	0	0	0	0	0
27	G27	∇	0	0	0	0	0
28	G28	∇	0	0	0	0	0
29	G29	∇	0	0	0	0	0
30	G30	∇	0	0	0	0	0
31	G31	∇	0	0	0	0	0
32	G32	∇	0	0	0	0	0
33	G33	∇	0	0	0	0	0
34	G34	∇	0	0	0	0	0
35	G35	∇	0	0	0	0	0
36	G36	∇	0	0	0	0	0
37	G37	∇	0	0	0	0	0
38	G38	∇	0	0	0	0	0
39	G39	∇	0	0	0	0	0
40	G40	∇	0	0	0	0	0
41	G41	∇	0	0	0	0	0
42	G42	∇	0	0	0	0	0
43	G43	∇	0	0	0	0	0
44	G44	∇	0	0	0	0	0
45	G45	∇	0	0	0	0	0
46	G46	∇	0	0	0	0	0
47	G47	∇	0	0	0	0	0
48	G48	∇	0	0	0	0	0
49	G49	∇	0	0	0	0	0
50	G50	∇	0	0	0	0	0
51	G51	∇	0	0	0	0	0
52	G52	∇	0	0	0	0	0
53	G53	∇	0	0	0	0	0
54	G54	∇	0	0	0	0	0
55	G55	∇	0	0	0	0	0
56	G56	∇	0	0	0	0	0
57	G57	∇	0	0	0	0	0
58	G58	∇	0	0	0	0	0
59	G59	∇	0	0	0	0	0
60	G60	∇	0	0	0	0	0
61	G61	∇	0	0	0	0	0
62	G62	∇	0	0	0	0	0
63	G63	∇	0	0	0	0	0
64	G64	∇	0	0	0	0	0
65	G65	∇	0	0	0	0	0
66	G66	∇	0	0	0	0	0
67	G67	∇	0	0	0	0	0
68	G68	∇	0	0	0	0	0
69	G69	∇	0	0	0	0	0
70	G70	∇	0	0	0	0	0
71	G71	∇	0	0	0	0	0
72	G72	∇	0	0	0	0	0
73	G73	∇	0	0	0	0	0
74	G74	∇	0	0	0	0	0
75	G75	∇	0	0	0	0	0
76	G76	∇	0	0	0	0	0
77	G77	∇	0	0	0	0	0
78	G78	∇	0	0	0	0	0
79	G79	∇	0	0	0	0	0
80	G80	∇	0	0	0	0	0
81	G81	∇	0	0	0	0	0
82	G82	∇	0	0	0	0	0
83	G83	∇	0	0	0	0	0
84	G84	∇	0	0	0	0	0
85	G85	∇	0	0	0	0	0
86	G86	∇	0	0	0	0	0
87	G87	∇	0	0	0	0	0
88	G88	∇	0	0	0	0	0
89	G89	∇	0	0	0	0	0
90	G90	∇	0	0	0	0	0
91	G91	∇	0	0	0	0	0
92	G92	∇	0	0	0	0	0
93	G93	∇	0	0	0	0	0
94	G94	∇	0	0	0	0	0
95	G95	∇	0	0	0	0	0
96	G96	∇	0	0	0	0	0
97	G97	∇	0	0	0	0	0
98	G98	∇	0	0	0	0	0
99	G99	∇	0	0	0	0	0

APPENDIX 3 TABLE OF COMMAND VALUE RANGES

	Input in mm Output in mm	Input in inches Output in mm	Input in mm Output in inches	Input in inches Output in inches
Least input increment	0.001 mm	0.0001 inch	0.001 mm	0.0001 inch
Maximum stroke (distance from the reference point)	±99999.999 mm	±99999.999 mm	±3937.0078 inches	±9999.9999 inches
Maximum programmable dimension	±99999.999 mm	±3937.0078 inches	±99999.999 mm	±9999.9999 inches
For a cutting feedrate override of 100%	Feed per minute Feed per revolution	0.01 ~ 600.00 inches/min 0.0001 ~ 50.0000 inches/rev	1 ~ 15000 mm/min 0.01 ~ 500.00 mm/rev	0.01 ~ 600.00 inches/min 0.0001 ~ 50.0000 inches/rev
Rapid traverse rate (Separate for each axis)	30 ~ 15000 mm/min	30 ~ 15000 mm/min	3.0 ~ 600.0 inches/min	3.0 ~ 600.0 inches/min
Upper limit of value of cutting feedrate				
Manual rapid traverse rate	6 ~ 15000 mm/min	6 ~ 15000 mm/min	0.6 ~ 600.0 inches/min	0.6 ~ 600.0 inches/min
Fo				
Manual jog feedrate	1 ~ 2000 mm/min	0.04 ~ 78.7 inches/min	0.5 ~ 1016 mm/min	0.02 ~ 40 inches/min
Thread lead	F code E code	0.0001 ~ 50.0000 inches 0.000001 ~ 9.999999 inches	0.01 ~ 500.00 mm 0.0001 ~ 500.0000 mm	0.0001 ~ 50.0000 inches 0.000001 ~ 9.999999 inches
Maximum spindle speed	5000 rpm	5000 rpm	5000 rpm	5000 rpm
Coordinate value of 2nd reference point (Distance from reference point)	0 ~ ±99999.999 mm	0 ~ ±99999.999 mm	0 ~ ±3937.0078 inches	0 ~ ±9999.9999 inches
Tool offset	0 ~ ±999.999 mm	0 ~ ±99.9999 inches	0 ~ ±999.999 mm	0 ~ ±99.9999 inches
Minimum value in incremental feed	0.001 mm	0.0001 inch	0.001 mm	0.0001 inch
Backlash compensation amount	0 ~ 0.255 mm	0 ~ 0.255 mm	0 ~ 0.0255 inch	0 ~ 0.0255 inch
Area of stored stroke limit (Value from reference point)	±99999.999 mm	±99999.999 mm	±3937.0078 inches	±9999.9999 inches
Dwell	0 ~ 99999.999 sec	0 ~ 99999.999 sec	0 ~ 99999.999 sec	0 ~ 99999.999 sec

APPENDIX 4 NOMOGRAPHS

4.1 Incorrect Threaded Length

The leads of a thread are generally incorrect in δ_1 and δ_2 , as shown in Fig. 4.1, due to automatic acceleration and deceleration.

Thus distance allowances must be made to the extent of δ_1 and δ_2 in the program

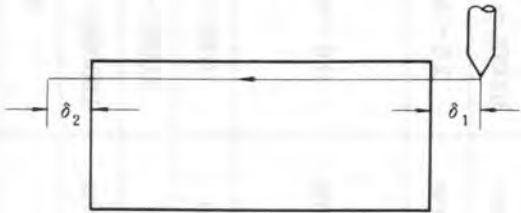


Fig. 4.1 Incorrect threaded length

- (1) δ_2 is determined by cutting speed V (mm/sec) and time constant (T_1) of the servo system as shown below.

$\delta_2 = T_1 \cdot V \text{ (mm)}$ (1)

where T_1 is in sec., and V is in mm/sec.
 V is determined by thread lead L and spindle speed R as shown below.
 $V = R \text{ (rpm)} \cdot L \text{ (mm)}/60$
 $L \text{ (mm)}$ Lead of thread
 $R \text{ (rpm)}$ Spindle-speed
o Time constant T_1 (sec.) of the servo system: Usually 0.033 sec.

- (2) **How δ_1 is determined**

The value of δ_1 is determined by cutting speed V in thread cutting, time constant T_1 for the servo system, and thread accuracy “a” as shown below.

$\delta_1 = \left\{ t - T_1 + T_1 \exp \left(-\frac{t}{T_1} \right) \right\} V \text{ (pulses)}$ (2)

$a = \exp \left(-\frac{t}{T_1} \right)$ (3)

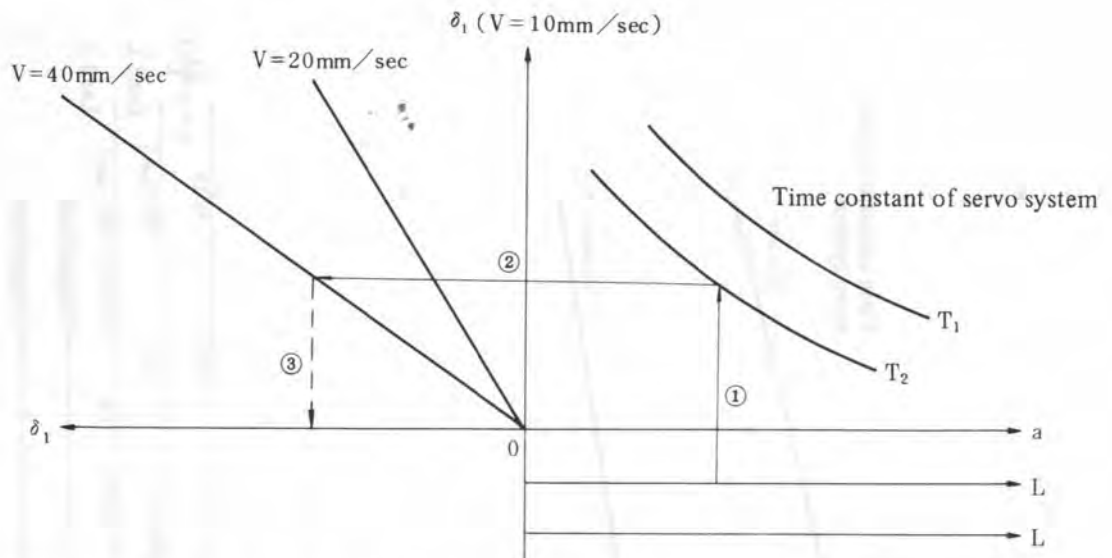
The lead at the beginning of thread cutting is shorter than the specified lead, L , and the allowable lead error is ΔL .

Then,

$a = \frac{\Delta L}{L}$

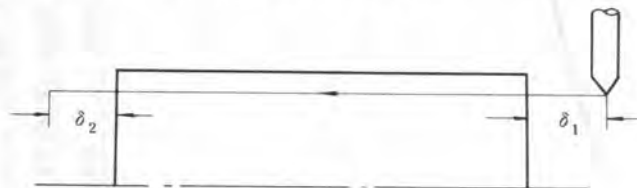
When the value of “a” is determined, the time lapse, t , until the thread accuracy is attained can be calculated by formula (3). The time “t” is substituted in (2) to determine δ_1 . Constants V , T_1 and T_2 are determined in the same way as for δ_2 .

Since the calculation of δ_1 is rather complex, a nomography is provided on the following pages. Instructions on how to read the appropriate δ_1 value on the nomography is shown below.



First specify the class and the lead of a thread. The thread accuracy, a , will be obtained at ①, and depending on the time constant of the speed-up circuit of the machine used, δ value when $V = 10 \text{ mm/sec}$ will be obtained at ②. Then, depending on the speed of thread cutting the approach distance δ_1 will be obtained at ③ for other speeds than 10 mm/sec .

4.2 Simple Calculation of Incorrect Thread Length



$$(1) \quad \delta_2 = \frac{L \cdot R}{1800^*} \text{ (mm)} \quad \begin{array}{l} L: \text{ Thread lead (mm)} \\ R: \text{ Spindle speed (rpm)} \end{array}$$

$$(2) \quad \delta_1 = \frac{L \cdot R}{1800^*} \cdot (-1 - 1na) = \delta_2 \cdot (-1 - 1na)$$

* When time constant T of the servo system is 0.033 sec .

An " a " indicates the error allowances. And the " $-1 - 1na$ " follows as below.

a	$-1 - 1na$
0.005	4.298
0.01	3.605
0.015	3.200
0.02	2.912

(Example)

$$R = 350 \text{ rpm}$$

$$L = 1 \text{ mm}$$

$$a = 0.01$$

$$\delta_2 = \frac{350 \times 1}{1800} = 0.194 \text{ mm}$$

$$\delta_1 = \delta_2 \times 3.605 = 0.701 \text{ mm}$$

V: speed in thread cutting

$V = 40 \text{ mm/sec}$ ($\div 1.57 \text{ in/sec}$)
 $V = 30 \text{ mm/sec}$ ($\div 1.18 \text{ in/sec}$)
 $V = 20 \text{ mm/sec}$ ($\div 0.79 \text{ in/sec}$)
 $V = 10 \text{ mm/sec}$ ($\div 0.39 \text{ in/sec}$)

$V = 2 \text{ in/sec}$

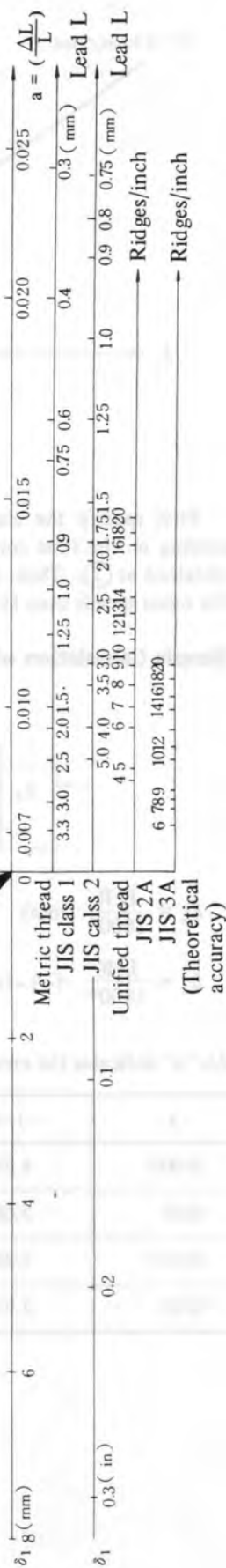
$V = 1 \text{ in/sec}$

δ_1 ($V = 10 \text{ mm/sec}$)

Time constant of the
servo system

50 msec

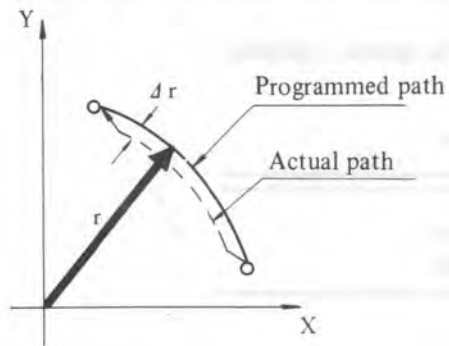
33 msec



Nomograph for approach distance δ_1

4.3 Error in Radius Direction in Circular Cutting

In linear interpolation, the tool moves on a specified segment of a line; thus no error is generated. In circular interpolation, especially in high-speed circular cutting, an error is generated in the radius direction. This error can be calculated by the procedure described below.



Δr : Maximum error in radius direction (mm)
 v : Feedrate (mm/sec.)
 r : Radius of arc (mm)
 T_1 : Time constant for cutting feed
 T_2 : Time constant for servo system

$$\Delta r = \frac{1}{2} \cdot (T_1^2 + T_2^2) \cdot \frac{v^2}{r} \quad (1)$$

For actual processing, a processing radius, r (mm), of the workpiece and an allowable error, Δr (mm), is given, and the allowable speed limit, v (mm/sec.), can be calculated by formula (1).

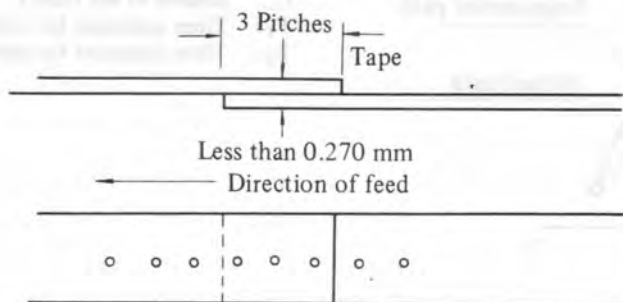
Since acceleration/deceleration time constant for cutting feed differs with machine tool builders, please refer to the manual issued by your machine tool builder.

The time constant for servo system is $T_2 = 33 \times 10^{-3}$ (sec), for DC servo motors.

APPENDIX 5 TAPE JOINING

If a loop of punched tape is to be made or tape is broken or in similar cases, two tape sections must be joined. Join these tape section as described below.

- Butt the two tape sections and paste them together, placing the section on the side on which the tape is fed, on top of the other.
- The joint at which the two tape sections overlap must be approx. 3 pitches.



- Make sure the punched holes of the overlapped tape sections at the joint are correctly aligned. Use special care with the feed holes.

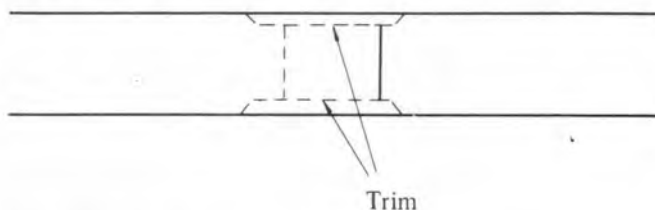


Incorrect



Correct

- Trim both edges of the joint to smooth them.



- Make sure punched holes are not clogged up with the paste.

APPENDIX 6 PARAMETERS

Display and Setting Procedure of Parameter

Parameters must be set correctly so that the DC servo motor and the machine tool are given their full characteristics, specifications and functions, when the NC is combined with the machine tool. Since parameters differ with machine tool, refer to the parameter table prepared by machine tool builder for the contents. End user must not change parameters.

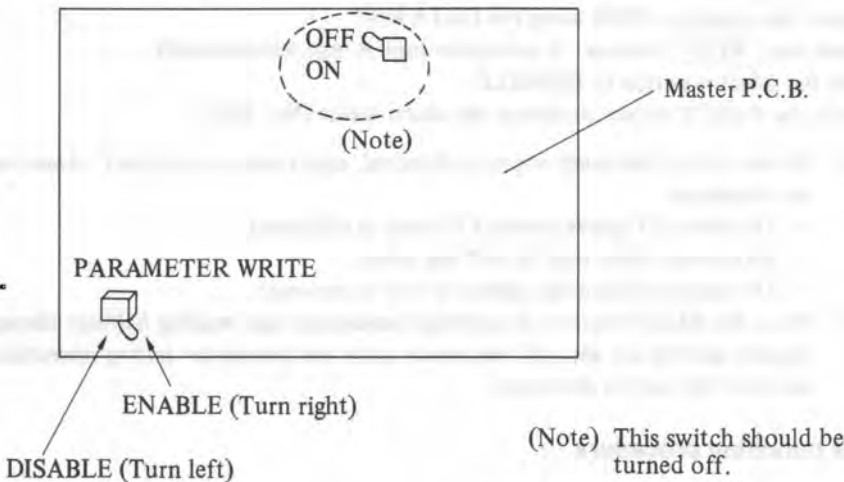
6.1 Parameter display procedure

- (a) Press the **PARAM** button.
- (b) Press the **PAGE** button to select the page. **N** a parameter number **INPUT** can also be used to select a screen.

6.2 Parameter setting procedure

(1) Setting from the MDI

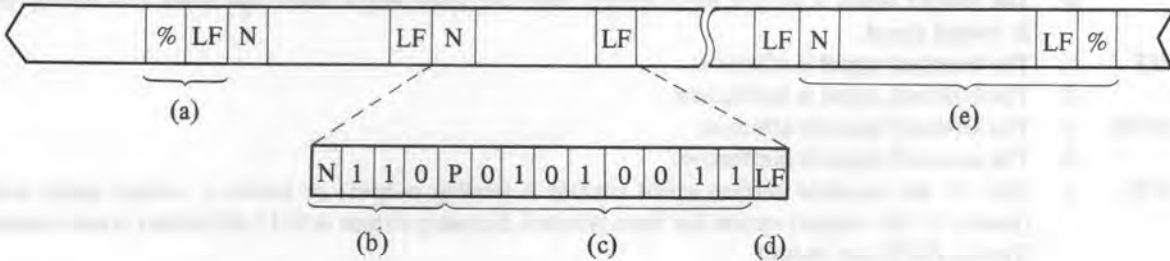
- (a) Set the selector switch on the master P.C.B. to ENABLE.



- (b) Set the mode switch to MDI.
- (c) Press the **PARAM** button.
- (d) Keying in **N** and **parameter number to be set** and pressing the **INPUT** button, selects the page of that parameter number and moves the cursor to below the number (the **PAGE** button and the **CURSOR** button can also be used).
- (e) Key in **P** and **data to be set** and press the **INPUT** button. When you are aware of an error during keying-in, press the **CAN** button.
- (f) Check whether the data was set correctly.
- (g) Turn the **PARAMETER WRITE** switch on the master PCB to **DISABLE** after all parameters were set and confirmed.
- (h) Push a reset button to release alarm status (number 100).

(2) Setting from a tape

Parameters can also be entered from a tape. Prepare a tape for parameter setting as follows.



- (a) Punch %LF (for ISO code) or ER CR (for EIA code) at the start of the tape.
- (b) Punch the parameter number following address N after the LF (or CR).
- (c) Punch the parameters corresponding to the parameter number following address P.
- (d) Punch LF (or CR).
Repeat (b) through (d) the required number of times. Each block must start with a parameter number following address N.
Leading zeros can be omitted for parameters following address P.
- (e) Punch LF% or (or CRER) at the end to stop reading parameters.

Parameters not punched on the tape do not change after the tape is read into the NC. So you can prepare several tapes; one for backlash compensation data, one for pitch error compensation data, etc.

A parameter tape is read into the NC as follows:

- ① Set the selector switch on the master PCB to ENABLE. Alarm No. 100 will be displayed on the CRT.
- ② Load the parameter tape in the tape reader.
- ③ Push the EMERGENCY STOP button. Confirm the LABEL SKIP status. If the LABEL SKIP status is not in effect, push the EMERGENCY STOP button in a mode other than MDI mode to enter LABEL SKIP status.
- ④ Push the **PARAM** button.
- ⑤ Push **P** button.
- ⑥ Input the number -9999 using the DATA key.
- ⑦ Push the **READ** button. A parameter tape is read automatically.
- ⑧ Set the selector switch to DISABLE.
- ⑨ Push the RESET button to release the alarm status (No. 100).

(Note 1) If one of the following errors is detected, tape reading is stopped. Alarm numbers, however, are not displayed.

- TH alarm, TV alarm (when TV check is effective).
- An address other than N or P was input.
- The number following address N or P is incorrect.

(Note 2) Press the RESET button to interrupt parameter tape reading halfway through. Some parameters require turning on and off the power after the parameter setting operations. At this time alarm number 000 will be displayed.

6.3 Parameter punching procedure

- (a) Connect the puncher to the tape punch unit.
- (b) Select punching code, EIA/ISO.
- (c) Set the MODE SELECT switch to EDIT.
- (d) Push the **PARAM** button.
- (e) Input **P** **-** **9** **9** **9** **9** **PUNCH**.

(Note) When a puncher is connected or disconnected to the NC, both NC and puncher power must be turned off.

6.4 Parameters

	0	0	5
--	---	---	---

ORWD	EIT	EENB	SCD	FMIC	MDL	MIC	SCW
7	6	5	4	3	2	1	0

- ORWD 1: The rewind signal is output only when a tape is rewound by the reset & rewind signal.
0: The rewind signal is output for a certain time (100 msec) even when tape is not rewound by the reset & rewind signal.
- EIT 1: The interlock signal is effective.
0: The interlock signal is ineffective.
- EENB 1: The servo-off signal is effective.
0: The servo-off signal is ineffective.
- SCD 1: Even if the constant surface speed control B (analog output) or constant surface speed control A (binary 12 bit output) option has been selected, S analog output or S 12 bit output is not executed but S codes (BCD) are output.
0: If the constant surface speed control B (analog output) or constant surface speed control A (binary 12 bit output) option has been selected, the proper data is output. If those options are not used, S codes (BCD) are output.

- FMIC 1: The least input increment of feedrate for metric input is 1/10.
 0: The least input increment of feedrate for metric input is not 1/10.
- MDL 1: The least display increment of position display unit for metric input (inch input) is 0.01 mm (0.0001 inch).
 0: The least display increment of position display unit for metric input (inch input) is 0.001 mm (0.0001 inch).
- MIC 1: The least input increment for metric input (inch input) is 0.01 mm (0.0001 inch).
 0: The least input increment for metric input (inch input) is 0.001 mm (0.0001 inch).
- SCW 1: The least command increment is 0.0001 inch. (Machine tool: inch system)
 0: The least command increment is 0.001 mm. (Machine tool: metric system)

	0	0	6
--	---	---	---

ADFT	EOM	CINP	DCS	CLER	TVC	PPD	RDRN
7	6	5	4	3	2	1	0

- ADFT 1: Automatic drift is compensated for.
 0: Automatic drift is not compensated for.
- EOM 1: When M30 is specified, M30 is sent to the machine tool, then when the NC receives the FIN signal, the execution continues from the head of the program.
 If the NC does not receive the FIN signal but the external reset signal, execution returns to the head of the program and the NC is reset. (Memory operation mode)
 0: When M30 is specified, M30 is sent to the machine tool but execution does not return to the head of a program unless the NC receives the reset & rewind signal. (Memory operation mode)
- CINP 1: NC control goes to the next block after the specified feedrate becomes zero with deceleration and the NC confirms that the machine position is at the specified position at the intersection of two blocks other than two cutting feed blocks or when error detection is effective. (Confirmation is called in-position check.)
 0: NC control goes to the next block after the specified feedrate becomes zero with deceleration at the intersection of two blocks other than two cutting feed blocks or when error detection is effective. (No in-position check is made.)
- DCS 1: The system is generated by the START button on the MDI panel without going through the machine tool. (In the MDI mode only)
 0: The start signal is sent to the machine tool by pressing the START button on the MDI panel. Then when the NC receives the start signal issued from the machine tool, the system is generated.
- CLER 1: The NC is cleared by the Reset button, External Reset or Reset & rewind signal.
 0: The NC is reset by the Reset button, External Reset or Reset & rewind signal. (Refer to appendix 8 for reset and clear status.)
- TVC 1: The TV check of the information in the comment zone is ineffective.
 0: The TV check of the information in the comment zone is effective.
- PPD 1: The indication of the position display unit is preset by G50.
 0: The indication of the position display unit is not preset by G50.
- RDRN 1: Dry run is effective for rapid traverse command.
 0: Dry run is not effective for rapid traverse command.

	0	0	7
--	---	---	---

ICR	IOF	GSP	SCTO	G90	G98		G00
7	6	5	4	3	2	1	0

- ICR 1: "LF" is punched as the EOB code in the ISO code system.
 0: "LF CR CR" is punched as the EOB code in the ISO code system.
- IOF 1: When the optional incremental offset is used, the input offset value is regarded as the incremental value.
 0: The input offset value is regarded as the absolute value.
- GSP 1: Special G code B is used.
 0: Standard G code is used.
- SCTO 1: In an S command or when the movement changes from the rapid traverse rate to the cutting feedrate, the speed arrival signal is checked.
 0: The speed arrival signal is not checked.

- G90 1: G90 is selected when the power is turned on or the system is cleared. (Special G code B/C only)
 0: G91 is selected when the power is turned on or the system is cleared. (Special G code B/C only)
- G98 1: G98 is selected when the power is turned on or the system is cleared. (G94 in the case of special G code B/C)
 0: G99 is selected when the power is turned on or the system is cleared. (G95 in the case of special G code B/C)
- G00 1: G00 is selected when the power is turned on or the system is cleared.
 0: G01 is selected when the power is turned on or the system is cleared.

	0	0	8
--	---	---	---

ORC	RWL		TLCC	T2D	TOC	NMTN	TDRN
7	6	5	4	3	2	1	0

- ORC 1: The offset value is a radius designation.
 0: The offset value is a diameter designation.
- RWL 1: The inhibited area of stored stroke limit 3 is outside.
 0: The inhibited area of stored stroke limit 3 is inside.
- TLCC 1: The new offset value is effective from the next block when the offset value is modified.
 0: The new offset value is effective from the next T code when the offset value is modified.
- T2D 1: The offset number of the T code is the lower 2 digits.
 0: It is the lowest 1 digit.
- TOC 1: Offset is canceled by a reset operation.
 0: Offset is not canceled by a reset operation.
- NMTN 1: M, S and T commands are executed but a move command is not when the STLK signal is on.
 0: Move commands and M, S, and T commands are not executed when the STLK signal is on.
- TDRN 1: Dry run in thread cutting is effective.
 0: Dry run in thread cutting is ineffective.

	0	0	9
--	---	---	---

TCW	CWM	SOV	SSCR	MOR	SOVC	REDT	ISOT
7	6	5	4	3	2	1	0

TCW, CWM The sign of the output of the S-4 digit (analog output)

Parameter		Sign in signal output	
TCW	CWM	M03	M04
0	0	+	+
0	1	-	-
1	0	+	-
1	1	-	+

- SOV 1: Spindle override is effective.
 0: Spindle override is ineffective.
- SSCR 1: The surface speed in a rapid traverse block in constant surface speed control mode is calculated from the end point coordinate value of the X-axis.
 0: The surface speed is calculated from the present value of X-axis as well as the cutting feed in constant surface speed control mode.
- MOR 1: In canned cycle mode, the specified canned cycle is executed even in a no movement command block.
 0: In no movement command block, canned cycles are not executed.
- SOVC 1: The spindle override is clamped to 100% during thread cutting. (G32, G76, G92).
 0: The spindle override is not clamped to 100% even during thread cutting.
- REDT 1: The part program is registered in memory by pushing the cycle start button in the EDIT mode.
 0: It is not registered.
- IOST 1: The manual rapid traverse rate becomes effective even if reference point return is not performed when the stored stroke limit option is used.
 0: The manual rapid traverse rate does not become effective until the reference point return is performed.

	0	1	0
--	---	---	---

DGNE	SETE	DECI	SSPB	NPRD	PROD	CTHD	
7	6	5	4	3	2	1	0

- DGNE 1: Data output is effective in DIAGNOSE.
0: Data output is ineffective in DIAGNOSE.
- SETE 1: Setting is possible even when the LOCK key is locked.
0: Setting is not possible when the LOCK key is locked.
- DECI 1: Deceleration signal "1" in reference point return indicates deceleration.
0: Deceleration signal "0" in reference point return indicates deceleration.
- SSPB 1: The rotation of the spindle stops when the spindle stop input signal SSTP is 0.
0: The rotation of the spindle stops when the spindle stop input signal SSTP is 1.
- NPRD 1: Decimal point programming and the display are not used. (Not used)
0: Decimal point programming and the display are used.
- PROD 1: The position programmed is displayed in present value display.
0: The actual tool position with tool offset and tool nose radius compensation is displayed.
- CTHD 1: Continuous thread cutting is possible. (Not used)
0: Continuous thread cutting is impossible.

	0	1	2
--	---	---	---

		ZGMZ	ZGMX			ZMZ	ZMX
7	6	5	4	3	2	1	0

- ZGMX, ZGMZ X axis, Z axis reference point return method in order.
1: Magneswitch method
0: Grid method
- ZMX, ZMZ X axis, Z axis reference point return direction in order, and the initial backlash direction when turning on the power.
1: Reference point return and the backlash are performed in the negative direction.
0: Reference point return and the backlash are performed in the positive direction.
- (Note 1) For the axis functioning in reference point return, the reference point return direction and the initial backlash direction are the same. For another axis, this parameter specifies only the initial backlash direction.
- (Note 2) The backlash compensation is performed for the first time when motion along the axis moves in the direction opposite to the direction set by this parameter after the power is turned on.

	0	1	3
--	---	---	---

PSG2	PSG1		PHS				
7	6	5	4	3	2	1	0

- PSG2, PSG1 Gear ratio of spindle and position coder

Magnification	PSG2	PSG1
× 1	0	0
× 2	0	1
× 4	1	0
× 8	1	1

$$\text{Magnification} = \frac{\text{Number of spindle rotations}}{\text{Number of position coder rotations}}$$

- PHS Primitive setting of phase shift amount in DSCG method (Resolver/Inductosyn).
1: The phase shift amount is not set automatically.
0: The phase shift amount is set automatically.
Once this is done, PHS = 1 is assigned automatically.

	0	1	4
--	---	---	---

DMRX		GRDX	
7	6	5	4
3	2	1	0

	0	1	5
--	---	---	---

	DMRZ	GRDZ	
7	6	5	4
3	2	1	0

DMRX,DMRZ Command multiplying ratio for X-axis and Z-axis in order.

Setting code			Multipling ratio	
			Pulse coder	Resolver, inductosyn
0	0	0	1/2	1/8
0	0	1	1	1/4
0	1	0	1	1/4
0	1	1	2	1/2
1	0	0	3/2	3/8
1	0	1	3	3/4
1	1	0	2	1/2
1	1	1	4	1

GRDX, GRDZ Capacity of reference counter for X axis and Z axis in order.

Setting code				One cycle capacity
0	0	0	1	2000
0	0	1	0	3000
0	0	1	1	4000
0	1	0	0	5000
0	1	0	1	6000
1	0	0	1	10000

If a code not in the above table is set, capacity is set 8000.

	0	1	8
--	---	---	---

	DIC		CPF2	CPF1
	3	2	1	0

DIC 1: Input unit 1/10 (0.0001 mm, 0.00001 inch)

0: Input unit 1/1 (0.001 mm, 0.0001 inch)

CPF2, CPF1 Pulse frequency of backlash compensation (Common for all axes)

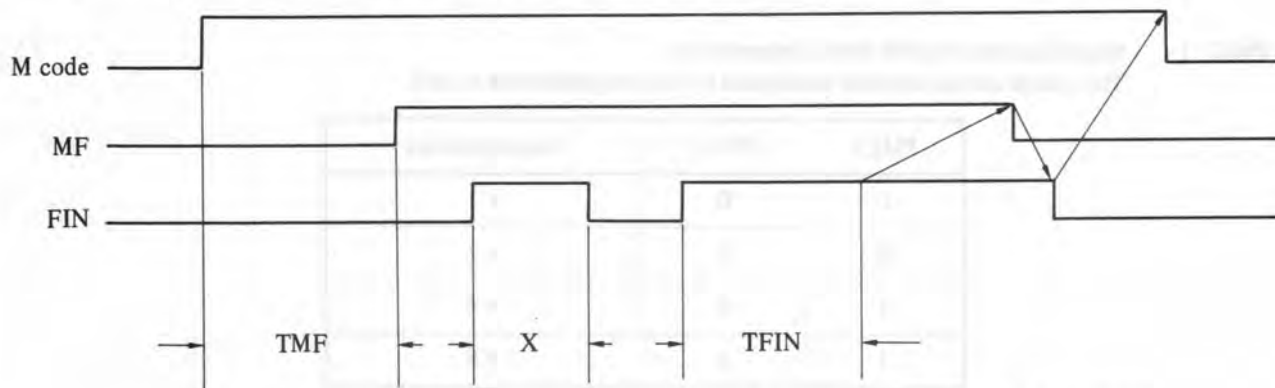
Frequency (kHz)	CPF2	CPF1
32	0	0
64	0	1
128	1	0
256	1	1

(Be sure to set the frequency to 256 Hz.)

	0	1	9
--	---	---	---

TMF				TFIN			
7	6	5	4	3	2	1	0

TMF Time from M, S, T code issuance to MF, SF, TF issuance: 16 ~ 256 msec (16 msec increments)
TFIN Time of reception width for the FIN signal: 16 ~ 256 msec (16 msec increments)



Because of $x < TFIN$, it is neglected.

TMF	TFIN	Setting			
16 msec	16 msec or more	0	0	0	0
32 msec	32 msec or more	0	0	0	1
48 msec	48 msec or more	0	0	1	0
64 msec	64 msec or more	0	0	1	1
80 msec	80 msec or more	0	1	0	0
96 msec	96 msec or more	0	1	0	1
112 msec	112 msec or more	0	1	1	0
128 msec	128 msec or more	0	1	1	1
144 msec	144 msec or more	1	0	0	0
160 msec	160 msec or more	1	0	0	1
176 msec	176 msec or more	1	0	1	0
192 msec	192 msec or more	1	0	1	1
208 msec	208 msec or more	1	1	0	0
224 msec	224 msec or more	1	1	0	1
240 msec	240 msec or more	1	1	1	0
256 msec	256 msec or more	1	1	1	1

	0	2	0
--	---	---	---

CLSI		ZTNZ	ZTNX
7		1	0

CLSI 1: Servo position LSI is not checked.
0: Servo position LSI is checked.

ZTNZ, ZTNX Reference point return function of X axis, Z axis in order.

- 1: Reference point return function is valid.
- 0: No reference point return function.

	0	2	4		PML2	PML1	DLME	RDAL				
					7	6	5	4	3	2	1	0

PML2, 1 Magnifications of pitch error compensation.
The compensation amount multiplied by this magnification is used.

PML2	PML1	Magnifications
0	0	× 1
0	1	× 2
1	0	× 4
1	1	× 8

(Common for all axes)

- DLME 1: When registering a program tape in memory, it is registered after all programs already registered are deleted.
 0: These programs are not deleted.
- RDAL 1: When a program tape is registered in memory, all programs are always registered.
 0: When a program tape is registered in memory, whether programs are registered one by one or all programs are registered depends on MDI operation.

	0	2	5		MUSR		MSUB	MPRM				TSE
					7	6	5	4	3	2	1	0

- MUSR 1: Custom macro interrupt function is used.
 0: Custom macro interrupt function is not used.
- MSUB 1: Subprogram type custom macro inerrupt.
 0: Macro type custom macro interrupt.

(Note) Values of local variables used in called macro program do not change by subprogram type custom macro interrupt. Values of local variables used in called macro program differ from those used in the current program by macro type custom macro interrupt.

- MPRM 1: Two M codes used to make custom macro interrupt effective and ineffective are set by parameters.
 0: M96 and M97 are used for custom macro interrupt.
- TSE 1: Status trigger method is used for custom macro interrupt.
 0: Edge trigger method is used for custom macro interrupt.

(Note) Macro interruption is made as long as interrupt signal is input on status trigger method.
 Macro interruption is made only when interrupt signal is tured on edge trigger method.

	0	2	6
--	---	---	---

	FHDL	NGMP	OFFVY				CKIM
7	6	5	4	3	2	1	0

- FHDL 1: The movement amount per scale of manual pulse generator is fixed at 0.01 mm/0.001 inch and is not influenced by the MP1 and MP2 signals. (Not used)
 0: The movement amount per scale of manual pulse generator varies depending on MP1 and MP2.
- NGMP The movement amount per scale of manual pulse generator is as follows:

NGMP	MP2	MP1	Movement amount
0	0	0	0.001 mm/0.0001 inch
0	0	1	0.01 mm/0.001 inch
0	1	0	0.1 mm/0.01 inch
0	1	1	0.1 mm/0.01 inch
1	0	0	0.01 mm/0.001 inch
1	0	1	0.001 mm/0.0001 inch
1	1	0	0.1 mm/0.01 inch
1	1	1	0.1 mm/0.01 inch

OFFVY 1: The servo alarm is not generated even if VRDY is on before PRDY is output.

0: The servo alarm is generated if VRDY is on before PRDY is output.

CKIM 1: Ignores the turning on or off of the machine lock signal during automatic operation (keeps the same state when cycle start was applied).

0: Makes the machine lock signal valid immediately.

(The machine lock signal is always made valid in the manual mode.)

	0	2	7
--	---	---	---

CMRX

	0	2	8
--	---	---	---

CMRZ

CMRX, CMRZ

Command multiplier for each axis.

Setting code	Multiplier
1	0.5
2	1
4	2
10	5
20	10

	0	3	1
--	---	---	---

VLOCX

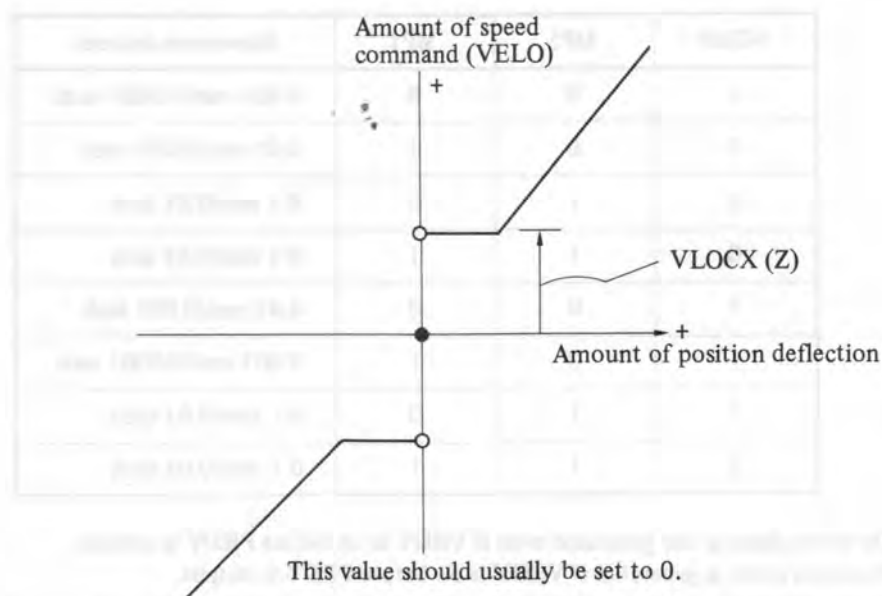
	0	3	2
--	---	---	---

VLOCZ

VLOCX, VLOCZ

Clamp of feed command value of X axis, Z axis in order.

Setting amount 0 ~ 7 VELO.



	0	3	5
--	---	---	---

MBUF1

	0	3	6
--	---	---	---

MBUF2

MBUF1, MBUF2

Up to two M codes are set. If this M code is read, a block following the M code is not read in the buffer storage.

Setting amount 00 ~ 97.

	0	3	7
--	---	---	---

SPGST

SPGST Spindle motor speed at spindle gear shift. (for constant surface speed control option)

Setting amount 0 ~ 255.

Setting value = $\frac{\text{Spindle motor speed at gear shift}}{\text{Spindle motor maximum speed}} \times 4095$

	0	3	8
--	---	---	---

SPSOR

SPSOR Spindle speed (rpm) at spindle orientation. (for constant surface speed control option)

Setting amount 0 ~ 255 (Unit: rpm)

	0	4	0
--	---	---	---

EXOMAX

EXOMAX An allowable value at limit check time in external input tool compensation B function.

Setting value 0 ~ 255 Unit: 0.002 mm (mm input)

0 ~ 255 Unit: 0.0002 inch (inch input)

	0	4	1
--	---	---	---

EXOFS1

	0	4	2	EXOFS2
	0	4	3	EXOFS3
	0	4	4	EXOFS4
	0	4	5	EXOFS5
	0	4	6	EXOFS6
	0	4	7	EXOFS7
	0	4	8	EXOFS8
	0	4	9	EXOFS9
	0	5	0	EXOFS10
	0	5	1	EXOFS11
	0	5	2	EXOFS12
	0	5	3	EXOFS13/MACINTON*

	0	5	4	EXOFS14/MACINTOF*
--	---	---	---	-------------------

	0	5	5	EXOFS15
--	---	---	---	---------

	0	5	6	EXOFS16
--	---	---	---	---------

* Either parameter is effective.

EXOFS1 ~ EXOFS16

Data multiplier of diameter checking measurement in external tool compensation B function.

Setting amount 0 ~ 15 Unit: 0.1

	0	5	3	MACINTON
--	---	---	---	----------

MACINTON: M code which makes the custom macro interrupt effective

Settings: 03 to 97

	0	5	4	MACINTOF
--	---	---	---	----------

MACINTOF: M code which makes the custom macro interrupt ineffective

Settings: 03 to 97

(Note) Parameters 053 and 054 are effective only when MPRM (309, bit 4) = 1.

	0	5	7	TMHOR (Hour)
--	---	---	---	--------------

	0	5	8	TMMIN (Minute)
--	---	---	---	----------------

	0	5	9	TMSEC (Second)
--	---	---	---	----------------

This sets and displays the run hour (Time when STL lamp is on) in units of hr, min and sec.

TMHOR Indication in hours: 0 ~ 255 (One hour increments)

TMMIN Indication in minutes: 0 ~ 59 (One minute increments)

TMSEC Indication in seconds: 0 ~ 58 (Two seconds increments)

The run hour (flickering time of STL lamp) is displayed in hours, minutes and seconds.

The run hour is held in bubble memory even if the power is cut off. But since the run hour is read into bubble memory every 6 minutes, a time less than 6 minutes is cleared when the power is turned off.

When you wish to preset the run hour to zero, follow the normal setting operation. (Run hour presetting can be executed by the setting operation.)

	0	6	2
--	---	---	---

SCTTIM

SCTTIM Delay timer when the spindle speed arrival signal is checked. (Setting of the time from an S code execution to a spindle speed arrival signal checking.)
Set value 0 ~ 255 (Unit: msec.)

Spindle speed arrival signal

S command

Delay timer

SCTTIM

Pulse distribution

	0	6	4
--	---	---	---

THDCH

THDCH Width of chamfering for thread cutting.
Setting amount: 0 ~ 31 (Unit: 0.1 pitch)
This value can also be set by the setting operation.

	0	6	5
--	---	---	---

GROVE

GROVE Return amount in G74, G75 (Radius designation)
Setting amount mm input: 0 ~ 16383 (Unit: 0.001 mm)
inch input: 0 ~ 16383 (Unit: 0.0001 inch)
This value can also be set by the setting operation.

	0	6	6
--	---	---	---

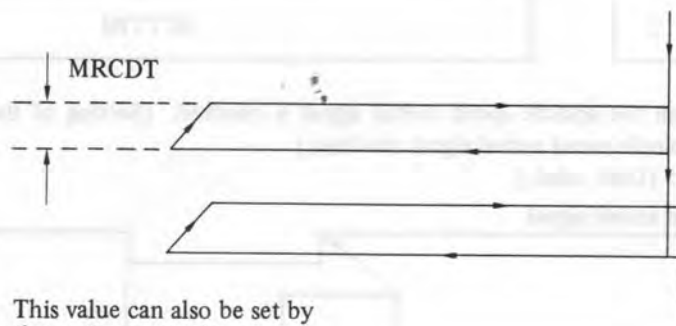
THDFN

THDFN Finishing allowance in G76. (Radius designation)
Setting amount mm input: 0 ~ 16383 (Unit: 0.001 mm)
inch input: 0 ~ 16383 (Unit: 0.0001 inch)
This value can also be set by the setting operation. (See the figure in 068-THCLM)

	0	6	7
--	---	---	---

MRCDT

MRCDT Escaping amount in multiple repetitive cycle G71, G72. (Radius designation)
Setting amount mm input: 0 ~ 16383 (Unit: 0.001 mm)
inch input: 0 ~ 16383 (Unit: 0.0001 inch)



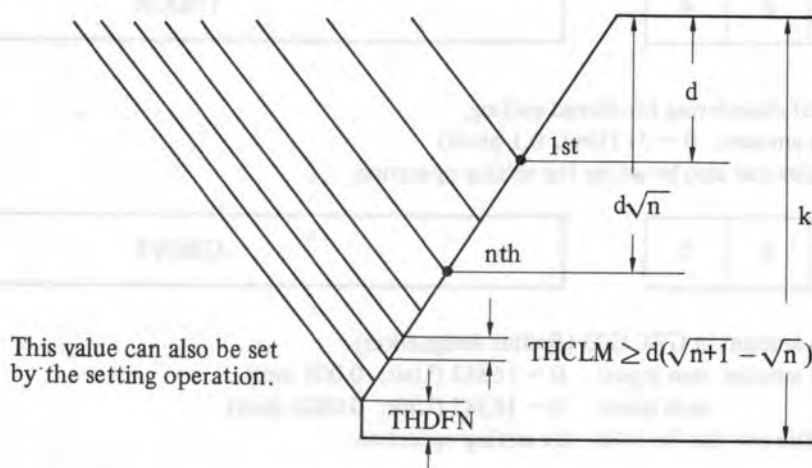
0	6	8
---	---	---

THCLM

THCLM Minimum allowance in thread cutting cycle G76. (Radius designation)

Setting amount mm input: 0 ~ 16383 (Unit: 0.001 mm)

inch input: 0 ~ 16383 (Unit: 0.0001 inch)



When $d(\sqrt{n+1} - \sqrt{n})$ becomes less than THCLM, the cutting amount is THCLM for subsequent cuttings.

0	6	9
---	---	---

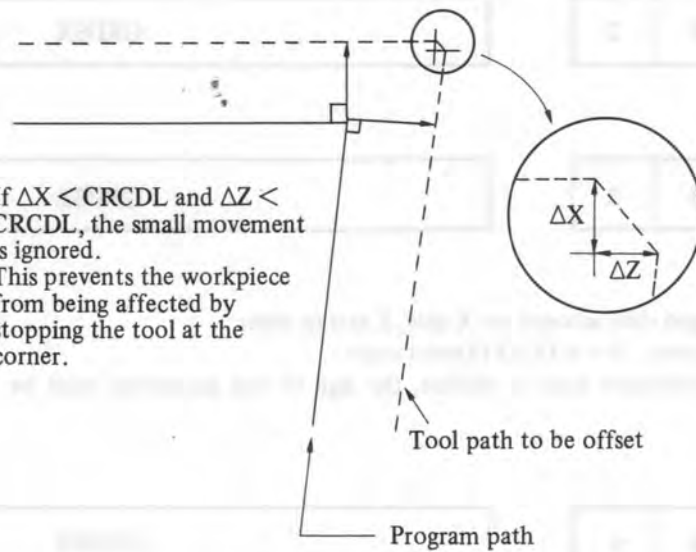
CRCDL

CRCDL When tool moves along the outside of an acute angle close to 90° during tool nose radius compensation, limitations on ignoring a small movement amount.

Setting amount mm input: 0 ~ 16383 (Unit: 0.001 mm)

inch input: 0 ~ 16383 (Unit: 0.0001 inch)

If $\Delta X < \text{CRCDL}$ and $\Delta Z < \text{CRCDL}$, the small movement is ignored.
This prevents the workpiece from being affected by stopping the tool at the corner.



	0	7	0
--	---	---	---

INPX

	0	7	1
--	---	---	---

INPZ

INPX, INPZ Width of X axis, Z axis inposition in order.
Setting amount: 0 ~ 32767 (Detect unit)

	0	7	4
--	---	---	---

STPEX

	0	7	5
--	---	---	---

STPEZ

STPEX, STPEZ

Limitation value of position deflection amount on the X axis, Z axis stoppage in order.
Setting amount: 0 ~ 32767 (Detect unit)

	0	7	8
--	---	---	---

SERRX

	0	7	9
--	---	---	---

SERRZ

SERRX, SERRZ

Limitation value of position deflection amount during X axis, Z axis movement in order.
Setting amount: 0 ~ 32767 (Detect unit)

	0	8	2
--	---	---	---

GRDSX

	0	8	3
--	---	---	---

GRDSZ

GRDSX, GRDSZ

Setting of grid shift amount for X axis, Z axis in order.

Setting amount: 0 ~ ±32768 (Detect unit)

When the reference point is shifted, the sign of this parameter must be specified according to the direction.

	0	8	6
--	---	---	---

LPGMX

	0	8	7
--	---	---	---

LPGMZ

LPGMX, LPGMZ

Setting of servo loop gain multiplier of X axis, Z axis.

Setting amount = $2048 \times \frac{E}{L} \times \alpha \times 1000$

$E = \begin{cases} 7 \text{ [v]} & (\text{For motor with 7V and 1000 rpm}) \\ 3.5 \text{ [v]} & (\text{For motor with 7V and 2000 rpm}) \end{cases}$

L Machine movement amount per motor revolution (mm or inch)

α Detect unit (mm or inch)

(Example) 2 mm per motor revolution at 1000 rpm/7V

Setting value: $2048 \times \frac{7}{2} \times 0.0005 \times 1000 = 3584$

(At detect unit 0.0005 mm)

	0	9	0
--	---	---	---

LPGIN

LPGIN

Setting of position control loop gain

Setting amount: 1 ~ 9999 (Unit: 0.01 sec⁻¹)

	0	9	1
--	---	---	---

JOGF

JOGF

JOG feedrate at rotary switch position 10.

Setting amount: 1 ~ 150 Unit: mm/min (mm output)

1 ~ 60 Unit: 0.1 inch/min (inch output)

	0	9	2
--	---	---	---

RPDFX

	0	9	3
--	---	---	---

RPDFZ

RPDFX, RPDFZ

Rapid traverse rate of X axis, Z axis in order.

Setting amount: 30 ~ 15000 Unit: mm/min (mm output)

30 ~ 6000 Unit: 0.1 inch/min (inch output)

	0	9	6
--	---	---	---

LINTX

	0	9	7
--	---	---	---

LINTZ

LINTX, LINTZ

Time constant of linear acceleration/deceleration for X axis, Z axis. (for rapid traverse rate)

Setting amount: 8 ~ 4000 (Unit: msec.)

	1	0	0
--	---	---	---

EXPTX

	1	0	1
--	---	---	---

EXPTZ

EXPTX, Z Time constant of exponential acceleration/deceleration in manual feeding for X axis and Z axis in order.

Setting amount: 8 ~ 4000 Unit: msec.

	1	0	4
--	---	---	---

THRDT

THRDT Time constant of exponential acceleration/deceleration of X axis in thread cutting (G92)

Setting amount: 1 ~ 4000. (Unit: msec.)

The most appropriate value should be set combined with the THDFL parameter (parameter number 107).

	1	0	5
--	---	---	---

FEEDT

FEEDT Time constant of exponential acceleration/deceleration in feed.

Setting amount: 8 ~ 4000 Unit: msec.

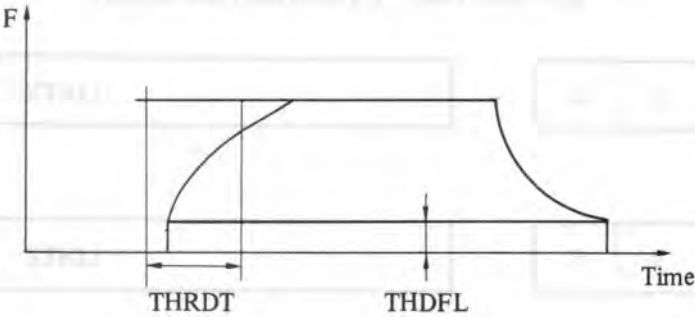
	1	0	6
--	---	---	---

FEDMX

FEDMX Upper speed of cutting feed (Available for all axes)
 Setting amount: 6 ~ 15000 Unit: mm/min (metric output)
 6 ~ 6000 Unit: 0.1 inch/min (inch output)

	1	0	7	THDFL
--	---	---	---	-------

THDFL Lower speed (FL) of X axis exponential acceleration/deceleration in thread cutting (G92)
 Setting amount: 6 ~ 15000 Unit: mm/min (metric output)
 6 ~ 6000 Unit: 0.1 inch/min (inch output)
 The most appropriate value should be set combined with the THRDT parameter (parameter number 104).



	1	0	8	FEDFL
--	---	---	---	-------

FEDFL The lower limit speed (FL) on exponential acceleration/deceleration in feeding.
 Setting amount: 6 ~ 15000 Unit: mm/min (mm output)
 6 ~ 6000 Unit: 0.1 inch/min (inch output)
 Normally, this amount is set to "0".

	1	0	9	JGFLX
--	---	---	---	-------

	1	1	0	JGFLZ
--	---	---	---	-------

JGFLX, JGFLZ The lower limit speed (FL) on exponential acceleration/deceleration in manual feeding along X axis and Z axis respectively.
 Setting amount: 6 ~ 15000 Unit: mm/min (mm output)
 6 ~ 6000 Unit: 0.1 inch/min (inch output)

	1	1	3	RPDFL
--	---	---	---	-------

RPDFL Least speed of rapid traverse override (Fo) (Common to all axes)
 Setting amount: 6 ~ 15000 Unit: mm/min (metric output)
 6 ~ 6000 Unit: 0.1 inch/min (inch output)

	1	1	4	ZRNFL
--	---	---	---	-------

ZRNFL Low speed for reference point return (FL) (Common to all axes)
 Setting amount: 6 ~ 15000 Unit: mm/min (metric output)
 6 ~ 6000 Unit: 0.1 inch/min (inch output)

	1	1	5
--	---	---	---

BKLX

	1	1	6
--	---	---	---

BKLZ

BKLX, BKLZ

Backlash amount for X axis, Z axis respectively.

Setting amount: 0 ~ 255 Unit: 0.001 mm (mm output)

0 ~ 255 Unit: 0.0001 inch (inch output)

The X axis value should be set with a diameter value when the diameter designation has been selected.

	1	1	9
--	---	---	---

SPDLC

SPDLC Set the spindle speed offset compensation value, that is, compensation value of zero offset of spindle speed command voltage.

Setting amount 0 ~ ±8191 (For optional constant surface speed control B)

	1	2	0
--	---	---	---

GRMX1

	1	2	1
--	---	---	---

GRMX2

	1	2	2
--	---	---	---

GRMX3

	1	2	3
--	---	---	---

GRMX4

GRMX1 ~ GRMX4

The spindle speed corresponding to gears 1 ~ 4 when the spindle speed command is 10V. (For optional constant surface speed control)

Setting amount: 1 ~ 32767 (Unit: RPM)

	1	2	4
--	---	---	---

DRFTX

	1	2	5
--	---	---	---

DRFTZ

DRFTX, DRFTZ

Compensation amount of drift generated in servo loop for X axis, Z axis.

Setting amount: 0 ~ ±500.

When automatic drift compensation parameter (006-ADFT) is set, this amount is changed automatically.

	1	2	8
--	---	---	---

PHAZX

	1	2	9
--	---	---	---

PHAZZ

PHAZX, PHAZZ

Amount of servo phase shift along X axis, Z axis, respectively.

Data corresponding with the phase of the signal returning from feedback is automatically set (for resolver and inductosyn)

Setting amount: 0 ~ 500

	1	3	2
--	---	---	---

LOWSP

LOWSP

Least spindle speed in constant surface speed control mode (G96) (for optional constant surface speed control)

Setting amount: 0 ~ 9999 (Unit: rpm)

Only effective when parameter 307-SLOW is 1.

	1	3	3
--	---	---	---

ACALFL

ACALFL

Feedrate during measuring in automatic tool compensation function (Common for all axes)

Setting amount: 6 ~ 15000 Unit: mm/min (mm output)

6 ~ 6000 Unit: 0.1 inch/min (inch output)

	1	3	4
--	---	---	---

SPDMAX

Allowable maximum spindle speed

Setting amount: 0 ~ 9999 (Unit: rpm)

(Note) No clamp is made if 0 is set.

	1	4	0
--	---	---	---

PSANGN

PSANGN

The data for the adjustment of gain of constant surface speed control B (analogue output)

Setting range: 700 ~ 1250

Standard setting value: 1000

(Procedure)

- (1) Set standard setting value 1000
- (2) Specify the max. value (10V)
- (3) Measure the output voltage
- (4) Reset the PSANGN according to the following:

$$\text{Setting value} = \frac{10.0}{\text{Measured voltage (V)}} \times 1000$$

- (5) Specify the max. value and confirm that the output voltage is 10V.

	1	4	1
--	---	---	---

					TIME1				
--	--	--	--	--	-------	--	--	--	--

TIME1 Run hour.
Setting amount: 0 ~ 32767 (Unit: 0.1 hr.)
This value can also be set by the setting operation.

	1	4	2
--	---	---	---

					TIME2				
--	--	--	--	--	-------	--	--	--	--

TIME2 Run hour
Setting amount: 0 ~ 99999999 (Unit: 0.1 hr.)

	1	4	3
--	---	---	---

					LT1X1				
--	--	--	--	--	-------	--	--	--	--

	1	4	4
--	---	---	---

					LT1Z1				
--	--	--	--	--	-------	--	--	--	--

	1	4	7
--	---	---	---

					LT1X2				
--	--	--	--	--	-------	--	--	--	--

	1	4	8
--	---	---	---

					LT1Z2				
--	--	--	--	--	-------	--	--	--	--

	1	5	1
--	---	---	---

					LT2X1				
--	--	--	--	--	-------	--	--	--	--

	1	5	2
--	---	---	---

					LT2Z1				
--	--	--	--	--	-------	--	--	--	--

	1	5	3
--	---	---	---

					LT2X2				
--	--	--	--	--	-------	--	--	--	--

	1	5	4
--	---	---	---

					LT2Z2				
--	--	--	--	--	-------	--	--	--	--

	1	5	5
--	---	---	---

					LT3X1				
--	--	--	--	--	-------	--	--	--	--

	1	5	6
--	---	---	---

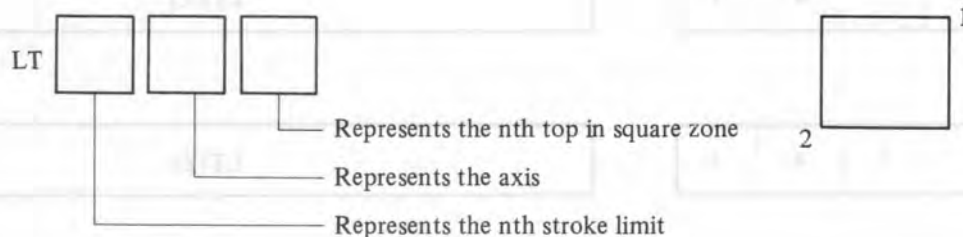
LT3Z1	
-------	--

	1	5	7
--	---	---	---

LT3X2	
-------	--

	1	5	8
--	---	---	---

LT3Z2	
-------	--



Set stroke limit mentioned above.

Setting amount: 0 ~ ±99999999 Unit: 0.001 mm (mm output)

0 ~ ±99999999 Unit: 0.0001 inch (inch output)

The setting operation can be used for parameters 151 to 158.

In diameter programming, the X axis value should be set with a diameter value.

	1	5	9
--	---	---	---

REF2X	
-------	--

	1	6	0
--	---	---	---

REF2Z	
-------	--

REF2X, REF2Z

Distance from the second reference point to the first reference point of X axis, Z axis.

Setting amount: 0 ~ ±99999999 Unit: 0.001 mm (mm output)

0 ~ ±99999999 Unit: 0.0001 inch (inch output)

In diameter programming, the X axis value should be set with a diameter value.

	1	6	3
--	---	---	---

GANMAX	
--------	--

	1	6	4
--	---	---	---

GANMAZ	
--------	--

GANMAX, GANMAZ

The amount of γ on the X axis and Z axis in order in the automatic tool compensation function.

Setting amount: 1 ~ 99999999 Unit: 0.001 mm (mm output)

1 ~ 99999999 Unit: 0.0001 inch (inch output)

The X axis value should be set with a radius value.

	1	6	5
--	---	---	---

	EPCX
--	------

	1	6	6
--	---	---	---

	EPCZ
--	------

EPCX, EPCZ

The amount of ϵ on the X axis and Z axis in order in the automatic tool compensation function.

Setting amount: 1 ~ 99999999 Unit: 0.001 mm (mm output)

1 ~ 99999999 Unit: 0.0001 inch (inch output)

The X axis value should be set with a radius value.

	3	0	6
--	---	---	---

SKPF				NEOP		TMCR	OFM
7	6	5	4	3	2	1	0

SKPF 1: When a skip cutting command (G31) is executed, the feedrate follows the FL speed set by parameter No. 342.

0: When executing a skip cutting command (G31), the feedrate follows the F code.

NEOP 1: When registering a tape in memory, M02, M30 or M99 is not counted as the program end.

0: When registering a tape in memory, M02, M30 or M99 is counted as the program end.

TMCR 1: The custom macro is called by the T code.

0: The custom macro is not called by the T code.

OFM 1: Tool offset is effected in a block with a move command.

0: Tool offset is effected even if the block does not have move command.

	3	0	7
--	---	---	---

FCUT	ABIC	SLOW					OTCS
------	------	------	--	--	--	--	------

FCUT 1: In a multiple repetitive cycle (G71, G72), finishing after the final rough cutting is not executed.

0: In a multiple repetitive cycle (G71, G72), finishing after the final rough cutting is executed.

ABIC 1: ABC or INC key on the MDI & CRT is effective for data of X or Z axis with special G code B and C.

0: ABC and INC keys are ineffective for X and Z-axis data with special G code B and C.

SLOW 1: This sets the clamp value of the lowest speed in constant surface speed control A/B for all gears in common. (parameter 132)

0: This sets the clamp value of the lowest speed in constant surface speed control A/B individually for each gear. (parameter 343, 344, 345, 346).

OTCS 1: When a movable part of a machine tool touches a mechanical limit switch, the machine stops instantaneously. (The machine position is cleared.)

0: When a movable part of machine tool touches a mechanical limit switch, the machine stops after deceleration. (The machine position is not cleared.)

	3	0	8
--	---	---	---

DIOM	MSFT		MANP	RSTB			
7	6	5	4	3	2	1	0

DIOM 1: It is possible to read and write DI, DO using a macro variable. (Not used)

0: It is not possible to read and write DI, DO using a macro variable.

MSFT 1: When the user macro option is used, the shift key is valid for key-input from the MDI.

0: The shift key is invalid for key-input from the MDI.

MANP 1: When there is no decimal point in an argument of a user macro, the argument is regarded as an integer. (Not used)

0: When there is no decimal point in an argument of a user macro, the argument is regarded as a value with decimal point according to conversion.

RSTB 1: On resetting by an emergency stop, external reset, or reset & rewind, no resetting signal is issued during the reset.

0: On resetting by an emergency stop, external reset or reset & rewind, a resetting signal is issued during the reset.

	3	0	9
--	---	---	---

TLSK	GRST		LCTM			ATPZ	ATPX
7	6	5	4	3	2	1	0

TLSK 1: Inputs group No. for tool skip

0: Inputs no group No.

GRST 1: Clears executed data of all registered groups when the Reset signal is input.

0: Clears executed data of only groups at the life end when the Reset signal is input.

LCTM 1: Specifies tool life by time.

0: Specifies tool life by frequency.

ATPX, ATPZ

Automatic coordinate system setting (option) for X axis and Z axis respectively is effective or ineffective.

1: Automatic coordinate system setting is effective.

0: Automatic coordinate system setting is ineffective.

See the explanation for parameters 375 and 376.

	3	1	0
--	---	---	---

		RSCB1	STP21		BAD1
--	--	-------	-------	--	------

	3	1	1
--	---	---	---

		RSCB2	STP22		BAD2
--	--	-------	-------	--	------

	3	1	2
--	---	---	---

		RSCB3	STP23		BAD3
--	--	-------	-------	--	------

	3	1	3
--	---	---	---

		RSTB4	STP24	BAD4			
7	6	5	4	3	2	1	0

RSCB1, 2, 3, 4

Specifies whether or not control codes (DC1 ~ DC4) are used on I/O devices 1, 2, 3, and 4 respectively.

1: The control codes are not used.

0: The control codes are used.

STP21, 2, 3, 4

In order of I/O devices 1, 2, 3 and 4, specifies whether the stop bit is to be 2 bits or 1 bit.

1: The stop bit is to be 2 bits.

0: The stop bit is to be 1 bit.

BAD1, 2, 3, 4

In order of I/O devices 1, 2, 3 and 4, specifies a baud rate.

(Note) Set the baud rate of data transmission between the NC and robot to BAD4 when a robot is used.

Baud rate	BAD1, 2, 3, 4			
50	0	0	0	0
100	0	0	0	1
110	0	0	1	0
150	0	0	1	1
200	0	1	0	0
300	0	1	0	1

Baud rate	BAD1, 2, 3, 4			
600	0	1	1	0
1200	0	1	1	1
2400	1	0	0	0
4800	1	0	0	1
9600	1	0	1	0

(Note) See the explanation for parameters 340 and 341.

	3	1	4
--	---	---	---

	MINT						HZ	HX
7	6	5	4	3	2	1	0	

- MINT 1: Assigned NC statement is executed after the current block is completed. (Custom macro interrupt type II)
0: Assigned NC statement is executed immediately. (Custom macro interrupt type I)
HX, HZ Valid/invalid settings of handle in X and Z axis.
1: Valid.
0: Invalid.

	3	1	5
--	---	---	---

	FCSS							
7	6	5	4	3	2	1	0	

- FCSS 1: S analog voltage output is changed at intervals of 8 ms (new specification).
0: S analog voltage output is changed at intervals of 64 ms (old specification).

	3	1	6
--	---	---	---

CDSCG	PCFBK	ACMR					DSCGZ	DSCGX
7	6	5	4	3	2	1	0	

- CDSCB 1: Specifies that a frequency check of DSCG feedback (resolver, inductosyn) not be carried out.
0: Specifies that a frequency check of DSCG feedback (resolver, inductosyn) be carried out.
(After field adjustment, this parameter should be always set to "0".)
PCFBK 1: Servo feedback check is performed.
0: Servo feedback check is not performed.
ACMR 1: Setting of the special CMR (Not used)
CMR set in parameter 027 and 028 is special.
0: Setting of the standard CMR
CMR set in parameter 027 and 028 is standard.
DSCGX, Z This specifies the type of position detection system for the X axis and Z axis respectively.
1: Resolver or inductosyn
0: Pulse coder

	3	1	7
--	---	---	---

UM#8	UM#7	UM#6	UM#5	UM#4	UM#3	UM#2	UM#1
7	6	5	4	3	2	1	0

- UM#1 ~ 8 For EIA code, this registers a code corresponding to "#" used in custom macro.
Example: UM#8 ~ UM#1 = 01001001
Code that has a hole in channels 1, 4, and 7 is counted as "#" in EIA code.
(UM#8 ~ UM#1 = 00000000 indicates that "#" is not used.)

	3	1	8
--	---	---	---

PRG9	MSC9	MPD9			NSRH	RSTL	
7	6	5	4	3	2	1	0

- PRG9 1: Programs with numbers 9000 ~ 9899 can not be edited.
0: Programs with numbers 9000 ~ 9899 can be edited.
- MSC9 1: If in the single block mode during execution of a program with number 9000 ~ 9899, single block stop is effective in executing macro statements of user macro.
0: If in the single block mode during execution of a program with number 9000 ~ 9899, single block stop is ineffective in executing macro statements of user macro.
- MPD9 1: The contents of the program are not displayed during execution of programs with numbers 9000 ~ 9899.
0: The contents of the program is displayed during execution of programs with numbers 9000 ~ 9899.
- NSRH 1: Signal OP is not issued during sequence number search.
0: Signal OP is issued during sequence number search.
- RSTL 1: When a program is loaded in memory by a cycle start in the edit mode, signal STL is not issued.
0: Signal STL is issued.

	3	1	9
--	---	---	---

PRG8	MCS8	MPD8					MCS7
7	6	5	4	3	2	1	0

This can also be set by the setting operation.

- PRG8 1: Programs with numbers 8000 ~ 8999 can not be edited.
0: Programs with numbers 8000 ~ 8999 can also be edited.
- MCS8 1: If in the single block mode during execution of a program with number 8000 ~ 8999, single block stop is effective in executing macro statements of user macro.
0: Even if in the single block mode during execution of a program with number 8000 ~ 8999, single block stop is ineffective in executing macro statements of user macro.
- MPD8 1: The contents of the program are not displayed during execution of programs with numbers 8000 ~ 8999.
0: The contents of the program are displayed during execution of programs with numbers 8000 ~ 8999.
- MCS7 1: If in the single block mode during execution of a program with number 0001 ~ 7999, single block stop is effective in executing macro statements of user macro.
0: Even if in the single block mode during execution of a program with number 0001 ~ 7999, single block stop is ineffective in executing macro statements of user macro.

	3	2	0
--	---	---	---

UMMCD1

	3	2	1
--	---	---	---

UMMCD2

	3	2	2
--	---	---	---

UMMCD3

UMMCD1, 2, 3

This sets up to 3 M codes used to call user macros.

Setting amount: 01 ~ 97

(When M00 is specified, the user macro cannot be called. When 00 is set, it is equivalent to no setting.)

	3	2	3
UMGCD0			
	3	2	4
UMGCD1			
	3	2	5
UMGCD2			
	3	2	6
UMGCD3			
	3	2	7
UMGCD4			
	3	2	8
UMGCD5			
	3	2	9
UMGCD6			
	3	3	0
UMGCD7			
	3	3	1
UMGCD8			
	3	3	2
UMGCD9			

UMGCD0, 1, ..., 9

This sets up to 10 G codes used to call custom macros.

Setting amount: 001 ~ 255

(When G00 is specified, the custom macro cannot be called. When 00 is set, it is equivalent to no setting.)

	3	3	6
--	---	---	---

PEXZRZ			
--------	--	--	--

	3	3	7
--	---	---	---

PECZRZ			
--------	--	--	--

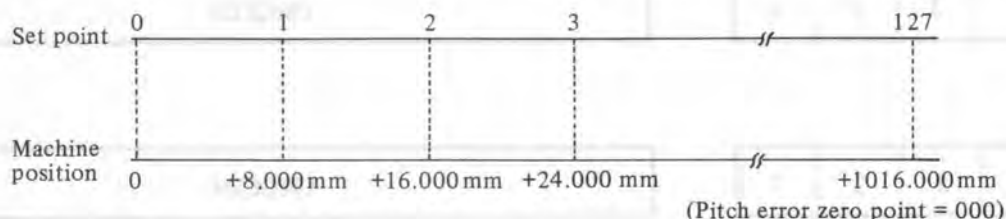
PECZRZ, Z

Pitch error zero point on X and Z axes in order.

Setting amount: 0 ~ 127

Set a point which corresponds to the reference point.

For example, if point 0 is specified, point 1 is at +8.000 mm position and point 127 is at +1016.000 mm. And compensation is made over the range from 0 to 1016.000 mm.



(Pitch error compensation interval (parameter 379,380) is 8000 in this example.)

	3	4	0
--	---	---	---

IDVICE			
--------	--	--	--

	3	4	1
--	---	---	---

ODVICE			
--------	--	--	--

IDVICE This selects an input device when registering a program in memory. (When INPUT DEVICE 2 = 1 (RS232C) has been set, this setting becomes valid.)

ODVICE This selects an output device for punching.

Set value	I/O device
0	On input, tape reader On output, FACIT PUNCHER
1	In both input/output, ASR33/ASR43. Parameters such as a baud rate, etc. should be set in parameter No. 310.
2	In both input/output, RS232C. Parameters such as a baud rate, etc. should be set in parameter No. 311.
3	In both input/output, RS232C. Parameters such as a baud rate, etc. should be set in parameter No. 312.
4	In both input/output, RS232C. Parameters such as a baud rate, etc. should be set in parameter No. 313.

The above parameters can also be set by the setting operation.

	3	4	2	PSKPFL
--	---	---	---	--------

PSKPFL FL speed of skip cutting (common to all axes)
 Setting amount: 6 ~ 15000 Unit: 1mm/min (mm output)
 6 ~ 6000 Unit: 0.1 inch/mm (inch output)

	3	4	3	GRMIN1
--	---	---	---	--------

	3	4	4	GRMIN2
--	---	---	---	--------

	3	4	5	GRMIN3
--	---	---	---	--------

	3	4	6	GRMIN4
--	---	---	---	--------

GRMIN1, 2, 3, 4
 The lowest spindle speed in a constant surface speed control for gears 1, 2, 3, 4 respectively.
 (for constant surface speed control option)
 Setting amount: 0 ~ 9999 Unit: rpm
 This parameter is valid only when parameter 307-SLOW = 0.

	3	6	3	PCFBKX
--	---	---	---	--------

	3	6	4	PCFBKZ
--	---	---	---	--------

PCFBKX/Z Amount of movement when servo feedback check is performed in X/Z, respectively
 Settings: 0 to $\pm 32,767$; detection unit
 (Note 1) When the setting is 0, servo feedback check is not performed.
 (Note 2) The amount of movement for servo feedback check is fixed at 0.255 mm (0.0255 inch).
 (Note 3) The setting is a value whereby the machine can be moved. Therefore, use a small value as much as possible. However, note that setting too small value will activate the alarm.

	3	7	5
--	---	---	---

PPRTMX			
--------	--	--	--

	3	7	6
--	---	---	---

PPRTMZ			
--------	--	--	--

PPRTMX, Z The coordinates for automatic coordinate system setting in mm input for X axis and Z axis respectively. This sets the distance from the zero point of the coordinate system to be set to the 1st reference point in the mm system.

Setting amount: 0 ~ 99999999 Unit: 0.001 mm

When the inch/metric conversion option is used, parameter Nos. 379 and 380 should be also set.

Only the axis that is specified to be effective for automatic coordinate system setting in parameter 309 is effective.

	3	7	9
--	---	---	---

PPRTIX			
--------	--	--	--

	3	8	0
--	---	---	---

PPRTIZ			
--------	--	--	--

PPRTIX, Z The coordinates for automatic coordinate system setting for inch for X axis and Z axis respectively.

This sets the distance from the zero point of the coordinate system to be set to the 1st reference point in the inch system.

Setting amount: 0 ~ 99999999 Unit: 0.0001 inch

	3	8	7
--	---	---	---

Secret number			
---------------	--	--	--

Register a secret number to be locked in advance.

Setting range: 1 to 99,999,999

	4	0	8
--	---	---	---

Lock/Unlock			
-------------	--	--	--

Entering the same number as that of parameter 387 effects unlocking. Entering a different number effects locking.

(Note 1) Whenever parameter 387 is zero, an unlocked state is provided. Turning off/on the NC power does not effect locking. Note that any number other than zero should not be set in parameter for programs not to be locked.

When the inch/metric conversion option is used, parameter Nos. 375 and 376 should also be set.

Only the axis that is specified to be effective for automatic coordinate system setting in parameter 309 is effective.

The list of parameters which can be set by the setting operation (Function button **SET**) is as follows:

Data number	Content
000	Setting on I/O (RS232C, RMT, INCH, ISO, TVON)
057*	Run hour (TMHOR) (Unit: 1 hour)
058*	Run hour (TMMIN) (Unit: 1 minute)
059*	Run hour (TMSEC) (Unit: 2 seconds)
064*	Chamfering width of thread (THDCH)
065*	Return amount in G74, G75 (GROVE)
066*	Finishing allowance in G76 (THDFN)
067*	Retracting amount in G71, G72 (MRCDT)
068*	Minimum allowance in G76 (THCLM)
141*	Run hour (TIME1) (Unit: 0.1 hour)
151*	1st-vertex X-axis value of 2nd stored stroke limit
152*	1st-vertex Z-axis value of 2nd stored stroke limit
153*	2nd-vertex X-axis value of 2nd stored stroke limit
154*	2nd-vertex Z-axis value of 2nd stored stroke limit
155*	1st-vertex X-axis value of 3rd stored stroke limit
156*	1st-vertex Z-axis value of 3rd stored stroke limit
157*	2nd-vertex X-axis value of 3rd stored stroke limit
158*	2nd-vertex Z-axis value of 3rd stored stroke limit
319*	Various settings (PRG8, MSBL)
340*	Selection of input device when storing the program
341*	Selection of output device when punching out

- Select the function key SET.
- Blank is displayed in Data No. except as mentioned above.
- A Data No. with * is set to the same Data No. in function key PRARAM.
- See the explanation of parameters with the same Data No.

APPENDIX 7 ALARM LIST

Number	Content	Remarks
000	Re-apply the power after the parameter was input. (Parameter numbers 012 ~ 015, 018, 027, 028, 031, 032, 082, 083, 086, 087, 090, 124, 125, 128, 129, 316)	
001	TH alarm (a character with incorrect parity was input in the significant information zone). Correct the tape.	
002	TV alarm (the number of characters in a block is odd). This alarm will be generated only when the TV check is effective. Correct the tape.	
003	Data exceeding the maximum allowable number of digits was input. (See the Section on max. programmable dimensions.)	
004	A numeral, the sign (–) or a decimal point was input without an address at the beginning of a block. (See section III. 10, 11 if user macro option is equipped).	
005	The address was not followed by appropriate data but was followed by another address or EOB code.	
006	Sign “–” input error. (Sign “–” was input after an address with which it can’t be used. Or two or more “–” signs were input.)	
007	Decimal point “.” input error. (A decimal point was input after an address with which it can’t be used. Or two or more decimal points were input.)	
008	The tape reader control switch was set to other than AUTO (without reel) or REEL ON, OFF (with reel).	
009	An invalid character was input in the significant information zone (B, C, Y, V, J or H)	
010	An invalid G code was specified. (This alarm is also generated when a G code that cannot be used by the controller is specified.)	
011	The feedrate was not specified for cutting feed or the feedrate was inadequate.	
012	An E code was specified for an NC which does not have the E code option.	
014	The lead increment/decrement value indicated by address K exceeded the Max. command value or a negative value was specified in variable lead thread cutting.	
022	In a circular interpolation command, radius designation was performed for an NC which is not equipped with the radius designation option.	
023	In a circular interpolation command by radius designation, zero or a negative value was specified for address R.	
029	An offset value exceeded 6 digits. The offset value should be reset.	
030	The tool offset number was too large for the T function.	
031	In an offset value input command (by G10 or by custom macro input command), the numerical value under the address P specifying an offset number was too large or P was missing.	
032	In an offset value input command (by G10 or by custom macro input command), the specified offset value was too large.	
033	A point of intersection can not be determined for tool nose radius compensation.	

Number	Content	Remarks
034	The start up or cancel was going to be performed in the G02 or G03 mode in tool nose radius compensation.	
035	Skip cutting (G31) was specified in tool nose radius compensation mode.	
038	Overcutting will occur in tool nose radius compensation because the arc start point or end point coincides with the arc center.	
039	Chamfering or corner R was specified with a start-up, a cancel, or switching between G41 and G42 in tool nose radius compensation. The program may cause overcutting to occur in chamfering or corner R.	
040	Overcutting will occur in tool nose radius compensation in a canned cycle G90 or G94.	
041	Overcutting will occur in tool nose radius compensation.	
047	G code G27 to G30 was specified for an axis which has no reference point.	
048	G30 was specified without performing reference point return after the power was turned on or an emergency stop was executed. A move command was executed without performing reference point return after the power was turned on or after an emergency stop was executed in an NC which includes optional stored stroke limit.	
050	The chamfering or a corner R was specified in a block which includes a thread cutting command.	
051	The block after a block containing a chamfering or a corner R specification was not a G01 command.	
052	The move direction or the move amount in a block following chamfering or a corner R command was not adequate.	
053	Two or more I, K and R values were specified in a block in the chamfering or corner R command.	
054	A block in which the chamfering or the corner R was specified includes a taper command.	
055	The move distance in the block which includes the chamfering or the corner R specification is smaller than the chamfering distance or the corner R.	
056	The addresses X (or Z) and I (or K) were specified together in a block which includes the chamfering command. G01 X (U) _____ I _____ ; or G01 Z (W) _____ K _____ ;	
059	The selected workpiece number program cannot be found. (External workpiece number selection function A)	
060	The specified sequence number was not found in the sequence number search.	
061	Either address P or Q was not specified in a block including a G70, G71, G72 or G73 command.	

Number	Content	Remarks
062	○ The number following address D_j in a block which includes a G71, G72 or G73 command was not positive. ○ The number following address K or D in a block which includes a G76 command was not positive. The number following address A in a block which includes a G76 command was not valid. ○ The number following address I, K or D in a block which includes a G74 or a G75 command was negative. Address X was specified in spite of the fact that the numeral following address I was zero or address Z was specified in spite of the fact that the numeral following address K was zero in a block which includes a G74 or G75 command.	
063	The sequence number specified by address P or Q in a block which includes a G70, G71, G72 or G73 command was not found.	
065	○ The block which was specified by address P in a block which includes a G71, G72 or G73 command does not include a G00 or G01 command. ○ In G71 and G72, Z (W) has been specified (G71) or X (U) has been specified (G72) in the block with the sequence No. specified by P.	
066	A G code other than G00, G01, G02, G03 or G04 was specified between the two blocks specified by addresses P and Q respectively in a block which includes the G70, G71, or G72 command.	
067	Either G70, G71, G72 or G73 was specified in the tape mode or the MDI mode. (G70, G71, G72 or the G73 command is effective only in MEMORY mode.)	
068	The number of pockets exceeded 10 in G71 or G72 of type II.	
069	The final move command in the blocks inserted between the two blocks specified by P and Q of G71, G72 and G73 ended with chamfering or corner R.	
070	The memory area is insufficient.	
071	The address searched for was not found.	
072	The number of programs to be stored exceeds 95 or 191.	
073	The program number has already been used.	
074	The program number is other than 1 ~ 9999.	
075	The program number and sequence number were not found in the first block of the program.	
076	Address P was not specified in a block which includes a M98, G65 or G66 command.	
077	The subprogram was called 3 times (5 times with user macro option).	
078	The program number (specified program number when called by G, M or T) or the sequence number which was specified by address P in a block which includes an M98, M99, G65, G66 and call by G, M or T was not found. Or the sequence number specified in a GO TO statement was not found.	
079	Stored program and tape contents do not coincide. (Collating program)	
080	In the area specified by parameter ϵ , the approach end signal does not come on. (Automatic tool compensation function)	
081	Automatic tool compensation was specified without a T code. (Automatic tool compensation function)	

Number	Content	Remarks
082	T code and Automatic tool compensation were specified in the same block. (Automatic tool compensation*function)	
083	In Automatic tool compensation, an invalid axis was specified or the command is incremental. (Automatic tool compensation function.)	
085	In reading by the RS232C interface, the number of bits of input data does not match or the baud rate is not correct.	
086	In reading or in output by the RS232C interface, a transmission or I/O device problem occurred.	
087	In reading by the RS232C interface, after DC3 (stop code) was transmitted, data over 10 characters long was input.	
088	The data error signal (DERR) was input. (External input tool compensation function A)	
089	BCD data is not 0 ~ 9. (External input tool compensation function A)	
090	The grid method reference point return cannot be performed normally because one-revolution signal (or reference zero point signal for the linear scale) is not input from the pulse coder.	
091	The grid method reference point return cannot be executed normally, because the feedrate is too low to synchronize the one-revolution signal (or reference zero point signal for the linear scale) of the pulse coder with the reference counter.	
092	Motion along the axis specified in G27 has not yet returned to the reference point.	
094	No P type can be specified for program restart (because after program interruption, programming of absolute zero point, ORIGIN, etc. have been operated for).	
096	No P type can be specified for program restart (because after program interruption, the coordinate shift value has changed).	
097	No P type can be specified for program restart (because after the power was turned on or after an emergency stop or stroke limit alarm (instantaneous stop) was released, automatic operation was not executed).	
098	When program restart was specified with no reference point return after the power was turned on or after the emergency stop or stroke limit alarm was released, G28 was found during block search by a program restart command.	
099	A move command was performed in the MDI mode after the search for program restart.	
100	The parameter enable switch is on. Push the reset button after turning off the switch.	
101	The power was turned off while rewriting the contents of memory in a part program storage & editing operation. When this alarm is generated, you must turn on the power while pushing the DELET and RESET buttons to clear the memory.	
110	The absolute value of data of fixed point representation was not within the valid range.	
111	The exponent of data of floating point representation exceeds the upper bound.	

Number	Content	Remarks
112	Division by zero was specified.	
113	A function that cannot be used by custom macro A was used.	
114	A format error (except in [Format]) occurred.	
115	A value not defined as a variable number was assigned.	
116	The left side of a substituted sentence was invalid.	
118	Nesting of brackets exceeds the upper limit (5).	
119	The argument of SQRT is negative. The argument in BCD is a negative value or is composed of data other than 0 to 9.	
122	Macro nesting is not within the valid range (1 ~ 4).	
123	A macro control command was used in tape mode.	
124	DO-END statements do not correspond one to one.	
125	A format error occurred in [Formula].	
126	$1 \leq n \leq 3$ is not satisfied in a DO n statement.	
127	An NC command and macro command are mixed.	
128	$0 \leq n \leq 9999$ is not satisfied in a GO TO n statement.	
129	An invalid address was used in [Argument assignment].	
130	In external data input, the greater-address data contained an error.	
131	In an external alarm message, five or more alarms occurred.	
132	In an external alarm message clear, no corresponding alarm number existed.	
133	In an external alarm message or an external operator message, the smaller-address data contained an error.	
140	The group No. exceeds the maximum value (16).	
141	The tool group specified in the machining program is not set.	
142	The number of tools in the group exceeds 16.	
143	No T code is stored in the block of the program to set tool groups.	
145	Group No. $\Delta\Delta$ specified by T $\Delta\Delta$ 88 does not coincide with the group No. of the tool currently being used in the machining program.	
146	P and L were not specified at the beginning of the program to set the tool group.	
147	The number of the tool groups exceeds the maximum value (16).	
170	Programs with numbers 8000 ~ 8999 and 9000 ~ 9899 are being edited, but parameter setting inhibits these programs from being edited. (See parameters 318-PRG9, and 319-PRG8.)	
210	A movable part of the machine tool has touched the X axis plus stroke limit switch.	
211	A movable part of the machine tool has touched the X axis minus stroke limit switch.	

Number	Content	Remarks
212	While moving along the X axis in the plus direction, the tool entered the forbidden area of stored stroke limit 1.	
213	While moving along the X axis in the minus direction, the tool entered the forbidden area of stored stroke limit 1.	
214	While moving along the X axis in the plus direction, the tool entered the forbidden area of stored stroke limit 2 or 3.	
215	While moving along the X axis in the minus direction, the tool entered the forbidden area of stored stroke limit 2 or 3.	
220	A movable part of the machine tool touched the Z axis plus stroke limit switch.	
221	A movable part of the machine tool touched the Z axis minus stroke limit switch.	
222	When moving along the Z axis in the plus direction, the tool entered the forbidden area of stored stroke limit 1.	
223	When moving along the Z axis in the minus direction, the tool entered the forbidden area of stored stroke limit 1.	
224	When moving along the Z axis in the plus direction, the tool entered the forbidden area of stored stroke limit 2 or 3.	
225	When moving along the Z axis in the minus direction, the tool entered the forbidden area of stored stroke limit 2 or 3.	
400	The controller received the X or Z axis overload signal.	
401	The READY signal (VRDY) of the velocity controller has turned off.	
404	The READY signal (VRDY) of the velocity controller did not turn off even though the READY signal (PRDY) of the position controller turned off. The READY signal (VRDY) of the velocity controller turned on even though the READY signal (PRDY) of the position controller has not yet turned on.	
405	Reference point return might not be performed correctly because of NC or servo system faults. Re-try manual reference point return.	
406	Servo feedback check detects an error in the position transducer. Alarms 414 and 424 are activated for their respective axes simultaneously with this alarm.	
410	The value of the position deviation amount on the X axis is larger than the value allowed while the machine is stopped.	
411	The value of the position deviation amount on the X axis is larger than the value allowed while the machine is moving.	
412	The drift amount on the X axis is excessive. (Exceeds 500VELO)	
413	The value of the position deviation amount on the X axis exceeded ± 32767 . Or the velocity command value of the DA converter is out of the range of $+8191$ to -8192 . Incorrect settings will cause this alarm.	
414	A fault in the position detection system of resolver or inductosyn in the X axis was detected.	
415	A feedrate exceeding 511875 detection unit/sec. was specified for the X axis. An invalid CMR setting causes this error.	

Number	Content	Remarks
416	The position detector of the X-axis pulse coder is faulty (burn-out alarm).	
417	X axis servo position LSI is faulty.	
420	The value of the position deviation amount on the Z axis is larger than the value allowed while the machine is stopped.	
421	The value of the position deviation amount on the Z axis is larger than the value allowed while the machine is moving.	
422	The drift amount on the Z axis is excessive. (Exceeds 500VELO)	
423	The value of the position deviation amount on the Z axis exceeded ± 32767 . Or the velocity command value of the DA converter is out of the range of +8191 to -8192. Incorrect settings will cause this alarm.	
424	A fault in the position detection system of resolver or inductosyn on the Z axis was detected.	
425	A feedrate exceeding 511875 detection units/sec. was specified for the Z axis. An invalid CMR setting causes this error.	
426	The position detector of the Z-axis pulse coder is faulty (born-out alarm).	
427	Z axis servo position LSI is faulty.	
600	A data transfer error took place in the connection unit.	
601	Slave ready has turned off.	
602	The PC program has not yet been loaded. (PC-Model A only)	
603	The correspondence between NC and PC is invalid or was interrupted.	
604	No hold is effective for the PC model B MPU.	
605	A system error occurred in the PC model B MPU (watch dog timer alarm).	
606	A RAM/ROM parity error occurred in the PC model B MPU.	
607	A data transfer error took place in MDI & CRT.	
700	The main PCB overheated.	
702	The DC motor overheated.	
900	A fault occurred in the bubble device (input signal fault for bubble device).	NC alarm
901	A fault occurred in the bubble device (The initial point in the bubble was not detected immediately after power was turned on.)	NC alarm
902	A fault occurred in the bubble device (Page size error, undefined command)	NC alarm
903	A fault occurred in the bubble device (Transfer missing)	NC alarm
904	A fault occurred in the bubble device (Parity error)	NC alarm
905	A fault occurred in the bubble device (No marker)	NC alarm
906	A fault occurred in the bubble device (Many defect loops)	NC alarm
907	A fault occurred in the bubble device (Data cannot be written correctly).	
908	A fault occurred in the bubble device (Soft parity error).	
909	A fault occurred in the bubble device (Bubble Device Stop).	

APPENDIX 8 STATUS WHEN TURNING THE POWER ON, WHEN CLEARED AND WHEN RESET

○: The status is not changed or the movement is continued.

x: The status is cancelled or the movement is interrupted.

Item		When turning power on	When cleared	When reset
Setting data	Offset value	○	○	○
	Data set by the MDI setting operation	○	○	○
	Parameter	○	○	○
Other data	Programs in memory	○	○	○
	Contents in the buffer storage	x	x	○: In the MDI mode x: Mode other than the MDI mode
	Display of sequence number	x	○ (Note 1)	○ (Note 1)
	One shot G code	x	x	x
	Modal G code	Initial G codes (For G20 and G21, the one effective before cutting off the power is effective.)	Initial G codes. G20/G21 and G22/G23, don't change.	No change
	F, E	Zero	Zero	○
	S, T, M	x	○	○
	L	x	x	○: In the MDI mode x: Mode other than the MDI mode
Work coordinate value		Zero	○	○
Action in operation	Movement	x	x	x
	Dwell	x	x	x
	Issuance of M, S and T codes	x	x	x
	Tool offset	x	Depends on system parameter TOC	○: In the MDI mode Depends on system parameter TOC in other modes.
	Tool nose radius compensation	x	x	○: In the MDI mode x: In other than the MDI mode
	Storing called subprogram number	x	x (Note 2)	○: In the MDI mode x: In modes other than the MDI mode (Note 2)
	Rewind	x	x	x

280

Item		When turning power on	When cleared	When reset
Output signals for indication of LEDs	ALM	Extinguishes if there is no cause for the alarm	Extinguishes if there is no cause for the alarm	Extinguishes if there is no cause for the alarm
	NOT READY	x	x (lights in emergency stop)	x (lights in emergency stop)
	LSK	Lights	Lights	○: In the MDI mode Lights in other modes
	BUF	Extinguishes	Extinguishes	○: In the MDI mode Extinguishes in other modes
	REFERENCE POINT RETURN COMPLETION LED	x	○ (x in emergency stop)	○ (x in emergency stop)
	S and T codes	x	○	○
	M code	x	x	x
	M, S and T strobe signals	x	x	x
	Spindle revolution signal (S12 bits/S analogue signal)	○	○	○
	NC ready signal (MA, MB)	ON	○	○
	Servo ready signal	ON (When other than servo alarm)	ON (When other than servo alarm)	ON (When other than servo alarm)
	CYCLE START LED	x	x	x
	FEED HOLD LED	x	x	x

(Note 1) When a program is returned to its head, the start program number is displayed.

(Note 2) When a reset is performed during execution of a subprogram, execution returns to the head of the program. Execution cannot be started from the middle of the subprogram.

APPENDIX 9 STORED PITCH ERROR COMPENSATION FUNCTION

1. Function

Pitch error can be compensated in the unit of the least command increment for each axis. This function becomes effective after reference point return.

2. Specification

The following data on pitch error compensation must be set by parameters on each axis regarding the reference point as a standard position.

- (a) Pitch error compensation zero point (Parameter No. 336, 337)
- (b) Pitch error compensation interval (Parameter No. 383, 384)
- (c) Pitch error compensation amount (Parameter No. 1000 ~ 2127)

Also, pitch error compensation magnification (× 1, × 2, × 4, × 8) can be set to parameter 24 to modify pitch error compensation amount set in (c).

(1) Compensation axis: X and Z

(2) Number of compensation points
128 points (127 sections)

(3) Compensation amount
0 ~ ±7 × (Compensation magnification) (per compensation point) (Unit: least command increment)
Compensation magnification × 1, × 2, × 4, × 8 (common to all axes)

(4) Compensation interval

Command increment	Min. setting interval	Max. setting interval	Unit
Metric system	8000	20000000	0.001 mm
Inch system	4000	20000000	0.0001 inch

(Max. compensation range = Setting interval × 128)

Set a proper compensation interval according to the relation between max. compensation range within the above range and the machine stroke.

If the value less than min. setting value of interval is set in pitch error compensation, the compensation sometimes cannot be performed correctly. In this case, lower the feedrate.

If X axis diameter programming is used, the minimum setting interval will be:

16.000 mm (for metric system)

0.8000 inches (for inch system)

In this case, pitch error compensation data must be set with a diameter value.

3. Parameter

Parameters corresponding to pitch error compensation should be set in MDI mode or in emergency stop status. The parameter contents and parameter number are as follows.

(1) Pitch error compensation magnification

		2	4
--	--	---	---

Parameter number

PML2	PML1							
------	------	--	--	--	--	--	--	--

The value, with this magnification multiplied to the set compensation amount specified in parameter No. 1000 to 2127, is outputted.

PML2	PML1	Magnification
0	0	x 1
0	1	x 2
1	0	x 4
1	1	x 8

(Common to all axes)

(2) Pitch error zero point

	3	3	6
--	---	---	---

PECZRX

	3	3	7
--	---	---	---

PECZRZ

PECZRX, Z

Pitch error zero point of X and Z axes in order.

Pitch error zero point is the zero point of pitch error compensation table. Set the value from 0 to 127 for each axis according to the machine tool.

(3) Pitch error compensation interval

0	3	8	3
---	---	---	---

PECINTX

0	3	8	4
---	---	---	---

PECINTZ

PECINTX, Z

Pitch error compensation interval of X and Z axes in order.

Set the positive amount 8000 or more in metric system, or 4000 or more in inch system.

When zero is set to these parameter, compensation is not performed.

Setting unit: 0.001 mm (metric system)

0.0001 inch (inch system)

If X axis diameter programming is used, this amount shall be set to 16000 or more (metric system) or 8000 or more (inch system) for X axis.

In this case, this parameter shall be set with a diameter value.

(4) Compensation amount

Compensation amount of each axis and each point is set to the following parameters.

Axis name	Parameter number
X axis	1000 ~ 1127
Z axis	2000 ~ 2127

For the other parameter number shown above, no compensation amount can be set. Setting amount is 0 to ± 7 and the other values are neglected.

Setting unit: 0.001 mm (metric system)

0.0001 inch (inch system)

If X-axis diameter programming is used, this amount shall be set with a diameter value.

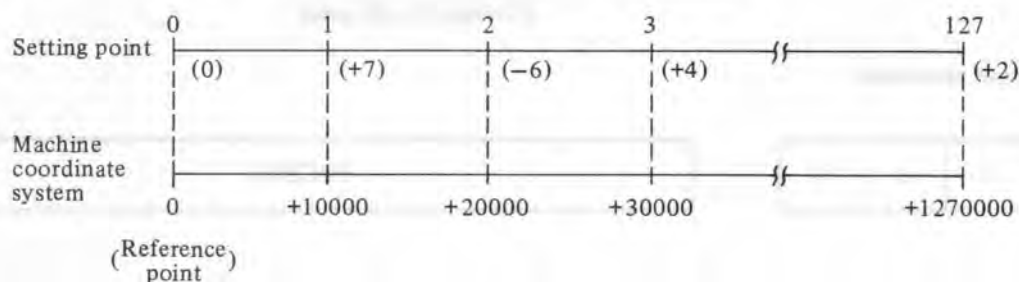
However, this amount is multiplied by 1, 2, 4 or 8 depending on the setting of parameter No. 24.

4. Example

(1) Example 1

Pitch error zero point = 0

Compensation interval = 10000



The beginning of the compensation table corresponds to the reference point. The set point 1 is at +10000 position and the other setting points correspond to the positions on the machine coordinate system which is shifted +10000 each other. Therefore, setting point 128 is at +1270000.

Compensation amount when the machine moves from 0 to +10000 is set to the setting point 1 and that when the machine moves from +10000 to +20000 is set to the setting point 2. Generally, compensation amount when the machine moves from $(n - 1) \times (\text{compensation interval})$ to $n \times (\text{compensation interval})$ is set to the setting point n .

In the above example, compensation data is

-7 between 0 and 10000

+6 between 10000 and 20000

-4 between 20000 and 30000

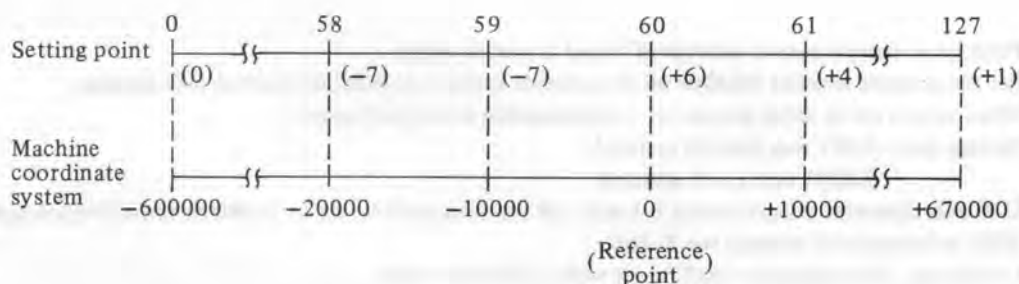
If the machine moves from the reference point to +30000, total compensation amount is as follows.

$$(+7) + (-6) + (+4) = +5$$

(2) Example 2

Pitch error zero point = 60

Compensation interval = 10000



The setting point 60 corresponds to the reference point. The setting point 61 is at +10000 position and the other setting points corresponds to the positions which is shifted +10000 each other. And compensation point 127 is at +670000.

And, the setting point 59 is at -10000 from the reference point and the other setting points corresponds to the positions which is shifted -10000 each other, that is, the setting point 0 is at -600000.

Compensation amount when the machine moves from $(n - 61) \times (\text{compensation interval})$ to $(n - 60) \times (\text{compensation interval})$ is set to the setting point n .

In the above example, compensation data is as follows.

+7 between -20000 and -10000

-6 between -10000 and 0

-4 between 0 and +10000

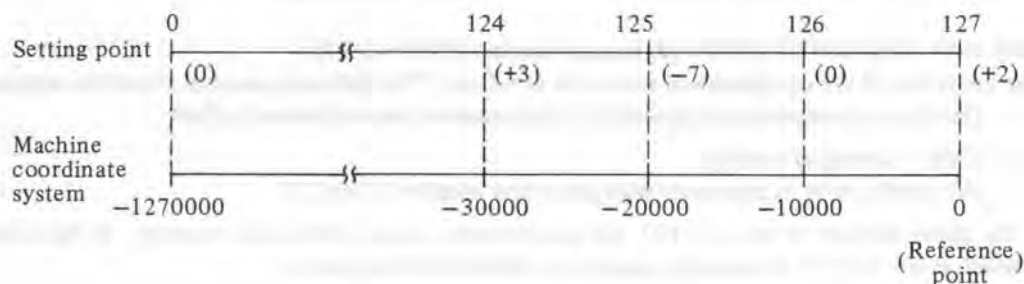
If the machine moves from -10000 to +10000, total compensation amount is as follows.

$$(-7) + (+6) + (+4) = (+3)$$

(3) Example 3

Pitch error zero point = 127

Compensation interval = 10000



The end of the compensation table corresponds to the reference point and the setting point 126 is at -10000 from the reference point and the other setting points correspond to the position which is shifted -10000 each other. Setting point 0 is at -1270000 in the machine coordinate system.

Compensation amount when the machine moves from -10000 to 0 is set to the setting point 127 and that when the machine moves from -20000 to -10000 is set to the setting point 126. Generally, compensation amount when the machine moves from $(n - 128) \times (\text{compensation interval})$ to $(n - 127) \times (\text{compensation interval})$ is set to the setting point n .

In the above example, compensation data is as follows.

- 3 between -40000 and -30000
- +7 between -30000 and -20000
- 0 between -20000 and -10000
- 2 between -10000 and 0

If the machine moves from -40000 to the reference point, total compensation amount is as follows.

$$(+3) + (-7) + (0) + (+2) = (-2)$$

5. Notes on compensation amount

The compensation amount described above do not directly concern the relationship between the reference point and the compensation zero point, moving direction of the machine and compensation interval, but mechanical error between $[n - \{ (\text{Compensation zero point}) + 1 \}] \times (\text{Compensation interval})$ and $\{ n - (\text{Compensation zero point}) \} \times (\text{Compensation interval})$ for setting point n .

Accordingly, if the machine slide moves more than the specified move distance (positive error), set the compensation amount with a negative value. If the machine slide moves less than the specified move distance (negative error), set it with a positive value.

6. Others

(1) Deletion of compensation amount

If the previous compensation values are to be deleted to set new compensation values, they can be deleted on each axis by the following method.

- (1) Select a parameter number to which compensation amount is set on the axis. (Any parameter number is allowed so long as it belongs to the axis.)
- (2) Specify -9999 as a pitch error compensation amount.

(2) Punch out of compensation amount

They are punched out by the same procedure as parameter punch out. However, in the procedure, input -9998 instead of -9999.

Punching out of the desired axis only cannot be performed.

(3) Precautions in setting

- (a) Compensation interval (Parameter number 383, 384)

If the compensation interval is positive value, compensation is performed by the set value. If it is negative value, compensation is performed by the absolute value of it. And if it is zero, the compensation for the axis is not performed. (Even if it is negative value, it is displayed as positive value.)

- (b) The pitch error compensation becomes effective after reference point return. If reference point return is not performed, it is not effective after parameter setting. The parameter setting should be performed after turning power on and before reference point return. If the parameters for pitch error compensation is altered after reference point return, perform reference point return again.

- (c) Pitch error compensation amount (Parameter number 1000 ~ 2127)

The limitation of the compensation amount is as follows. The following amount should be within ± 127 .

$$(\text{Pitch error compensation amount}) \times (\text{Compensation magnification}) \times \text{CMR}$$

(Note) CMR: Command multiply

For details, refer to parameter with parameter number 27 and 28.

If the above amount is out of ± 127 , the compensation is not performed correctly. If the compensation amount which is out of ± 127 is necessary, assign it to adjacent setting points.

APPENDIX 10 LIST OF OPERATIONS

Classification	Function	Key Switch	Parameter Enable Switch	Mode Switch	Function key	Operation
Clear	Bubble Memory All Clear		☐	Power ON	-	O + DELET
	Clearing Parameter		☐	Power ON	-	CAN + DELET
	Clearing Stored Program			Power ON	-	RESET + DELET
Data Input from Tape	Parameter		☐	Emergency Sw. ON.	PARAM	P → -9999 → READ
	Offset Value			TAPE	-	START
	Program Input	☐		EDIT	-	O → Program No. → READ
	Add Program	☐		EDIT	-	O → CAN → READ
	Registration of all programs			EDIT	-	O → -9999 → READ
Data Input from MDI	Parameter		☐	MDI	PARAM	N → Parameter No. → INPUT → P → Data → INPUT → Parameter Enable Switch OFF → RESET
	Offset value	☐		Any mode (except EDIT)	OFFSET	N → Offset No. → INPUT → * → Offset Data → INPUT (Note)
	Setting Data	☐		MDI	SET	CURSOR to Set No. → P → Data → INPUT
	Parameter			EDIT	PARAM	P → -9999 → PUNCH
Tape Punch	Offset Value			EDIT	OFFSET	P → -9999 → PUNCH
	All Program			EDIT	-	O → -9999 → PUNCH
	One Program			EDIT	-	O → Program No. → PUNCH

(Note) : , , , , or

Classification	Function	Key Switch	Parameter Enable Switch	Mode Switch	Function key	Operation
Search	Program No. Search (MEMORY Mode Only)			EDIT	PRGRM ① PRGRM ②	0 → Program No. → CURSOR ↓ 0 → CAN → CURSOR ↓
	Sequence No. Search (MEMORY Mode Only)			MEMORY	PRGRM	Program No. Search → N → Sequence No. → CURSOR ↓
	Sequence No. Search (Tape Mode)			TAPE	PRGRM	N → Sequence No. → CURSOR ↓
	Address Word Search (MEMORY Mode Only)			EDIT	PRGRM	Searching Address and Data Input → CURSOR ↓
Program Editing	Address Search (MEMORY Mode Only)			EDIT	PRGRM	Searching Address → CURSOR ↓
	Deletion of all Programs	○		EDIT	PRGRM	0 → -9999 → DELET
	Deletion of a Program	○		EDIT	PRGRM	0 → Program No. → DELET
	Deletion of Several Blocks	○		EDIT	PRGRM	N → Sequence No. → DELET
	Deletion of a Block	○		EDIT	PRGRM	Search for the Block to be Deleted → EOB → DELET
	Deletion of a Word	○		EDIT	PRGRM	Search for the Word to be Deleted → DELET
	Modification of a Word	○		EDIT	PRGRM	Search for the Word to be Modified → Address → Data → ALTER
	Insertion of a Word	○		EDIT	PRGRM	Search for the Word before the Place in the Program → Address → Data → INSRT
Comparison	Memory Arrangement	○		EDIT	PRGRM	CAN → ORIGN
	Comparison in Memory with Tape			EDIT	PRGRM	/ → READ
	Comparison from Current Position			EDIT	PRGRM	EOB → READ

Classification	Function	Key Switch	Parameter Enable Switch	Mode Switch	Function key	Operation
Bubble cassette	Heading of file			EDIT mode	PRGRM	$\boxed{N} \rightarrow 0 \text{ (Zero)} \rightarrow \boxed{\text{INPUT}}$
				EDIT mode	PRGRM	$\boxed{N} \rightarrow -9999 \rightarrow \boxed{\text{INPUT}}$
				EDIT mode	PRGRM	$\boxed{N} \rightarrow -9998 \rightarrow \boxed{\text{INPUT}} \text{ (Modal)}$
				EDIT mode	PRGRM	$\boxed{N} \rightarrow \text{File no.} \rightarrow \boxed{\text{INPUT}}$
	Deletion of file	$\bigcirc$		EDIT mode	PRGRM	$\boxed{N} \rightarrow \text{File no. (k)} \rightarrow \boxed{\text{START}} \rightarrow \text{File no. (k+1)} \sim n \Rightarrow k \sim (n-1)$
	Output of one program			EDIT mode	PRGRM	$\boxed{O} \rightarrow \text{Program no.} \rightarrow \boxed{\text{PUNCH}}$
	Output of all programs			EDIT mode	PRGRM	$\boxed{O} \rightarrow -9999 \rightarrow \boxed{\text{PUNCH}}$
	Loading of programs			EDIT mode	-	Heading of file $\rightarrow \boxed{O} \rightarrow -9999 \rightarrow \boxed{\text{READ}}$
	Output of offset data			EDIT mode	OFFSET	$\boxed{P} \rightarrow -9999 \rightarrow \boxed{\text{PUNCH}}$
	Input of offset data			EDIT mode	-	Heading of file $\rightarrow \boxed{O} \rightarrow \text{Program no.} \rightarrow \boxed{\text{READ}} \rightarrow \text{Program execution}$
	Output of parameters			EDIT mode	PARAM	$\boxed{P} \rightarrow -9999 \rightarrow \boxed{\text{PUNCH}}$
	Input of parameters		$\bigcirc$	Emergency stop ON	PARAM	Heading of file $\rightarrow \boxed{P} \rightarrow -9999 \rightarrow \boxed{\text{READ}}$

APPENDIX 11 PROGRAM KEY LOCK

1. Introduction

Programs 9000 to 9899 can be locked. While locked, these programs cannot be displayed, edited, and punched out. This function protects the secrecy of special programs created by the machine tool builder by using macros and prevents them from being erased by mistake.

2. Program number

Only programs 9000 to 9899 can be locked. According to a certain method, described later, they are locked automatically. Therefore, programs not to be locked must be assigned with numbers other than 9000 to 9899.

3. Locked conditions

In a locked state (see Section 4), programs 9000 to 9899 are as follows:

- 1) The contents are not displayed even during execution.
- 2) Program number search in the edit (memory) mode cannot be performed (alarm 071). Therefore, nor can editing.
- 3) Memory arrangement is not carried out.
- 4) The program numbers are not included in All Program Number Display.
- 5) The programs cannot be punched out (even by All Program Punch-Out).
- 6) The programs cannot be cancelled (even by All Program Cancel).
- 7) The programs cannot be registered in the memory (alarm 170).

4. Locking and unlocking method

- 1) Set the secret numbers (1 to 99,999,999) in parameter 387 in advance. Note the contents of this parameter. Setting zeros does not effect locking.
(Note 1) This parameter can be set and changed only in an unlocked state.
(Note 2) This parameter cannot be cancelled even by All Parameter Clear.
(Note 3) This parameter is cleared to zero by Bubble All Clear, that is, an unlocked state is entered.
- 2) For unlocking, enter into parameter 408 the same number as that set in parameter 387.
(Note 1) The contents of this parameter are not displayed.
(Note 2) This parameter is not stored in the bubble memory.
- 3) To lock following unlocking, take either of the following actions.
 - a) Enter into parameter 408 a number different from that set in parameter 387.
 - b) Turn the NC power off once and on again.

	3	8	7
--	---	---	---

Secret number

Register a secret number to be locked in advance.
Setting range: 1 to 99,999,999

	4	0	8
--	---	---	---

Lock/Unlock

Entering the same number as that of parameter 387 effects unlocking. Entering a different number effects locking.

(Note 1) Whenever parameter 387 is zero, an unlocked state is provided. Turning off/on the NC power does not effect locking. Note that any number other than zero should not be set in parameter for programs not to be locked.

5. Notes

- (1) If the set secret number (parameter 387) has become unknown, take the following action.
 - (a) Perform Bubble All Clear (unlocked).
 - (b) Enter all parameters (except parameter 387).
 - (c) Store the secret program in the memory.
 - (d) Set a secret number in parameter 387 (locked).
 - (e) Input offset data and other programs.
- (2) For replacement of the faulty bubble memory, also follow operation 2) above. For quick recovery, secret programs must be attached to the NC equipment.
- (3) If the mode is changed to EDIT during execution of a locked program, the PRGRM screen will be displayed as follows. That is, the program will not be displayed. Execute reset before displaying and editing of other programs. (Executing reset does not permit the program to be continued.) Executing reset returns the screen to an ordinary program display. To continue the program, return the mode to MEMORY or TAPE and resume machining by pressing the cycle-start button.



- (4) After registering or editing programs 09000 to 09899, call out a different program and effect locking. Locking, while programs 09000 to 09899 are displayed, provides the screen with the figure above in the edit mode. Applying reset displays the program.

APPENDIX 12 CUSTOM MACRO INTERRUPT FUNCTION

1. OUTLINE

Entering an interrupt signal (UNIT) from the machine during execution of a program permits call-out of another program. This function is called the macro interrupt function.

The interrupt command is executed by specifying, in programming,

M96 Pxxxx ; (Custom macro interrupt on)

M97 ; (Custom macro interrupt off)

This function permits any execution block of the program to call out another program, thus enabling program operations according to the ever changing conditions.

[Applications]

1. Processing for tool error detection is initiated by an external signal.
2. Another program is inserted in a machining series without the stopping of machining operation.
3. The current machining information is read at fixed time intervals, etc.

Thus, adaptive control-like applications are possible.

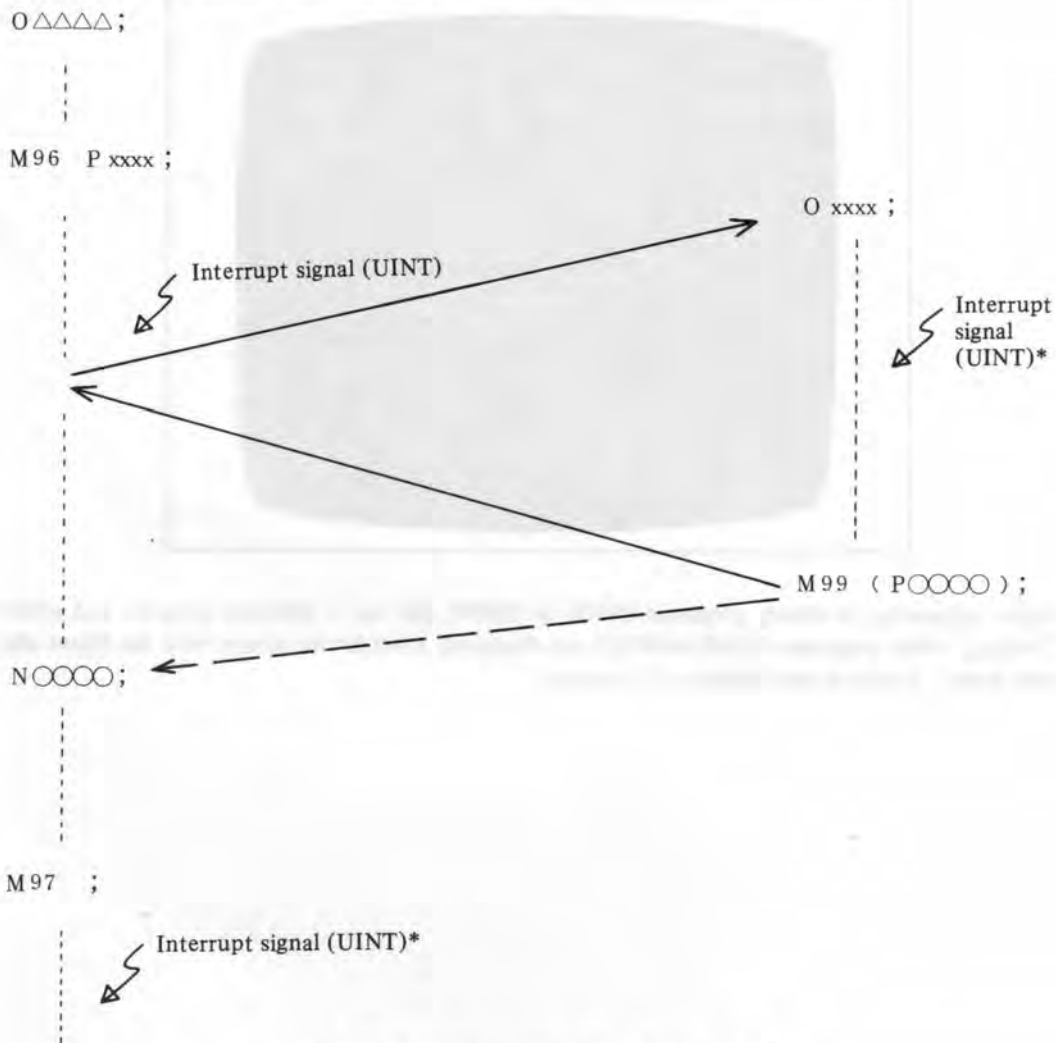


Figure 1 Outline of custom macro interruption

The figure shows that specifying M96 Pxxxx in programming permits a program specified by Pxxxx to interrupt the current program and be executed, when the interrupt signal (UNIT) is entered. The interrupt signal is ignored during execution of the interrupt program and after M97 (interrupt signals marked *).

2. COMMANDING

2.1 Effective Conditions

Custom macro interruption can be used only during program execution. Therefore, it is effective when:

1. The memory, tape, or MDI mode has been selected;
2. The start lamp (STL) is on;
3. The custom macro interruption is not in progress.

The macro interruption cannot be applied during manual operations (jog, step, handle, etc.).

2.2 Command Format

The custom macro interrupt function is, in principle, executed by making the interrupt signal effective and ineffective by M96 and M97. That is, the interrupt signal entered until M97 is commanded or the NC is reset, after M96 is commanded, initiates the custom macro interruption. On the other hand, entering the interrupt signal after M97 is commanded or after the NC is reset, does not initiate the custom macro interruption, but the interrupt signal entered until M96 is commanded is ignored.

[Format]

M96 Pxxxx ; Custom macro interrupt on

↑
Specifies the interrupt program number.

M97 ; Custom macro interrupt off

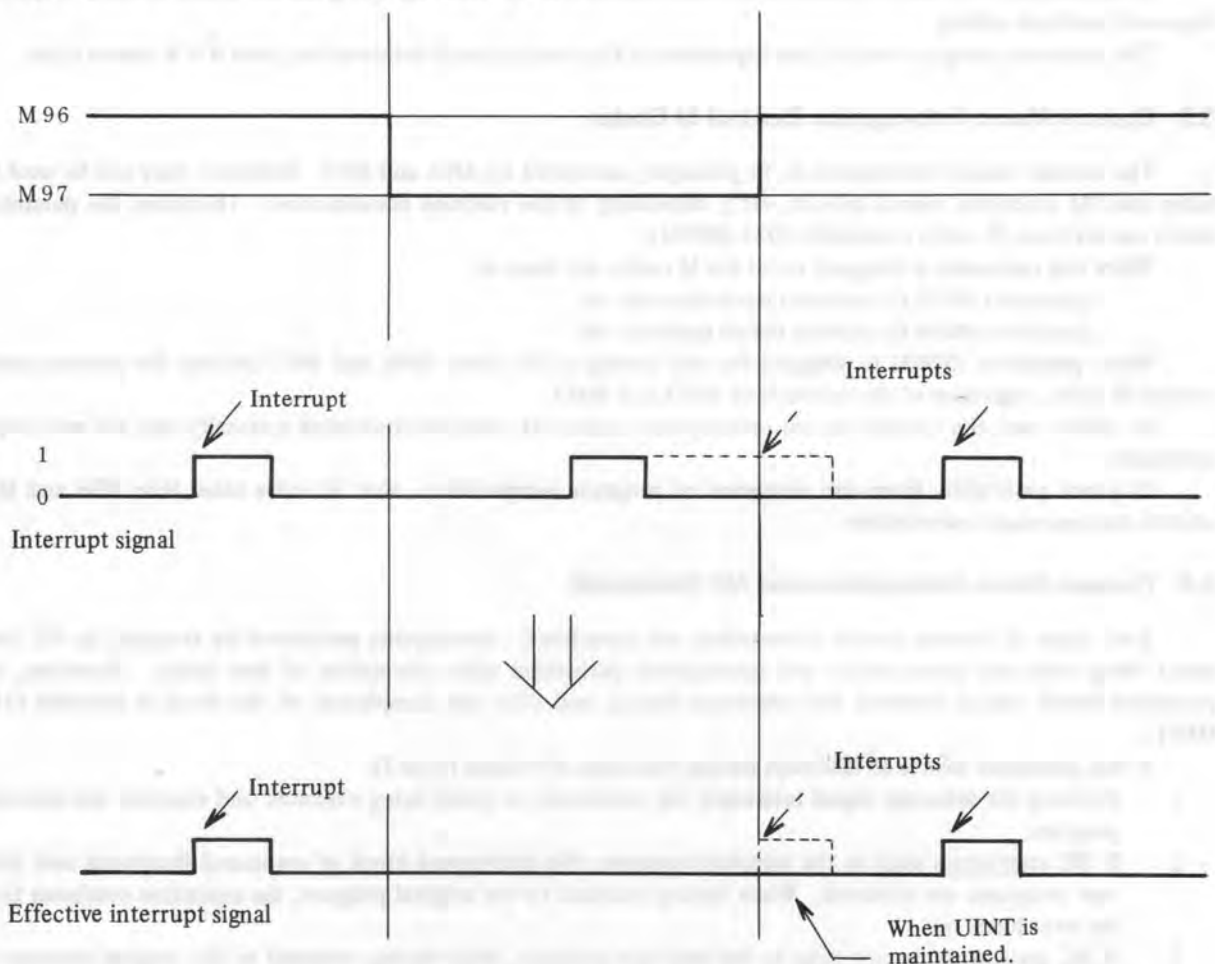


Figure 2.2 Relations between M96, M97, and interrupt signal

The interrupt signal becomes effective after M96 is entered. It is ignored after M97 is commanded, but if the signal entered after M97 is commanded is maintained until M96 is commanded, it initiates the custom macro interruption immediately after M96 is commanded.

3. DETAILED DESCRIPTIONS

3.1 Custom Macro Interruption Enable/Mask

When the custom macro interrupt function is not to be used, it can be assumed that the program is not to be modified. Thus, the parameter which selects between Enable and Mask for the custom macro interrupt function is available (025-MUSR).

When this parameter specifies that the custom macro interrupt function is to be masked, M96 and M97 become ordinary M codes and are output externally, but when it specifies that the function is to be enabled, they are processed internally and are not output externally.

3.2 Subprogram-type Interrupt and Macro-type Interrupt

The custom macro interrupt method includes subprogram type and macro type. Thus, the parameter which selects between the two types is available (025-MSUB).

- Subprogram type

The interrupt program is called out as a subprogram. That is, the local variable level does not change before and after the interruption. Also, this interruption is not added to the degrees of multiple subprogram calling.

- Macro type

The interrupt program is called out as a custom macro. That is, the local variable level changes before and after the interruption. Also, this interruption is not added to the degrees of multiple macro calling.

A subprogram call and a custom macro call executed in the interrupt program are added to their respective degrees of multiple calling.

The execution program cannot pass arguments in the custom macro interruption, even if it is macro type.

3.3 Custom Macro Interruption Control M Codes

The custom macro interruption is, in principle, controlled by M96 and M97. However, they can be used for other uses (M functions, macro call-out, etc.), depending on the machine manufacturer. Therefore, the parameter which can set these M codes is available (025-MPRM).

When this parameter is designed to set the M codes, set them in:

parameter #053 for custom macro interrupt on;

parameter #054 for custom macro interrupt off.

When parameter MPRM is designed for not setting of M codes, M96 and M97 become the custom macro control M codes, regardless of the contents of #053 and #054.

In either case, the custom macro interruption control M codes are processed internally and are not output externally.

It is not preferable, from the viewpoint of program compatibility, that M codes other than M96 and M97 control custom macro interruption.

3.4 Custom Macro Interruption and NC Command

Two types of custom macro interruption are considered: interruption performed by stopping an NC command being executed prematurely; and interruption performed after completion of that block. Therefore, the parameter which selects between the interrupts during, and after the completion of, the block is available (314-MINT).

If this parameter selects an interrupt during execution of a block (type I):

1. Entering the interrupt signal interrupts the movement or dwell being executed and executes the interrupt program;
2. If NC statements exist in the interrupt program, the interrupted block of command disappears and interrupt programs are executed. When having returned to the original program, the execution continues from the next block on;
3. If NC statements do not exist in the interrupt program, when having returned to the original program by M99, the execution continues from the interrupted command on.

If this parameter selects an interrupt after completion of a block (type II):

1. Entering the interrupt signal executes the interrupt program without interrupting the command being executed;

2. If NC statements exist in the interrupt program, they are executed after completion of the command being executed.

In either case, note that control passes to the interrupt program as soon as the interrupt signal is input.

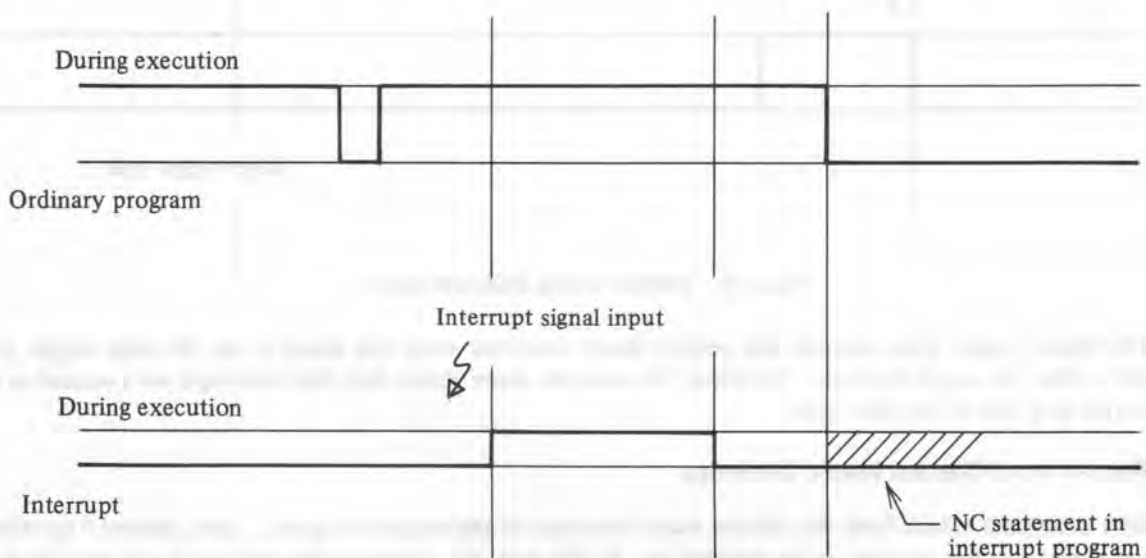
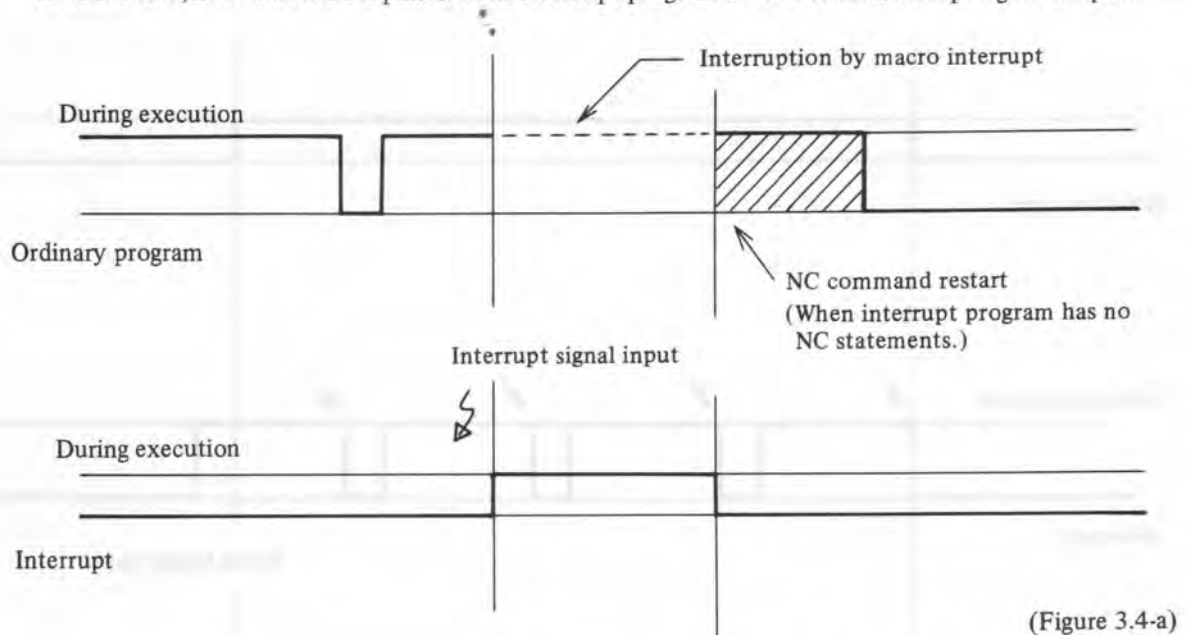


Figure 4 Custom macro interruption and NC command

The figure shows an interrupt during execution of a block (3.4-a) and an interrupt after completion of the block (3.4-b). In either case, the interrupt is executed when the interrupt signal has been input.

3.5 Acceptance of Custom Macro Interrupt Signal (UNIT)

There are two methods of accepting the custom macro interrupt signal: status trigger and edge trigger. The status trigger makes the signal effective when it is on; the edge trigger makes it effective when it turns from off to on. The parameter selects between the two types (025-TSE).

When this parameter specifies the status trigger, if the interrupt signal is on (1) when it has become effective, the custom macro interrupt occurs. Therefore, while the signal is on, the interrupt program can be executed repeatedly.

When this parameter specifies the edge trigger, the interrupt signal becomes effective only when it turns on, and the interrupt program is terminated in a moment (programs of macro statements alone, etc.). Therefore, when the status trigger is not suitable, or when the custom macro interrupt is to be performed only once through the whole program (the interrupt signal stays on), this type may be used.

Either type will provide the same effect in practice, except for special uses, such as noted above. (There are no differences, e.g., one type takes longer than the other from the input of the signal to the execution of the custom macro interrupt.)

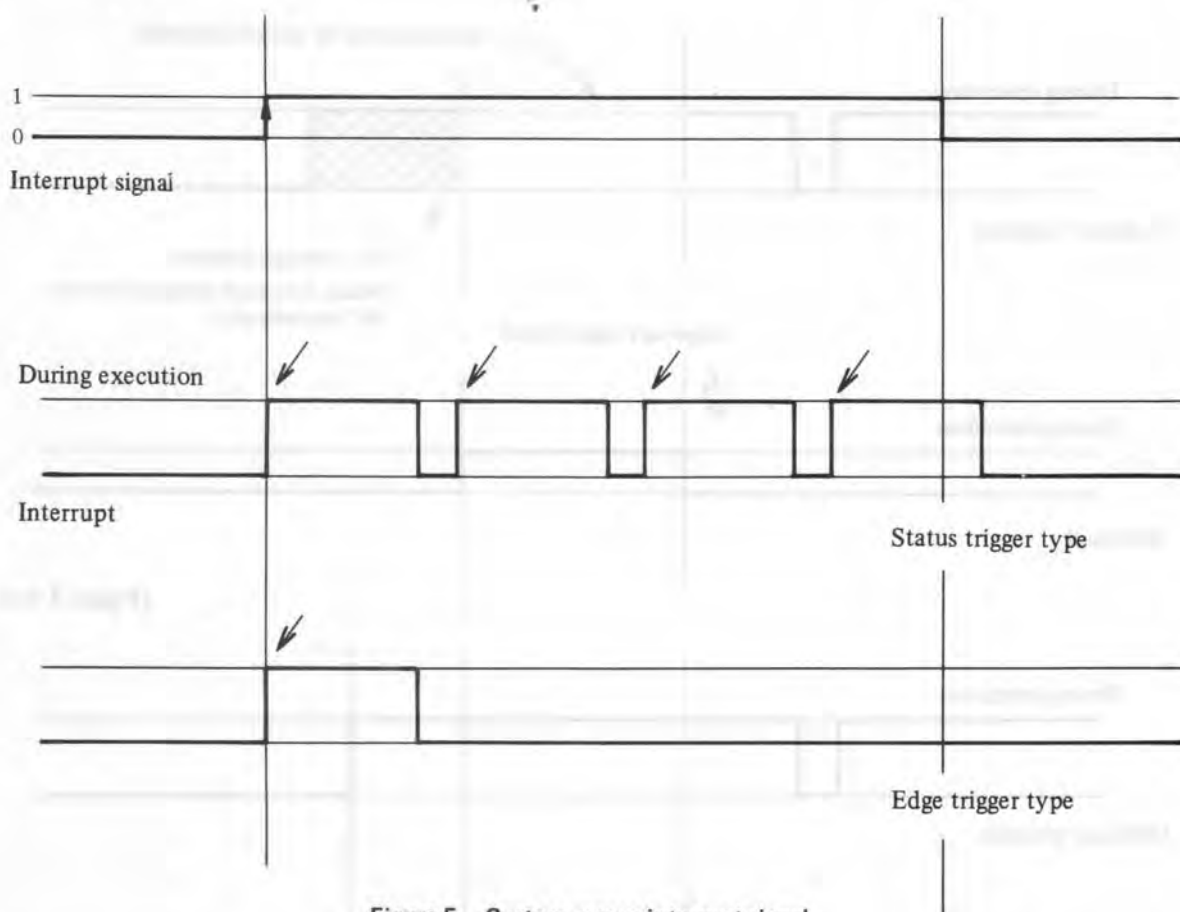


Figure 5 Custom macro interrupt signal

The status trigger type executes the custom macro interrupt when the signal is on; the edge trigger type executes it when the signal turns on. Therefore, the example above shows that four interrupts are executed in the former type; only one in the latter type.

3.6 Return from Custom Macro Interrupt

M99 commands return from the custom macro interrupt to the original program. Also, address P specifies a sequence number in the program to be returned to. In this case, the corresponding program is searched from its beginning and the sequence number appearing first is returned to (same as M98).

Another interrupt does not occur during execution of a custom macro interrupt program, but M99 clears this situation. When M99 is specified independently*, it is executed before completion of the preceding command. Therefore, the custom macro interrupt also becomes effective for the last command of the interrupt program. If this is inconvenient, control the custom macro interrupt by M96 and M97 in the program.

* : block with M99 alone

The M99 block, which consists of addresses O, N, P, L, or/and M alone, is assumed to be the same block as the preceding block in programming. Therefore, the single block stop is not performed. In programming,

Gxx Xxxxx ;

M99 ; and

Gxx Xxxxx M99 ;

provide the same effect.

(They differ in whether Gxx is executed or not before M99 is known.)

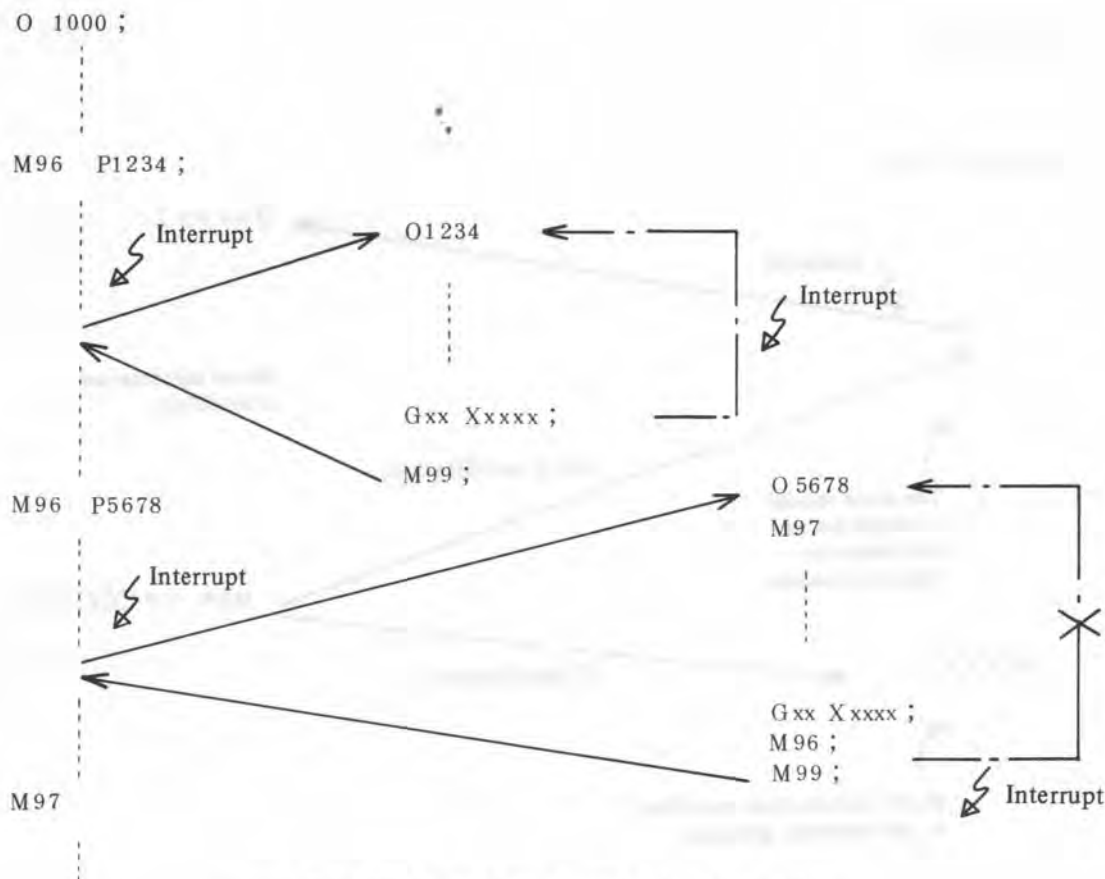


Fig. 3.6 Return from custom macro interrupt

A custom macro interrupt does not overlap with another which is being executed. That is, if an interrupt occurs, another is prohibited automatically; if M99 is executed, the custom macro interrupt becomes effective again. Since the M99 block is executed independently before completion of the preceding block, it can also interrupt the Gxx block of O1234, and if the signal is input, the O1234 program is executed again. On the other hand, the O5678 program is controlled by M96 and M97 and the interrupt becomes effective after it returns to O1000.

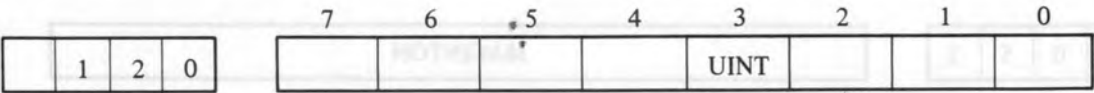
3.7 Custom Macro Interruption and Modal Information

The custom macro interrupt, unlike the ordinary program call, is initiated by the interrupt signal. Therefore, it is not desirable that modal information is modified in the interrupt program and as a result, the original program is affected. Thus, even if modal information is modified in the interrupt program, when the original program is returned to by M99, the modal information is restored to that before interruption.

When the interrupt program returns to the original program by M99 Pxxxx, modal information can be controlled in the program. Therefore, the modal information modified in the interrupt program is taken over. (Conversely, when modal information in the original program is to be taken over, movement after return may vary depending on the modal information at the time of interruption.) Therefore, in this case:

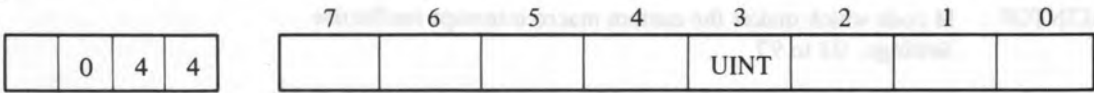
- 1) Command modal information in the interrupt program
- or
- 2) Command necessary modal information at the returned point.

5. DIAGNOSE



UNIT 1 : Custom macro interrupt on
0 : Custom macro interrupt off

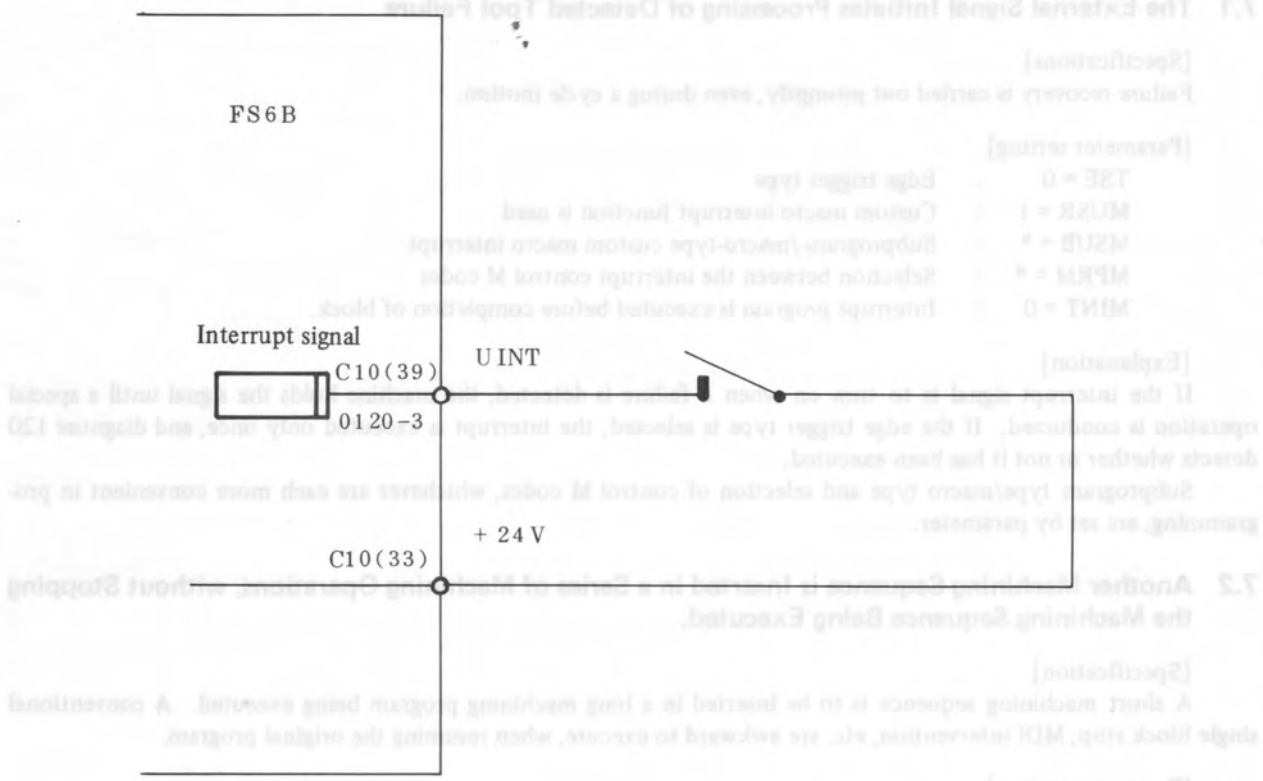
This signal is an external signal for custom macro interrupt. This signal, considering uses requiring high velocity, detects not only signals created by the magnetics control but also external signals directly. Therefore, the machine manufacturer should connect the external signal to the signal position noted below and not use it for other uses.



Verify that the diagnose numbers are #044 and #120.



6. INTERNAL CONNECTION DIAGRAM



To connect this signal, connection unit 2 is required.

7. APPLICATION EXAMPLES

7.1 The External Signal Initiates Processing of Detected Tool Failure

[Specifications]

Failure recovery is carried out promptly, even during a cycle motion.

[Parameter setting]

TSE = 0 : Edge trigger type
MUSR = 1 : Custom macro interrupt function is used.
MSUB = * : Subprogram-/macro-type custom macro interrupt
MPRM = * : Selection between the interrupt control M codes
MINT = 0 : Interrupt program is executed before completion of block.

[Explanation]

If the interrupt signal is to turn on when a failure is detected, the machine holds the signal until a special operation is conducted. If the edge trigger type is selected, the interrupt is executed only once, and diagnose 120 detects whether or not it has been executed.

Subprogram type/macro type and selection of control M codes, whichever are each more convenient in programming, are set by parameter.

7.2 Another Machining Sequence is Inserted in a Series of Machining Operations, without Stopping the Machining Sequence Being Executed.

[Specification]

A short machining sequence is to be inserted in a long machining program being executed. A conventional single block stop, MDI intervention, etc. are awkward to execute, when resuming the original program.

[Parameter setting]

TSE = * : Selection between the trigger types
MUSR = 1 : Interrupt function is used.
MSUB = 0 : Macro-type interrupt
MPRM = * : Selection between the interrupt control M codes
MINT = 1 : Interrupt program is executed after completion of block.

[Explanation]

Selection between the trigger types and between the interrupt control M codes, whichever are each more convenient, is set by parameter. In order for the custom macro interrupt not to affect the block being executed, the interrupt is prohibited during execution of the block, and the macro-type interrupt is taken. The interrupt program restores modal information, machine position, etc. at the time of interruption, when returning to the original program, whereby any program can be performed. If fixed, the program can be specified directly by M96 Pxxx; if not, it can be called out by M98 P#100, etc.

7.3 Machining Information is Read at Fixed Time Intervals.

[Specifications]

For machining conditions management, machining information is transmitted externally at fixed time intervals. The machining sequence is not to be affected.

[Parameter setting]

TSE = 0 : Edge trigger type
MUSR = 1 : Interrupt function is used.
MSUB = 0 : Macro-type interrupt
MPRM = * : Selection between the interrupt control M codes
MINT = 0 : Interrupt program is executed before completion of block.

[Explanation]

Suppose that the interrupt program includes no NC statements. The edge trigger type is taken (because the status trigger type repeats interrupts continuously while the signal is on), and turning on and off the signal at each time interval initiates the interrupt program once. Since the interrupt is possible during execution of blocks, as specified above, interrupts are carried out each time the signal turns from off to on. Machining information, such as modal information and position information, is output externally by the use of Custom Macro DO.

The interrupt program is carried out concurrently with the original program block; the machining operation stops for a moment, when the interrupt program is not completed, even if the original program block is completed.

7.4 The Same Program Is Selectively Used for Ordinary Cutting and for Special Cutting.

[Specifications]

A special motion is to be added at each execution of blocks. It is inconvenient to specify such a command in programming.

[Parameter setting]

TSE = 1	:	Status trigger type
MUSR = 1	:	Interrupt function is used.
MSUB = *	:	Selection between the subprogram and macro types
MPRM = *	:	Selection between the interrupt control M codes
MINT = 1	:	Interrupt program is executed after completion of block.

[Explanation]

The interrupt program is commanded as:

```
Oxxxx ;  
M97 ;      Prohibits interrupt.  
.  
.  
.  
M96 ;      Enables interrupt.  
M99 ;
```

The interrupt signal stays on; the status trigger type is taken, whereby the custom macro interrupt is executed each time one block is completed in the original program. The interrupt program specifies in advance a special motion to be executed. M97 makes interrupts invalid for program portions for interrupts not to be executed.

The interrupt program is carried out concurrently with the original program block; the machining operation stops for a moment, when the interrupt program is not completed, even if the original program block is completed.

7.4 The Same Program is Selectively Used for Ordinary Cutting and for Special Cutting.

[Specifications]
A special motion is to be added at each execution of block. It is inconvenient to specify such a command in programming.

[Parameter setting]
TSE = 1 : Status trigger type
MUR = 1 : Interrupt function is used.
MSUB = * : Selection between the subprogram and master type
MPRM = * : Selection between the interrupt control M codes
MINT = 1 : Interrupt program is executed after completion of block.

[Explanation]
The interrupt program is commanded as:

Gxxx ;
M97 ; Prohibit interrupt.

M98 ; Enable interrupt.
M99 ;

The interrupt signal stays on; the status trigger type is taken, whereby the custom macro interrupt is executed each time one block is completed in the original program. The interrupt program specifies in advance a special motion to be executed. M97 makes interrupts invalid for program portions for interrupts not to be executed.

- *No part of this manual may be reproduced in any form.*
- *All specifications and designs are subject to change without notice.*

1983. 8
(Y 2300)

FANUC LTD